

Univerzita Hradec Králové

Pedagogická fakulta

Katedra informatiky Přírodovědecké fakulty

## **Výzkum koncepcí ve výuce programování**

Disertační práce

Autor: Mgr. Ondřej Kořínek

Studijní program: P 7507 Specializace v pedagogice

Studijní obor: Informační a komunikační technologie ve  
vzdělávání

Školitel: doc. RNDr. Štěpán Hubálovský, Ph. D.

2017

„Prohlašuji, že jsem tuto disertační práci vypracoval pod vedením školitele samostatně a uvedl jsem všechny použité prameny a literaturu.“

V Hradci Králové 25. 1. 2017

.....  
Mgr. Ondřej Kořínek

## Poděkování

Děkuji svému školiteli, doc. RNDr. Štěpánovi Hubálovskému, Ph.D., za vedení disertační práce, připomínky při jejím zpracování a podporu při psaní práce v průběhu doktorského studia.

## **Abstrakt**

KOŘÍNEK Ondřej. *Výzkum koncepcí ve výuce programování*. Hradec Králové, 2017. 215 s. Disertační práce. Univerzita Hradec Králové, Pedagogická fakulta, Katedra informatiky Přírodovědecké fakulty.

Tématem disertační práce je výzkum koncepcí ve výuce programování objects-first a objects-later. V práci jsou popsána teoretická východiska nutná k dosažení cílů disertační práce. Teoretickými východisky práce jsou paradigmaty programování, koncepce výuky programování a výzkumy koncepcí výuky programování objects-first a objects-later. Výzkumy koncepcí výuky programování různých autorů jsou východiskem vlastního výzkumu v disertační práci.

Vlastní výzkum probíhal ve formě pedagogického experimentu na Přírodovědecké fakultě Univerzity Hradec Králové v akademických letech 2013/2014 a 2014/2015 v předmětu Programování 1. V akademickém roce 2013/2014 se výzkumu v předmětu Programování 1 zúčastnilo 28 studentů. V akademickém roce 2014/2015 se výzkumu v předmětu Programování 1 zúčastnilo 34 studentů.

Studenti byli, před vlastní realizací pedagogického experimentu, rozděleni do experimentální a kontrolní skupiny podle výsledků zkouškového testu (pretest vlastního výzkumu) v předmětu Algoritmy a datové struktury.

V průběhu výuky předmětu Programování 1 studenti absolvovali jednotlivé testy (posttest, retest 1 a retest 2). Učební materiály byly pro studenty k dispozici na kurzu v prostředí Moodle.

Ve vlastním výzkumu byly zpracovány výsledky všech testů studentů. Anonymní dotazníkové šetření, které studenti vyplnili na konci semestru výuky předmětu Programování 1, zjišťovalo možný vliv vyučujícího, dostatečnost výukových materiálů nebo přípravu studentů.

## **Klíčová slova**

Koncepce výuky programování, objects-first, objects-later, paradigmaty programování, pedagogický experiment

## **Abstract**

KOŘÍNEK Ondřej. *Research of concepts of teaching programming*. Hradec Králové, 2017. 215 p. Dissertation thesis. University of Hradec Kralove, Faculty of Education, Department of Computer Science, Faculty of Science.

The dissertation thesis researches concepts of teaching programming objects-first and objects-later. The thesis describes various theoretical bases necessary for achieving the objectives of the dissertation. Theoretical bases of thesis are programming paradigms, concepts of teaching programming and researches of concepts of teaching programming objects-first and objects-later. Researches of programming concepts by different authors are the basis of research in the dissertation. concepts by different authors are the basis of research in the dissertation.

The research itself was conducted in the form of pedagogical experiment at the Faculty of Science of the University of Hradec Králové in the academic years 2013/2014 and 2014/2015 in the course Programming 1. 28 students participated in the research in the academic year 2013/2014. 34 students participated in the research in the academic year 2014/2015.

Prior to the actual pedagogical experiment, students were divided into experimental and control group according the results of the final tests (pre-test of research itself) in the course Algorithms and data structures.

During the course of the course Programming 1 students completed separate tests (posttest, retest1 and retest 2). Learning materials were available for students in learning environment Moodle.

In research itself the results of all students' tests were processed. An anonymous questionnaire that students filled out at the end of the Programming 1 course examined the possible influence of the teacher, the sufficiency of instructional materials or preparation of students.

## **Keywords**

Concepts of teaching programming, objects-first, objects-later, programming paradigms, pedagogical experiment

# OBSAH

<b>SEZNAM POUŽITÝCH ZKRATEK, ZNAČEK, TERMÍNŮ .....</b>	<b>10</b>
<b>ÚVOD.....</b>	<b>11</b>
<b>1 VYMEZENÍ ZÁKLADNÍHO POJMU.....</b>	<b>14</b>
<b>2 CÍL DISERTAČNÍ PRÁCE.....</b>	<b>15</b>
<b>3 TEORETICKÁ VÝCHODISKA.....</b>	<b>19</b>
3.1 Paradigmata programování.....	20
3.1.1 Rozdělení paradigmat programování.....	20
3.1.2 Imperativní paradigma programování .....	22
3.1.3 Objektově orientované programování .....	24
3.1.4 Srovnání OOP a strukturovaného programování s ohledem na výuku.....	30
3.2 Koncepce výuky programování v úvodních předmětech programování .....	32
3.2.1 Koncepce výuky programování bez OOP v úvodním předmětu programování.....	34
3.2.2 Koncepce výuky programování s OOP v úvodním předmětu programování.....	43
3.3 Výhody/nevýhody OOP a strukturovaného programování v úvodních předmětech programování.....	47
3.4 Výzkumy koncepcí výuky programování objects-first vs. objects-later.....	49
3.4.1 Výzkumné otázky a hypotézy ve výzkumu koncepcí výuky programování.....	50
3.4.2 Rozdělení studentů do dvou skupin podle různých kritérií .....	50
3.4.3 Postup řešení výzkumu .....	52
3.4.4 Způsob testování studentů.....	55
3.4.5 Výsledky výzkumné studie .....	57
3.5 Zhodnocení výzkumů koncepcí výuky programování.....	58
<b>4 METODY ZPRACOVÁNÍ A ZPŮSOB ŘEŠENÍ .....</b>	<b>60</b>
4.1 Strategie výzkumu.....	60
4.2 Výzkumné metody a techniky .....	61

4.2.1	Dotazník.....	61
4.2.2	Přirozený experiment.....	62
4.2.3	Testy .....	62
4.3	Způsob měření získaných dat v testech.....	64
4.4	Možnosti ověřování studentských programátorských znalostí.....	66
4.4.1	Vyhodnocování vyučujícím .....	66
4.4.2	Automatické vyhodnocování .....	69
4.5	Metody zpracování dat v pedagogickém výzkumu.....	70
4.5.1	Wilcoxonův test.....	71
4.5.2	U-test Manna a Whitneyho .....	72
4.5.3	Studentův t-test.....	72
4.5.4	Spearmanův koeficient pořadové korelace .....	72
<b>5</b>	<b>PRŮBĚH PEDAGOGICKÉHO EXPERIMENTU .....</b>	<b>74</b>
5.1	Pedagogický experiment - pretest.....	76
5.1.1	Rozdělení studentů do skupin .....	77
5.1.2	Zvolené vývojové prostředí a programovací jazyk .....	78
5.1.3	Obsahová náplň jednotlivých vyučovaných skupin studentů .....	78
5.1.4	Testy .....	84
<b>6</b>	<b>VÝSLEDKY VLASTNÍHO VÝZKUMU .....</b>	<b>92</b>
6.1	Výsledky studentů z akademického roku 2013/2014.....	95
6.1.1	Výsledky studentů z posttestu .....	95
6.1.2	Výsledky studentů z posttestu, kteří absolvovali i retest 1 .....	100
6.1.3	Výsledky studentů z retestu 1.....	104
6.1.4	Výsledky studentů z retestu 2.....	108
6.1.5	Srovnání výsledků studentů z posttestu a z retestu 1.....	111
6.1.6	Výsledky dotazníkového šetření .....	118
6.2	Výsledky studentů z akademického roku 2014/2015.....	118
6.2.1	Výsledky studentů z posttestu .....	118
6.2.2	Výsledky studentů z posttestu, kteří absolvovali i retest 1 .....	122
6.2.3	Výsledky studentů z retestu 1.....	127
6.2.4	Výsledky studentů z retestu 2.....	132
6.2.5	Srovnání výsledků studentů z posttestu a retestu 1 .....	135

6.2.6	Výsledky dotazníkového šetření .....	142
6.3	Výsledky studentů z akademického roku 2014/2015 - bez studentů opakujících předmět.....	143
6.3.1	Výsledky studentů z posttestu.....	143
6.3.2	Výsledky studentů z posttestu, kteří absolvovali i retest 1 .....	148
6.3.3	Výsledky studentů z retestu 1.....	153
6.3.4	Výsledky studentů z retestu 2.....	158
6.3.5	Srovnání výsledků studentů z posttestu a z retestu 1.....	158
6.4	Výzkum výsledků studentů vyučovaných stejným vyučujícím.....	165
6.4.1	Výsledky studentů z posttestu.....	167
6.4.2	Výsledky studentů, kteří absolvovali i retest 1, z posttestu .....	168
6.4.3	Výsledky studentů z retestu 1.....	169
6.4.4	Výsledky studentů z posttestu bez studentů opakujících předmět Programování 1.....	171
6.4.5	Výsledky studentů z posttestu, kteří absolvovali i retest 1 bez studentů opakujících předmět Programování 1 .....	172
6.4.6	Výsledky studentů z retestu 1 bez studentů opakujících předmět Programování 1 .....	174
6.4.7	Výsledky dotazníkového šetření .....	175
<b>7</b>	<b>SHRNUTÍ VÝSLEDKŮ VLASTNÍHO VÝZKUMU .....</b>	<b>176</b>
7.1	Akademický rok 2013/2014 .....	176
7.2	Akademický rok 2014/2015 .....	179
7.2.1	Výsledky všech studentů.....	179
7.2.2	Výsledky studentů bez studentů opakujících předmět Programování 1 ....	183
7.3	Výsledky studentů, které vyučoval stejný vyučující.....	186
7.3.1	Výsledky se studenty, kteří opakovali předmět Programování 1 .....	186
7.3.2	Výsledky studentů, kteří neopakovali předmět Programování 1 .....	189
7.4	Závěrečné komplexní shrnutí výsledků vlastního výzkumu z jednotlivých akademických let.....	191
	<b>ZÁVĚR.....</b>	<b>194</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>198</b>
	<b>PUBLIKAČNÍ ČINNOST AUTORA.....</b>	<b>207</b>



<b>SEZNAM OBRÁZKŮ .....</b>	<b>209</b>
<b>SEZNAM TABULEK .....</b>	<b>210</b>
<b>PŘÍLOHY .....</b>	<b>216</b>

# SEZNAM POUŽITÝCH ZKRATEK, ZNAČEK, TERMÍNŮ

<b>Zkratka, název</b>	<b>Význam</b>
ALGDS	Algoritmy a datové struktury
ČR	Česká republika
IO	Input/Output
OO	Objektově orientovaný
OOP	Objektově orientované programování
PřF	Přírodovědecká fakulta
SP	Strukturované programování
ŠVP	Školní vzdělávací plány
UHK	Univerzita Hradec Králové

## ÚVOD

Programování se řadí mezi nejdůležitější a nejnáročnější disciplíny informatiky. Vyučuje se na některých základních, středních a vysokých školách v ČR. S programováním se zájemci nemusejí setkat pouze ve školách. Firmy vypisují pro zájemce programátorské kurzy nebo přeškolovací kurzy pro zkušené programátory. Je všeobecně známo, že „řada samouků nevyužívá nabídek škol či firem a učí se základy programování sama. Hlavně školy a programátorští samouci se obvykle potýkají s problémem, kterým stylem programování se budou zabývat. Dalšími dilematy může být výběr vhodného programovacího jazyka nebo někdy příliš nadměrná očekávání, která nebývají brzy naplněna. Především firmy, ale i některé střední a vysoké školy se snaží sledovat nejmodernější trendy. Ostatní školy na vývoj reagují pozvolna nebo jej skoro vůbec nesledují. Samouci si vyhledávají obvykle informace na internetu. Naleznou mnohdy velké množství informací. Ty ale mohou být neaktuální, chybné nebo nestrukturované. Z nich se, samouk, může naučit nevhodný způsob uvažování v programování.“

Styl programování obvykle souvisí se zvoleným programovacím jazykem. Programovacích jazyků existuje velké množství. Je takřka nemožné, aby se samouk v dostupných programovacích jazycích zorientoval. Totéž platí i pro některé učitele předmětů programování na různých školách. Odborní pracovníci ve firmách by měli mít detailní informace o výhodách nebo nevýhodách alespoň některých programovacích jazyků. S rozvojem programovacích jazyků, který přináší různá nová řešení nebo nové pohledy na stávající problematiku, souvisí i obsah náplně výuky programování. Ta se v některých případech po řadu let nemění. Týká se to hlavně výuky programování na školách. Příčin může být několik. Např. někteří učitelé nechtějí opouštět zaběhlé koleje. Častá inovace a aktualizace ŠVP není na některých středních školách možná apod.

Ve firmách bývá situace jiná než např. ve školách. Aby firma uspěla na trhu, musí sledovat a začleňovat nejnovější trendy vývoje. Tento trend se ale brzy může týkat i řady středních či vysokých škol. Konkurence škol v malém populačním vývoji potencionálních studentů je značná. Aby školy uspěly, musí zájemcům nabízet atraktivní obory, po jejichž vystudování bude mít absolvent uplatnění na trhu práce.

Zařazená novinka ve způsobu programování, ve které není uplatněn její potenciál nebo je vysvětlována v „nelogicky“ navazujících tématech, nepřináší požadovaný efekt.

Velkou problematikou, o které se v odborných kruzích intenzivně diskutuje, bývá první zvolený styl programování, respektive první programovací jazyk, se kterým se studenti seznámí. V minulosti, na různých konferencích nebo články v odborných časopisech, se hodně střetávali zastánci či odpůrci stylu strukturovaného programování a stylu OOP. S oběma styly programování se studenti seznamují na různých typech škol. Z hlediska aktuálního vývoje aplikací a vývoje softwaru je OOP klíčovou kompetencí programátorů. Strukturované programování má ovšem v určitých oblastech také svoje uplatnění. Uplatňuje se např. v programování lego mindstorms apod., kde se nemusí programovat pomocí OOP. Nejen problematikou zvolení úvodního stylu programování se řadu let zabývají odborníci a vědci z akademické sféry.

V ČR se, na některých středních školách, programuje „po staru“ pomocí strukturovaného programování. Teprve po pochopení strukturovaného programování se studenti seznámí s principy OOP, jak uvádí např. Pecinovský (2006).

Obsahová náplň předmětů programování s pořadím probíraných témat bývá také hodně diskutovaným tématem. Touto problematikou se zabývali a zabývají různí autoři řady studií a výzkumů. V konkrétním stylu programování se k dosažení daného cíle může dosáhnout i různým pořadím probíraných témat. Probíraná témata v daném stylu programování mohou být také různá. Některý tematický celek nemusí být obsažen a vyučován v jednom pořadí probíraných témat, ale v jiném pořadí probíraných témat (pro srovnání myšleno druhé pořadí probíraných témat) být může. Označení *různé pořadí probíraných témat* bylo zvoleno autorem disertační práce při vymezení dané problematiky, kterou se zabývá řada autorů. Předpokladem označení *různé pořadí probíraných témat* je i možnost výskytu nebo absence probíraných **různých** témat.

V řadě odborných publikací autoři uvádějí, definují a navzájem ztotožňují různé pojmy, které se pojí se styly programování nebo pořadím témat obsahové náplně předmětů programování.

Téma disertační práce bylo vybráno s ohledem na výše uvedené problémy, kde i strukturované konstrukce mají na začátku výuky svoje opodstatnění.

Na základě výše uvedené problematiky byly formulovány následující výzkumné otázky:

1. Jsou výsledky studentů v testech ve výuce programování ovlivněny pořadím probíraných témat dvou různých stylů programování?
2. Ovlivňuje první probírané téma daného stylu programování výsledky studentů v programování i v pozdějších lekcích v předmětech programování?
3. Liší se obsahové náplně v předmětu programování dvou různých stylů programování pouze pořadím probíraných témat?
4. Ovlivňují různé typy testování a různé typy testů výsledky studentů v testech z předmětu programování?

V rámci disertační práce bude prezentován výzkum vlivu „*dvou různých pořadí probíraných témat*“ ve výuce programování. Výzkum bude vycházet ze studií a výzkumů jiných autorů.

Disertační práce je rozdělena do několika částí.

V první části práce je vymezen základní pojem.

Ve druhé části práce jsou stanoveny cíle disertační práce.

Třetí část disertační práce obsahuje teoretická východiska současného stavu problematiky.

Ve čtvrté části práce jsou uvedeny metody zpracování pedagogického experimentu.

V páté části práce je detailně popsán průběh pedagogického experimentu.

V šesté části práce jsou prezentovány výsledky vlastního výzkumu programových (objektových a strukturovaných) konstrukcí i možných faktorů ovlivňujících výsledky studentů v testech.

V sedmé části práce je shrnutí výsledků vlastního výzkumu, které bylo detailně uvedeno v předcházející kapitole.

# 1 VYMEZENÍ ZÁKLADNÍHO POJMU

Pro zpracování disertační práce a zodpovězení výzkumných otázek je zásadní vymezení termínu „*různé pořadí probíraných témat*“ ve výuce programování. Tímto termínem je myšleno pořadí probíraných tematických okruhů (základních programátorských pojmů) v příslušném programovacím stylu programování i možnost výskytu nebo absence probíraných různých témat, jak již bylo uvedeno v Úvodu. Podle autorů, z jejichž výzkumů a vymezení pojmů čerpáme především, je tento termín popsán různě.

V odborném článku publikovaném M. Uysalem viz (Uysal, 2012) autor slovní spojení „*různé pořadí probíraných témat*“ nazývá „*metodou*“. V jeho studii uvádí kromě pojmu „*metoda*“ ještě jiný název, jímž je „*instructional approaches*“.

V článku publikovaném A. Ehlertem a C. Schultem viz (Ehlert a Schulte, 2007) autoři slovní spojení „*různé pořadí probíraných témat*“ nazývají pojmy „*přístup*“ nebo „*strategie*“. Autoři J. Sajaniemi a Ch. Hu v jejich článku Sajaniemi a Hu (2006), také nazývají slovní spojení „*různé pořadí probíraných témat*“ pojmem „*přístup*“. Společnosti ACM a IEEE-CS<sup>1</sup>(2001) v jejich učebních plánech nazývají slovní spojení „*různé pořadí probíraných témat*“ pojmem „*přístup*“ nebo „*model*“.

Rudolf Pecinovský v článku Pecinovský (2006) nazývá slovní spojení pojmy „*metodika výuky*“, „*přístup*“ nebo „*koncepce*“.

Rostislav Fojtík v článku Fojtík (2013) nazývá slovní spojení pojmem „*přístup*“.

**„*Různé pořadí probíraných témat*“ ve výuce programování bude v disertační práci označeno jako koncepce výuky programování.**

**Z předcházejících kapitol je zřejmé, že pro „*různé pořadí probíraných témat*“ používají autoři různá označení. V disertační práci se budeme držet pojmu koncepce výuky programování.**

---

<sup>1</sup> Pro lepší srozumitelnost textu budou u citovaných nebo parafrázovaných informací daných společností (z jejich studijních plánů jak z roku 2001, tak i z roku 2013) používány jejich zkratky, tj. společnost ASSOCIATION FOR COMPUTING MACHINERY bude označena zkratkou ACM a společnost IEEE COMPUTER SOCIETY bude označena zkratkou IEEE-CS.

## 2 CÍL DISERTAČNÍ PRÁCE

Hlavním cílem disertační práce je zjistit, jak různé pořadí probíraných témat (koncepce výuky programování) ovlivňuje výsledky studentů 1. ročníku oboru učitelství informatiky na PřF UHK v předmětu Programování 1 v programovacím jazyku C#.

K dosažení hlavního cíle vedou níže uvedené dílčí cíle:

- Vytvoření obsahové náplně výuky programování v programovacím jazyku C# pomocí koncepce objects-later
- Vytvoření obsahové náplně výuky programování v programovacím jazyku C# pomocí koncepce objects-first
- Vytvoření testů a jejich validace
- Komparace koncepcí výuky programování objects-later a objects-first a jejich základních programových konstrukcí

Na základě stanovených dílčích cílů disertační práce byla zformulovaná následující obecná nulová a alternativní hypotéza:

### **Nulová hypotéza $H_0$ :**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů jako studenti koncepce výuky programování objects-later.

### **Alternativní hypotéza $H_A$ :**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů než studenti koncepce výuky programování objects-later.

Výše uvedené hypotézy jsou zformulovány obecně. Tyto hypotézy se dále rozpadají na soubor 13 konkrétních hypotéz, ve kterých je podrobně uvedeno, o které výsledky z testů se jedná.

### **1. Nulová hypotéza $H_{10}$ :**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů jako studenti koncepce výuky programování objects-later.

### **Alternativní hypotéza $H_{1A}$ :**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů než studenti koncepce výuky programování objects-later.

**2. Nulová hypotéza H2<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické části jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H2<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické části než studenti koncepce výuky programování objects-later.

**3. Nulová hypotéza H3<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z teoretické části jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H3<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z teoretické části než studenti koncepce výuky programování objects-later.

**4. Nulová hypotéza H4<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické objektové části jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H4<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické objektové části než studenti koncepce výuky programování objects-later.

**5. Nulová hypotéza H5<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické strukturované části jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H5<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické strukturované části než studenti koncepce výuky programování objects-later.



**6. Nulová hypotéza H<sub>60</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z teoretické části Doplnění kódu jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H<sub>6A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z teoretické části Doplnění kódu než studenti koncepce výuky programování objects-later.

**7. Nulová hypotéza H<sub>70</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z teoretické části Interaktivní prvky jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H<sub>7A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z teoretické části Interaktivní prvky než studenti koncepce výuky programování objects-later.

**8. Nulová hypotéza H<sub>80</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z teoretické části Doplnění hodnot proměnných jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H<sub>8A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z teoretické části Doplnění hodnot proměnných než studenti koncepce výuky programování objects-later.

**9. Nulová hypotéza H<sub>90</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické objektové části Definice konstruktoru a datových položek jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H<sub>9A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické objektové části Definice konstruktoru a datových položek než studenti koncepce výuky programování objects-later.

**10. Nulová hypotéza H10<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické objektové části Metody a vlastnosti jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H10<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické objektové části Metody a vlastnosti než studenti koncepce výuky programování objects-later.

**11. Nulová hypotéza H11<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické objektové části Instance a volání metod jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H11<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické objektové části Instance a volání metod než studenti koncepce výuky programování objects-later.

**12. Nulová hypotéza H12<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické strukturované části Naplnění a výpis posloupnosti jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H12<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické strukturované části Naplnění a výpis posloupnosti než studenti koncepce výuky programování objects-later.

**13. Nulová hypotéza H13<sub>0</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou stejných výsledků z testů z praktické strukturované části Návrh algoritmů metod jako studenti koncepce výuky programování objects-later.

**Alternativní hypotéza H13<sub>A</sub>:**

Studenti koncepce výuky programování objects-first dosáhnou odlišných výsledků z testů z praktické strukturované části Návrh algoritmů metod než studenti koncepce výuky programování objects-later.

### 3 TEORETICKÁ VÝCHODISKA

Vývoj programování, respektive vývoj programovacích jazyků souvisí s vývojem počítačů. Je všeobecně známo, „že programovací jazyky se dají rozdělit podle určitých kritérií. Mezi nejčastější dělení programovacích jazyků je jejich rozdělení na nižší programovací jazyky a vyšší programovací jazyky. Podle způsobu překladu se dají programovací jazyky rozdělit na interpretované, kompilované nebo hybridní. Další členění programovacích jazyků souvisí s paradigmatem programování, které podporují. Vývoj paradigmat programování také úzce souvisí s vývojem programovacích jazyků a počítačů. Paradigmata programování mají zásadní vliv na tvorbu programů, softwaru, respektive vývoj informačního systému“.

Ve školství je důležité, s jakým paradigmatem programování, respektive programovacím jazykem, se studenti nejdříve seznámí v jejich prvním předmětu programování. To má zásadní vliv na pochopení, jakým způsobem budou studenti vytvářet programy. Výběr vhodného paradigmatu ve výuce programování má zásadní vliv i na pozdější uplatnění studentů v praxi u různých firem. Bývá prospěchem věci, když se studenti ve studiu seznámí s více paradigmaty programování, popřípadě s koncepcemi výuky programování, které různá paradigmata programování využívají.

Kapitola 3 je rozdělena na jednotlivé podkapitoly, ve kterých jsou definovány, vymezeny a formulovány základní pojmy týkající se paradigmat programování a koncepcí výuky programování. Jsou zde také uvedeny i některé názory autorů na dané koncepcí výuky programování a paradigmata programování.

### 3.1 Paradigmata programování

Bieliková a Návrat (2009, s. 1) uvádějí, že *„jednotlivé druhy, způsoby či přístupy k programování označujeme jako paradigma programování“*. Autoři dále k paradigmatu uvádějí, že *„jsou to názory, teorie, metody, metodiky a pod., které se v dané oblasti uznávají“*.

Paradigma je podle Průchy, Walterové a Mareše (2003, s. 152) definováno jako *„zásadní obraz předmětu určité vědy a jejích výzkumných přístupů. Představuje nejširší konsenzus mezi vědci pracujícími v určité oblasti vědeckého výzkumu, odlišuje členy různých vědeckých komunit, popř. subkomunit. Definuje, co má být studováno, jaké otázky a jak mají být kladeny, podle jakých pravidel mají být interpretovány získané výsledky“*.

Jinou definici paradigmatu programování uvádějí Konečný a Vychodil (2008, s. 16), že *„paradigma je de facto styl, kterým lze programovat v jakémkoliv jazyku, který jej podporuje (umožňuje)“*.

Vujošević-Janičić a Tošić (2008, s. 63) uvádějí, že *„paradigma ve výpočetní technice může být definováno jako ucelený soubor metod, které byly nalezené jako efektivní pro zvládnutí určitého typu problému. Paradigma lze popsat pomocí jednoduchých základních principů“*.

Rýdlo (2012, s. 8) uvádí, že *„paradigma je souhrn základních postupů jak formulovat a řešit problém a určuje tedy způsob nebo strukturu, jakou programátor tvoří kód“*.

#### 3.1.1 Rozdělení paradigmat programování

Vujošević-Janičić a Tošić (2008, s. 67) uvádějí čtyři hlavní programovací paradigmatata: *„imperativní paradigma, objektově orientované paradigma, funkcionální paradigma a logické paradigma“*. Stejně rozdělení paradigmat programování uvádí i Bolshaková (2005, s. 286-287). Podobné rozdělení paradigmat programování uvádějí i Bieliková a Návrat (2009), Rýdlo (2012, s. 9) nebo Dohnal (2009, s. 27), kde autoři uvádějí místo imperativního paradigmatu programování jiné označení.

Bieliková a Návrat (2009, s. 1) uvádějí procedurální paradigma programování. Dohnal (2009, s. 27-28) uvádí strukturované programování. Rýdlo (2012, s. 9) uvádí imperativní strukturované.

Konečný a Vychodil (2008, s. 16-17) uvádějí celkem pět paradigmat programování: procedurální, funkcionální, logické, objektově orientované a paralelní paradigma programování. Autoři dále na straně 16 uvádějí, že „*procedurální paradigma je jedním ze dvou nejstarších paradigmat programování, někdy též bývá nazýváno paradigmatem imperativním*“.

K procedurálnímu programování Bieliková a Návrat (2009, s. 1-2) uvádějí „...*metody programování (např. strukturované, modulární)*...“.

„*Z hlediska stylu programování ještě procedurální paradigma programování můžeme jemněji rozdělit na: naivní..., nestrukturované... a strukturované...*“ (Konečný a Vychodil, 2008, s. 16).

Gewali a Minor (2006) uvádějí, že „*procedurální programování, funkcionální programování a objektově orientované programování jsou nejpoužívanější programovací paradigmatata v institucích vyššího vzdělání a ve vysoce technicky vyspělém průmyslu*“.

V literatuře se ještě vyskytuje pojem aspektově orientované programování. Zikmund (2010, s. 1) uvádí, že aspektově orientované programování „...*se snaží doplnit objektově orientované programování o prvky, které napomáhají větší přehlednosti kódu a efektivnosti práce*“. Kiczales et al. (1997) nazývají aspektově orientované programování programovací technikou.

S ohledem na výše uvedené definice a formulace paradigmat programování se paradigmatata programování mohou rozdělit následujícím způsobem:

- Imperativní (procedurální) paradigma programování
  - Naivní programování
  - Nestrukturované programování
  - Strukturované programování
  - Modulární programování
- Objektově orientované paradigma programování
  - Aspektově orientované programování
- Funkcionální paradigma programování

- Logické paradigma programování
- Paralelní paradigma programování

S ohledem na zaměření výzkumu disertační práce bude blíže uvedeno imperativní paradigma programování a objektově orientované programování. V imperativním paradigmatu programování bude uvedeno strukturované programování.

### 3.1.2 Imperativní paradigma programování

*„Programy napsané v procedurálním<sup>2</sup> stylu jsou tvořeny sekvencemi příkazů, které jsou postupně vykonávány“* (Konečný a Vychodil, 2008, s. 16).

Při vykonávání programu, který obsahuje např. sekvenci, větvení apod., se vykonávají příkazy, tj. pracuje se s proměnnými, do kterých se ukládají a dále využívají hodnoty. Základním stavebním kamenem tohoto paradigmatu je symbol přiřazení. (Konečný a Vychodil, 2008, s. 16)

#### Strukturované programování

Jensen (1981, s. 31) uvádí, že *„strukturované programování je způsob organizování a kódování programu, které dělají programy snadno srozumitelnými a upravitelnými“*.

Hořejš (1984, s. 2) definuje strukturované programování, tak, že *„SP zahrnuje všechny metodické postupy a odpovídající jazykové prostředky, které ulehčují proces konstrukce spolehlivých, přehledných a tedy modifikovatelných programů“*.

Dohnal (2009, s. 28) cituje definici strukturovaného programování Krčka a Kremla (1993), že *„strukturované programování je metoda systematického a metodického přístupu k vytváření takové řídicí struktury algoritmu, která by byla jasným obrazem řešeného problému a zvolené metody řešení a která by byla tvořena jen z několika málo typů jednoduchých stavebních konstrukcí“*.

Podle Konečného a Vychodila (2008, s. 16) *„strukturované procedurální paradigma nahrazuje příkazy skoku pomocí podmíněných skoků typu „opakuj~dokud platí podmínka“ a jiných strukturovaných konstrukcí, které se do sebe vnořují“*.

---

<sup>2</sup> Autoři Konečný a Vychodil (2008, s. 16) uvádějí, že *„Procedurální paradigma..., někdy též bývá nazýváno paradigmatem imperativním“*.

Podle výše uvedených definic a informací je zřejmé, že programy naprogramované pomocí strukturovaného programování musejí být přehledné a snadno upravitelné. V programech by měly být použity pouze základní strukturované konstrukce, které Drbal (1992, s. 11) nazývá konstrukty.

Drbal (1992, s. 11) uvádí, že „*pan Dijkstra formuloval tyto zásady pro tvorbu dobře strukturovaných programů:*

- *program vzniká hierarchickým rozkladem zadané úlohy (tj. shora-dolů),*
- *pro rozklad úlohy v podúlohy jsou použity co nejjednodušší konstrukty (konstrukt = provedení rozkladu),*
- *použité konstrukty musí mít jeden vstupní kontext a jeden výstupní kontext“.*

Strukturované programování je člověku blízké. Jednotlivé kroky programu jsou vykonávány postupně za sebou. Posloupnost příkazů je důležitá. Mezi základní příkazy strukturovaného programování patří: selekce, podmíněný příkaz (větvení) a iterace (opakování nebo cyklus). Program postupně vykonává všechny příkazy od začátku programu do konce programu. Návaznost jednotlivých kroků není náhodná. Tu při návrhu programu určuje programátor. Posloupnost jednotlivých kroků je ovlivněna pomocí základních příkazů strukturovaného programování. (Chen, 2004, s. 71)

O tom, jaká část programu se provede, záleží na hodnotách v paměťových místech. Tato paměťová místa se nazývají **proměnné**. Jejich hodnota bývá určující pro provedení určité posloupnosti příkazů. Daná proměnná může nabývat v průběhu vykonávání programu různých hodnot. Různé posloupnosti příkazů mohou ovlivnit různé hodnoty dané proměnné. (Vujošević-Janičić a Tošić, 2008, s. 69)

Opakující se kód v programu, respektive posloupnost stejných příkazů, se může označit názvem, který vystihuje jeho funkcionalitu, respektive funkcionalitu posloupnosti stejných příkazů. Pojmenované posloupnosti příkazů se pak nazývají podprogramy. Podprogramy se dají rozdělit, např. podle skriptovacího programovacího jazyku nebo programovacího jazyku, který podporuje strukturované programování, na procedury a funkce. V souvislosti s OOP se podprogramy označují

jako metody. Pomocí podprogramů se dají složitější úlohy rozdělit na dílčí podproblémy. (Vujošević-Janičić a Tošić, 2008, s. 69)

Dohnal (2009, s. 28) uvádí, že *„první zásadou dobře strukturovaných programů je nahrazení řešení jednoho složitého úkolu množstvím řešení jednoduchých, která jako celek dávají řešení úkolu složitého. Toto nahrazení probíhá ve třech fázích“*. Autor dále uvádí, že *„první fází je rozklad složitého úkolu na řešitelné dílčí úlohy (tj. analýza), dalším krokem je řešení jednotlivých podúkolů získaných při analýze počátečního problému a konečnou fází je spojení vyřešených dílčích úkolů v celkové řešení (tj. syntéza)“*.

### **Výhody strukturovaného programování**

Chen (2004, s. 71) uvádí, že *„strukturovaný jazyk obvykle vytváří programy, které rychle běží a využívají efektivně systémové zdroje“*.

Asagba a Ogheneovo (2008, s. 41) uvádějí, že strukturované programování *„je způsob kódování a organizace programů, co je dělá dobře srozumitelnými, snadno se ladí a snadno se upravují“*.

### **Nevýhody strukturovaného programování**

Chen (2004, s. 71) uvádí, že mezi nevýhody strukturovaného programování patří *„...programátorovo vidění problémů jako sérii určitých kroků, nicméně některé problémy si lze lépe představit jako interaktivní objekty nebo jako provázaná slova, koncepty a myšlenky“*.

Podle Pecinovského (2008) je výuka strukturovaného programování *„...kontraproduktivní, protože se studenti nejprve učí způsobu uvažování, který pak musejí při přechodu na skutečné objektové programování opustit“*.

### **3.1.3 Objektově orientované programování**

OOP může být různými autory chápáno rozdílně. Někteří autoři považují OOP za nadstavbu strukturovaného programování. Jiní považují OOP za rozdílný způsob programování, který řeší nedostatky strukturovaného programování. V kapitole je nejdříve definováno OOP s jeho základními vlastnostmi a pojmy. Dále jsou v kapitole uvedeny výhody a nevýhody OOP. Výhody a nevýhody OOP lze zhodnotit z několika hledisek. Jedním hlediskem je zhodnocení výhod a nevýhod vzhledem k výuce OOP na školách.



## Definice OOP

Drbal (1994, s. 3) uvádí, že „objektově programovat znamená používat objekty jako stavební kameny, ze kterých se staví program. Někaké dědění zde nemá žádný smysl. Daleko důležitější vztahy zde jsou:

- obsahuje,
- vlastní,
- používá,

a vztahy inverzní - je obsažen, je vlastněn, je používán“.

Honzík (1990, s. 1) uvádí „...náhled na OOP jako na soubor principů, metod a nástrojů, které mohou vést k tvorbě aplikačního programového díla, jehož kvality budou mnohem vyšší, než jsou kvality u současné produkce<sup>3</sup>“.

Pecinovský (2009, s. 37) definuje OOP tak, že „objektově orientované programování vychází z myšlenky, že všechny programy, které vytváříme, jsou simulací buď skutečného nebo nějakého námi vymyšleného virtuálního světa“.

Chen (2004, s. 71) definuje OOP tak, že „...formuluje program jako sérii objektů a metod, jež se vzájemně ovlivňují, aby splnily specifickou úlohu“.

I když OOP mnozí autoři vymezují pomocí základních pojmů, které budou uvedeny dále v kapitole, Čada (2009, s. 20) uvádí, že „...ani všechny dohromady nejsou tím základním a hlavním principem, na němž je idea objektového programování založena...“ Autor dále uvádí, že základním principem je „volná síť vzájemně komunikujících objektů“.

Asagba a Ogheneovo (2008, s. 42) uvádějí, že OOP „...používá pro tvorbu aplikací a počítačových programů tzv. „objekty“ a interakce mezi nimi“.

Vujošević-Janičić a Tošić (2008, s. 71) také uvádějí, že „...objektově orientované programování využívá předávání zpráv k zachycení působení mezi objekty“.

Z výše uvedených definic a formulací OOP je zřejmé, že mnozí autoři definují OOP jako vzájemně komunikující objekty. Komunikace objektů probíhá přes volání metod. V definicích OOP autoři nezmiňují základní pojmy, které jsou

---

<sup>3</sup> Autor má na mysli nejspíše tehdejší strukturované programování

s OOP úzce spjaty. Mezi tyto základní pojmy patří především princip zapouzdření, polymorfismus a dědičnost. Jiní autoři formulují a definují OOP v návaznosti na imperativní programování.

Vujošević-Janičić a Tošić (2008, s. 71) uvádějí, že „*objektově orientované programování je zobecněním imperativního programování*“.

Lister et al. (2006, s. 147) citují Burtona a Bruhna (2003) „*...ačkoliv OOP bezesporu zastupuje nové paradigma, v žádném případě nenahrazuje starší paradigma... spíše je jeho doplněním. Protože paradigma ve svém vlastním algoritmickém myšlení (spolu s procedurálním programováním) potřebuje být nejprve pochopeno odděleně od OOP. Studenti potřebují znát, jak OOP zapadá do většího obrazu a úvodní kurz by měl toto jistě umožnit... Všem studentům by měli být dány pevné základy algoritmů, strukturálního programování, procedur a uznání tohoto historického vývoje, před přechodem k objektově orientovanému programování*“.

Gewali a Minor (2006) uvádějí, že „*novější organizační abstrakce, která se stala součástí imperativního programování je objektově orientované paradigma*“.

Donchev a Todorova (2008, s. 159) uvádějí, že „*výuka<sup>4</sup> zahrnuje všechny důležité koncepty jako dobře strukturované programování, modularita, návrh programu a také techniky řešení problémů...*“.

Klimeš et al. (2008, s. 258) uvádějí, že „*každé, tj. i objektově orientované, programování v sobě na úrovni programování základních operací nese vlastnosti strukturovaného programování a ani to, že při tvorbě aplikací využíváme komponenty, neznamená, že programujeme objektově*“.

Donchev a Todorova (2008, s. 162) uvádějí další autory, kteří definují OOP z pohledu programovacího jazyka Smalltalk. Daná vymezení jsou ale staršího data.

Definice a formulace OOP jednotlivých autorů, uvedených v kapitole výše, jsou odlišné. Nejčastější definice a formulace OOP lze shrnout následovně: Autoři formulují OOP pomocí základních pojmů, jako je např. dědičnost viz Donchev a Todorova (2008). Jiní formulují OOP jako základ zobecnění imperativního paradigmatu programování. Řada autorů vymezuje a definuje OOP jako vzájemně komunikující objekty pomocí metod.

---

<sup>4</sup> Autoři mají na mysli výuku OOP

Dle autora disertační práce není formulace OOP, na základě imperativního paradigmatu programování, příliš vhodná. Jedná se o odlišný způsob návrhu programu. Vzájemné porovnání OOP a imperativního paradigmatu programování je uvedeno v kapitole 3.3. Obě dvě paradigmatu programování, jak OOP, tak imperativní paradigma, se mohou při návrhu programů vhodně doplňovat a mají některé společné vlastnosti. Mezi společné vlastnosti obou paradigmat programování patří např. přehlednost a modifikovatelnost programů, využívání podprogramů apod.

Využívání základních konstrukcí OOP při návrzích programů je důležité. Jejich použití může mít i některé negativní důsledky při tvorbě programů, které jsou uvedeny ke konci této podkapitoly.

Formulace OOP pomocí základních vlastností OOP může být zavádějící. Autoři mohou za klíčovou konstrukci považovat různé vlastnosti a podle nich definovat OOP rozdílným způsobem.

Definice OOP jako vzájemná komunikace objektů s voláním metod se jeví nejpřijatelněji, protože tato formulace OOP je nezpochybnitelná. Komunikace mezi objekty s voláním metod může být v některých programovacích jazycích realizována různým způsobem, ale princip komunikace zůstává stejný.

*„...mnozí programátoři považují za objektový jakýkoliv program naprogramovaný v jazyce, který objektově orientované programování podporuje. Neuvědomují si, že program nedělá objektově orientovaným používání tříd a dědičnosti, ale že OOP vyžaduje výrazně odlišný přístup k řešení problému“* (Pecinovský, 2008).

Dále v kapitole budou uvedeny základní vlastnosti a pojmy OOP.

### **Základní vlastnosti OOP**

Mezi základní vlastnosti OOP patří princip dědičnosti, polymorfismu a zapouzdření (seřazeno abecedně).

#### **Dědičnost**

Asagba a Ogheneovo (2008, s. 44) uvádějí u dědičnosti, že *„je to schopnost nějaké stávající třídy vytvořit třídu novou. Tato stávající třída se nazývá základní třída a nově vzniklé třídy se nazývají odvozené třídy. Odvozená třída dědí všechny vlastnosti základní třídy“*.

## **Polymorfismus**

Asagba a Ogheneovo (2008, s. 44) uvádějí, že „*polymorfismus je mechanismus umožňující objektům různých typů reagovat různým způsobem na volání stejné funkce*“.

## **Zapouzdření**

Asagba a Ogheneovo (2008, s. 44) uvádějí, že „*zapouzdření je schopnost uzavřít kódy a data a skrýt je před okolím (zabránit kontaktu), takže každý, kdo se k nim chce dostat, musí použít připojený kód*“.

Předešlá formulace pojmu zapouzdření je uváděna i jinými autory, popřípadě v jiné modifikaci např. Čada (2009, s. 32-33). Je všeobecně známo, že „*okolní objekty (entity) by neměly vědět, jak je daný objekt naprogramován. Komunikace mezi objekty a získávání informací o daném objektu probíhá přes volání metod*“.

## **Základní pojmy OOP**

Mezi základní pojmy OOP patří pojmy jako např. instance třídy, konstruktor, metoda, návrhové vzory, objekt, objektový datový typ, primitivní datový typ, třída, vlastnost apod. (seřazeno abecedně).

### **Instance třídy**

Podle Čady (2009, s. 27) „*instance je objekt, který není třídou (ale sám je od nějaké třídy odvozen)*“.

### **Konstruktor**

Je všeobecně známo, že „*konstruktor je speciální metoda, která slouží k vytvoření instance*“.

Schildt (2012, s. 149) uvádí u programovacího jazyku Java 7 (ale platí např. i pro programovací jazyk C#), že „*konstruktor inicializuje objekt při jeho vytvoření. Má stejný název jako jeho třída a je syntakticky podobný metodě. Konstruktory však nemají žádný explicitní návratový typ*“.

### **Metoda**

Pecinovský (2009, s. 53) uvádí, že „*metoda je část programu, kterou instance spustí v reakci na obdržení zprávy*“.

## Návrhové vzory

Reynolds-Haertle (2002, s. 413) uvádí, že „návrhový vzor popisuje určitou obecnou úlohu nacházející se v objektově orientovaném systému a poskytuje řešení této úlohy“.

### Objekt

Asagba a Ogheneovo (2008, s. 43.) vymezují objekt jako „...druh modulu obsahující data a podprogramy. Objekt je samostatná jednotka, která má určitý vnitřní stav (data, která obsahuje) a která může reagovat na zprávy (volání jeho podprogramů)“.

### Objektový datový typ

„**Objektové datové typy** jsou nám již dobře známé třídy. Těch bývá v programu definováno většinou mnohem víc“ (Pecinovský, 2009, s. 64).

Autor dále uvádí, že „objektové datové typy (tj. třídy) si může každý programátor definovat sám“.

Autor uvádí objektové datové typy v souvislosti s programovacím jazykem Java. Další literatura může objektové datové typy označovat jiným názvoslovím v souvislosti s daným programovacím jazykem.

### Primitivní datový typ

Pecinovský (2009, s. 64) uvádí, že „**primitivní datové typy**, mezi něž patří např. čísla, jsou zabudovány hluboko v jazyku a jejich chování je pevně dané“.

I když autor myslí především primitivní datové typy v programovacím jazyku Java, tuto formulaci lze aplikovat i na jiné programovací jazyky.

V různých programovacích jazycích mohou být u jednoho datového typu, např. celého čísla, další možnosti jemnějšího rozdělení, např. podle rozsahu, znaménka apod.

### Třída

Podle Drbala (1996, s. 2) je „**třída** identifikovatelný typ entity nebo konceptu v daném prostředí. Třída je skupina datových proměnných a chování (operace a funkce), což je sdíleno instancemi toho typu“.

Asagba a Ogheneovo (2008, s. 43) uvádějí, že „objekty lze deklarovat pomocí vytvoření vzoru pro lokální stav a metody. Tento vzor se nazývá třída a v zásadě se jedná o datový typ“.

## **Vlastnost**

Jak uvádí Reynolds-Haertle (2002, s. 23) u vlastností, tak „jsou to charakteristiky neboli kvantitativní znaky, jimiž se objekt vyznačuje...“.

## **Výhody OOP z hlediska výuky OOP**

Mezi výhody OOP Drbal (1996, s. 2) uvádí „...přísné oddělení částí programu (vnitřku objektu od ostatního programu) a nutnost jednoznačně definovat komunikační protokol mezi objektem a jeho okolím“.

Podle Pecinovského (2008) „OOP přineslo do programování další výrazné zmenšení sémantické mezery mezi způsobem popisu problému běžným člověkem a způsobem jeho popisu v analýze a následně v programu“.

Podle Čady (2009, s. 20) „...objektové programování je přímým modelem reálného světa; zdá se proto celkem pravděpodobné, že nelze vymyslet nic principiálně lepšího“.

## **Nevýhody OOP z hlediska výuky OOP**

Kölling (1999) uvádí, že „v našem pohledu to není v zásadě objektová orientace, která způsobuje problémy, ale nástroje dostupné učít to“.

Mezi nevýhody OOP je uváděná vysoká míra abstrakce, např. viz Donchev a Todorova (2008, s. 160).

Autoři dále na stranách 165 a 166 uvádějí (vybrány pouze některé) „...následující potíže při učení se OOP:

- *Neporozumění rozdílu mezi třídou a objektem a vztahu mezi nimi.*
- *Potíže při budování a pochopení vztahů mezi třídami a jejich vzájemným ovlivňováním.*
- *Studenti píšou kód s nízkým stupněm opakovatelné použitelnosti“.*

### **3.1.4 Srovnání OOP a strukturovaného programování s ohledem na výuku**

Pecinovský (2007b) uvádí: „když už se žáci naučí programovat strukturovaně, odmítají následně změnit způsob svého uvažování. Jejich programy proto často nejsou objektově orientovanými programy, ale pouze strukturovanými programy používajícími třídy“.

Kölling (1999, s. 1) také vidí velké problémy při přechodu ze strukturovaného programování na objektově orientované programování. Podle něj „se učení programování v objektově orientovaném stylu jeví jako velmi obtížné, když je používáno po procedurálním stylu“.

Autoři Lister et al. (2006, s. 149) se zabývají tvrzením, že „pro studenty, kteří se učí nejprve imperativně je OO programování obtížnější“.

Dále autoři uvádějí tvrzení některých účastníků, „...že přechod byl přirozeně obtížným posunem paradigmat...“:

- *Jako OO programátor MYSLÍTE jinak ... než jako procedurální programátor. (příspěvek 90)*
- *...přejít z OO na algoritmické/procedurální je snazší než jinou cestou...Zdá se snazší provést změnu, když paradigmata nesdílejí žádné rysy, než když ony některé rysy sdílejí. (příspěvek 28)“.*

Autoři také uvádějí tvrzení některých účastníků, že „...přechod je méně obtížný při vhodném vyučovacím přístupu“:

- *Změna paradigmat není snadná a zabere čas... S vhodnou oporou a jasným záměrem na změnu paradigma...(to) nemusí být tak obtížné jak se zdá. (příspěvek 68)*
- *...U nás je první semestrální kurz imperativní ... Naše úroveň uchování je dobrá a nevšimli jsme si žádných problémů se studenty, kteří by nepochopili objekty nebo objektově orientované programování. (příspěvek 34)“.*

Dále autoři „...nenalezli empirické studie spojené s potížností přechodu z imperativního na objektově orientované programování...“.

Autoři také dále uvádějí, že „jsme nenalezli důkazy ani v literatuře věnující se obrácenému postupu paradigmatu, z učení objects-first na učení imperativní programování“.

### 3.2 Koncepce výuky programování v úvodních předmětech programování

Ve výuce programování je třeba řešit problém s výběrem prvního paradigmatu programování – imperativní (strukturované) nebo OO, popřípadě jiným paradigmatem programování.

Při volbě paradigmatu programování musí vyučující zvolit vhodnou koncepci výuky programování<sup>5</sup> v daném paradigmatu. Společnosti ACM a IEEE-CS (2001, s. 24) definují šest koncepcí výuky programování v úvodních předmětech programování.

Z toho tři koncepce označují jako „*The programming-first implementations*“. Mezi ně zařazují následující koncepce výuky programování:

- imperative-first
- objects-first
- functional-first

Dále tři koncepce výuky programování označují jako „*alternativní*“. Mezi ně zařazují následující koncepce výuky programování:

- breadth-first
- algorithms-first
- hardware-first

Vujošević-Janičić a Tošić (2008, s. 68) uvádějí další dvě koncepce výuky úvodních programátorských kurzů:

- logical-first a concepts-first

Sajaniemi a Hu (2006) vymezují svou nově navrženou koncepci výuky programování, kterou autoři nazvali variables-first.

Pecinovský (Pecinovský, 2008; Pecinovský, 2013) vymezuje další dvě koncepce výuky úvodních programátorských kurzů:

- Design Patterns First a Architecture First

Rýdlo (2012, s. 12) uvádí koncepci výuky programování model-first.

---

<sup>5</sup> Jak již bylo uvedeno v kapitole 1, pojem koncepce výuky programování, jak jej chápeme v této práci, se v publikacích označuje různými termíny. Pro lepší srozumitelnost textu bude u citovaných nebo parafrázovaných informací používáno našeho označení: koncepce výuky programování.



Výše uvedené koncepce výuky programování jsou doporučované různými autory v úvodních předmětech programování. Řada z daných koncepcí výuky programování předpokládá (z kontextu jejich vymezení od výše uvedených autorů) výuku více paradigmat programování. Aby se mohly dané koncepce výuky programování nějakým způsobem kategorizovat, je důležité vzít v potaz, jaké paradigma programování bude navazovat na úvodní paradigma v dané koncepci výuky programování. U některých koncepcí výuky programování je další návaznost zřejmá a vyplývá z jejich formulací, u některých dalších návaznost dalšího paradigmatu programování zřejmá není. Předpokladem je, že se studenti na různých typech škol seznámí s více paradigmaty programování.

Je všeobecně známo, že „*OO*P patří mezi velice oblíbené a hojně využívané paradigma programování“. Podle Kravala (1998) nebo Pecinovského, viz např. Pecinovský (Pecinovský, 2007b; Pecinovský, 2008) aj., má *OO*P řadu výhod. Řada vyučujících i autorů, např. Johnson a Moses (2008) nebo Ehlert a Schulte (2009), se zabývají problémem, kdy toto paradigma programování, respektive jeho základní vlastnosti a pojmy, zařadit do výuky programování. Někteří autoři, např. Pecinovský (2006) nebo Kölling (1999), preferují jeho zařazení do úvodního předmětu programování. Jiní autoři, např. Donchev a Todorova (2008), preferují jeho zařazení až do navazujících předmětů programování.

Právě podle cíle vyučujících zařadit *OO*P do výuky programování (i s myšlenkou seznámení studentů s více paradigmaty ve výuce programování) lze koncepce výuky programování kategorizovat (rozdělit) na ty, které preferují *OO*P v úvodním předmětu programování a na ty, které preferují jiný styl programování v úvodním předmětu programování.

V nových vzdělávacích plánech společností ACM a IEEE-CS (2013, s. 39) vzhledem k vývoji počítačové vědy (z toho vyplývajícím používáním nových a moderních technologií), oproti jejich doporučením úvodních šesti koncepcí uvedených v ACM a IEEE-CS (2001) a výše v disertační práci, uvádějí komplexnější a rozmanitější rozdělení koncepcí v úvodních předmětech programování.

Společnosti ACM a IEEE-CS (2013, s. 13) uvádějí, že „...*my identifikujeme a popíšeme stávající úspěšné kurzy a učební plány abychom ukázali, jak významné znalosti jednotek jsou řešeny a začleněny v současných programech*“.

Společnosti ACM a IEEE-CS (2013, s. 14) kategorizují komplexnější a rozmanitější okruhy „do sady 18 znalostních oblastí“.

Dané společnosti také dále uvádějí na straně 39, že „...ne všechno relevantní pro počítačového vědce (programování, softwarové procesy, algoritmy, abstrakce, výkon, bezpečnost, profesionalita, atd.) lze vyučovat od prvního dne“.

Vzhledem k dalším informacím uvedených v dokumentu společností ACM a IEEE-CS (2013) i existencí výzkumů autorů různých koncepcí výuky programování, uvedené koncepce výuky programování na začátku kapitoly jsou stále vyučované na řadě škol. Předpoklad vyučování více paradigmat programování v průběhu studia studentů navíc umožňuje do učebních osnov zahrnout např. programování pomocí skriptovacích programovacích jazyků i programování aplikací pro přenositelná zařízení nebo programování robotů v robotice. Programování může být komplexnější a může navazovat právě na zvolenou úvodní koncepci výuky programování a tu dále rozšiřovat. Obsah vyučovaných předmětů by měl být uzpůsoben danému oboru, který studenti studují. Jiná obsahová náplň předmětů programování je např. v oborech informačních technologií, strojírenství apod. nebo na konkrétním oboru s různými zaměřenými. Na různých oborech nebo na oborech s jiným zaměřením, např. v předmětech programování, jsou mnohdy požadavky na pořadí probíraných témat ve výuce odlišné. (ACM a IEEE-CS, 2013, s. 39)

### **3.2.1 Koncepce výuky programování bez OOP v úvodním předmětu programování**

Mezi koncepce výuky programování, které preferují jiný styl programování, než je OOP, se dají zařadit následující koncepce výuky programování (seřazeny abecedně):

- algorithms-first
- breadth-first
- concepts-first
- functional-first
- hardware-first
- imperative-first
- logical-first

- variables-first

### **Algorithms-first**

Společnosti ACM a IEEE-CS (2001, s. 32) uvádějí, že „v této koncepci výuky programování jsou základní pojmy počítačové vědy uvedené pomocí pseudokódu namísto spustitelného jazyka“.

Programovací jazyky a vývojová prostředí jsou složitá. Aby se studenti zorientovali ve vývojovém prostředí a naučili se ho ovládat, stojí je to mnohdy velké úsilí i spoustu času. U programovacího jazyka se musejí seznámit nejen s jeho syntaxí, ale i s knihovnamy, kde jsou předdefinovány různé metody, vlastnosti apod. I po delší době není zaručeno, že se studenti s programovacím jazykem a jeho syntaxí ztotožní. V koncepci výuky programování algorithms-first odpadají výše zmíněné problémy týkající se programovacího jazyku a vývojového prostředí. Studenti se obvykle seznámí pouze se základními algoritmickými konstrukcemi a příkazy, které se pro tvorbu algoritmů používají. Jejich pochopení pomocí pseudokódu může být rychlejší než u konkrétního programovacího jazyku. I když studenti musejí pochopit při tvorbě algoritmů určitá „syntaktická“ pravidla, oproti programovacímu jazyku jich je mnohem méně a zabývání se syntaxí je do značné míry minimalizováno. Oproti tomu studenti musejí vyvinout značné úsilí, aby vytvořili a odůvodnili navržené algoritmy a v hlavě si představili funkčnost jejich jednotlivých kroků. (ACM a IEEE-CS, 2001, s. 32)

Kromě pseudokódu se dají algoritmy vytvářet např. pomocí vývojových diagramů, strukturogramů, modelovacího jazyku UML aj.

Jak uvádějí společnosti ACM a IEEE-CS (2001, s. 33) „proces ladění v prostředí pseudokódu je zcela odlišný od procesu ladění spustitelného programu“.

I když ve svém aktualizovaném dokumentu studijních plánů společnosti ACM a IEEE-CS (2013, s. 55) již neuvádějí koncepci výuky programování algorithms-first, v novém okruhu k algoritmům uvádějí, že „algoritmy jsou zásadní pro počítačovou vědu a softwarové inženýrství“.

Podle těchto společností „dobrý návrh algoritmu je proto rozhodující pro výkon všech softwarových systémů. Navíc, studium algoritmů dává náhled do skutečné povahy problémů, stejně jako možného řešení technik nezávislých na

*programovacím jazyce, programovacím paradigmatu, počítačovém hardwaru, nebo nějakém jiném aspektu implementace“ (ACM a IEEE-CS, 2013, s. 55).*

### **Nevýhody koncepce výuky programování algorithms-first**

*„...studenti na úvodní úrovni chtějí zažít sílu a spokojenost, které pochází z přinucení počítačů skutečně něco dělat. Kurzy zaměřené pouze na stavbu algoritmů v pseudokódu zmaří tuto motivaci a přání“ (ACM a IEEE-CS, 2001, s. 33).*

Pecinovský (2006) uvádí, že *„zkušenost však ukazuje, že právě absence této zpětné vazby a nemožnost si vše vyzkoušet je pro studenty silně demotivující“*. Zpětnou vazbou autor rozumí ladění programů.

*„...studenti se vymaní z potřeby demonstrovat, že jejich algoritmy pracují v kontextu fungující implementace“ (ACM a IEEE-CS, 2001, s. 33).* Dané společnosti dále uvádějí, že *„...vyžaduje značné úsilí při hodnocení“*.

### **Breadth-first**

ACM a IEEE-CS (2001, s. 31) uvádějí, že *„rozvoj úspěšné breadth-first implementace se však ukázal být obtížný. V našich průzkumech byla nejčastější implementace myšlenky vytvoření úvodního kurzu „zaměřeného na rozsah“ představující obor studentům se zaměřením i bez specializace. Takový kurz seznamuje studenty s celou řadou zajímavých a důležitých témat, než je zaměří do podrobností konkrétní oblasti“*.

*„Typický úvodní kurz breadth-first zahrnuje základy počítačového programování, programovacích jazyků, umělé inteligence, operačních systémů, počítačové grafiky atd.“ (Vujošević-Janičić a Tošić, 2008, s. 69).*

Autoři dále uvádějí, že *„... slouží k pomoci studentům hlavně k rozhodnutí se, zda se mají nadále zabývat výpočetní technikou, a vytvořit si souvislost pro následující kurzy“*.

Pecinovský (2006) uvádí, že *„studenti prošli takovými kurzy přistupují k řešení problémů z většího nadhledu a jsou je často schopni chápat v celé jeho šíři“*.

Vujošević-Janičić, Tošić (2008, s. 69) ještě dále uvádějí, že *„koncepce výuky programování breadth-first je spíše komplexní, protože zahrnuje širokou škálu netriviálních témat“*.

### **Výhody koncepce výuky programování breadth-first**

ACM a IEEE-CS (2001, s. 31) uvádějí, „...že poskytuje studentům okamžité uznání rozsahu počítačové vědy, což jim umožní snadnější rozhodnutí, zda chtějí tuto oblast studovat do hloubky nebo ne“.

### **Nevýhody koncepce výuky programování breadth-first**

Mezi nevýhody společnosti ACM a IEEE-CS (2001, s. 31) uvádějí, že „...je přidání dalšího hlavního předmětu a zpoždění úvodní sekvence o semestr“.

Pecinovský (2006) uvádí, že „kritici však této koncepci vytýkají, že odkládá výuku programování a tím i na ni navazující předměty o jeden až dva semestry, což není vždy vyváženo lepšími znalostmi studentů“.

### **Concepts-first**

Vujošević-Janičić a Tošić (2008, s. 68) uvádějí, že „koncepce výuky programování concepts-first poskytuje přesný a stručný základ programování ve všech paradigmatech (imperativní, logické, funkcionální a objektově orientované) stejně jako i pro paralelní, souběžné a distribuované vícevláknové programování“.

Autoři také dále k této koncepci uvádějí, že „...je velmi dostupná, není ovšem tolik rozšířená“.

Pro tuto koncepci výuky programování je ve výuce nutné použít multiparadigmatický programovací jazyk. Multiparadigmatických programovacích jazyků je více, např. C++, C# apod. V této koncepci výuky programování je vhodným programovacím jazykem např. programovací jazyk Kernel. Příklady ve výuce by měly být zaměřeny na provázanost více paradigmat programování. Protože některá paradigmata programování jsou odlišná, příklady by měly být navrženy v logické návaznosti. Jednotlivá paradigmata programování by měla být zavedena pomocí jejich základních konstrukcí v probíraných příkladech. Při výuce je důležité studentům jednotlivá různě probíraná paradigmata vysvětlit, aby se v nich dokázali orientovat a dovedli používat základní konstrukce jednotlivých paradigmat programování. (Vujošević-Janičić a Tošić, 2008, s. 68)

## **Functional-first**

Společnosti ACM a IEEE-CS (2001, s. 30) uvádějí, že tato koncepce výuky programování „...se vyznačuje používáním jednoduchého funkcionálního jazyka, např. Scheme, v prvním kurzu“.

Podle Vujošević-Janičić a Tošić (2008, s. 74) „někteří učitelé věří, že čistě funkcionální programovací jazyk je přímo ideální pro úvodní kurzy, pokud se zaměřuje na obecné koncepty spíše než na specifika funkcionálního programování“.

Dále autoři uvádějí, že „...funkcionální paradigma programování je, na rozdíl od imperativního, vhodnější pro představení návrhu a analýzy algoritmů studentům, protože umožňuje soustředit se na problém spíše než na základní hardwarové charakteristiky“.

### **Výhody koncepce výuky programování functional-first**

Společnosti ACM a IEEE-CS (2001, s. 30) mezi výhody koncepce programování functional-first uvádějí:

- „...poměrně málo studentů přicházejících na vysoké školy bude mít již zkušenosti s programováním v tomto paradigmatu.
- Minimalizovaná syntaxe funkcionálních jazyků znamená, že předměty se mohou soustředit na základnější problémy.
- ...rekurze, datové struktury a funkce jako prvotřídní datové objekty, se v této oblasti přirozeně vyskytují a lze je pokrýt mnohem dříve ve studijním plánu“.

### **Nevýhody koncepce výuky programování functional-first**

Koncepce výuky programování functional-first vyžaduje poněkud jiný způsob uvažování při řešení problémů. V následujících navazujících předmětech programování se studenti v obvyklých případech setkají s rozšířenějšími koncepcemi výuky programování, které vycházejí z imperativního nebo OO paradigmatu programování. Studenti pak budou muset změnit svůj způsob uvažování, což se některým nemusí úplně snadno podařit. Základy programování pomocí koncepce výuky programování functional-first, i vzhledem k odlišnosti od obvyklejších koncepcí výuky programování, nejsou v praxi tolik rozšířené. Studenti nemusejí mít takovou motivaci učit se koncepci výuky, která není tolik rozšířená a v praxi nemá dostatečné využití. (ACM a IEEE-CS, 2001, s. 31)

Jiný způsob uvažování v koncepci výuky programování functional-first oproti jiným koncepcím výuky programování může vést u studentů k dalším těžkostem. Začínající studenti programování by měli, např. při návrhu programů, řešení programových úloh, návrhu algoritmů aj. zažít úspěch. Některé programové konstrukce mohou být pro začínající studenty abstraktní. Nemožnost úspěšně pochopit či naprogramovat danou úlohu může být pro řadu studentů demotivující a odstrašující při dalším studiu. Koncepce výuky programování functional-first klade vyšší důraz na abstrakci, hlavně na počátku výuky, než tradičtější koncepce výuky programování. Vyšší neúspěšnost studentů v jejich úvodním kurzu programování může být vyšší. Vysoká míra abstrakce může být ale například i u OOP. (ACM a IEEE-CS, 2001, s. 31)

### **Hardware-first**

Společnosti ACM a IEEE-CS (2001, s. 33) uvádějí, že *„koncepce výuky programování hardware-first učí základy počínaje vědou o počítačích na úrovni stroje a směřuje k abstraktnějším pojmům. Základní filosofií této koncepce výuky programování je naučit studenty o práci na počítači krok za krokem bez zbytečné mystifikace jak je to možné“*.

Počítač vykonává instrukce sekvenčně, za sebou. To je blízké člověku. Člověk pracuje v určitých situacích podobně, sekvenčně, např. při vaření obědu, sestavování počítače apod. Studenti jsou nejdříve seznámeni s tím, z jakých součástí a součástek je zkonstruován počítač. Pro ukládání dat v počítači a jejich zobrazování se používá paměť. Paměti se dají rozdělit podle různých hledisek a kritérií. Přístupování, zápis do paměti a zobrazování informací z paměti si studenti mohou vyzkoušet pomocí programování v nižším programovacím jazyku, např. assembleru. V další části potom studenti programují ve vyšším programovacím jazyku. (ACM a IEEE-CS, 2001, s. 33; Pecinovský, 2006)

### **Výhody koncepce výuky programování hardware-first**

Společnosti ACM a IEEE-CS (2001, s. 34) uvádějí, že *„tato koncepce výuky programování funguje dobře u studentů, kteří dávají přednost pochopení procesu výpočtu ve všech jeho detailech“*.

K předmětu, kde by se koncepce výuky programování hardware-first uplatnila, společnosti také uvádějí na straně 34, že „...by mohl být schůdný ve výpočetní technice tam, kdy je včasné seznámení s hardwarovými problémy zásadní“.

### **Nevýhody koncepce výuky programování hardware-first**

Společnosti ACM a IEEE-CS (2001, s. 34) uvádějí, že „koncepce výuky programování hardware-first je v rozporu s ústředním významem softwaru a tendencí stále více důmyslných virtuálních strojů oddělit proces programování od základního hardwaru“.

Pecinovský (2006) uvádí, že „tato koncepce se uplatní pouze v několika speciálních oborech...“.

### **Imperative-first**

Jak uvádějí Pecinovský (2006), ACM a IEEE-CS (2001, s. 29) a Vujošević-Janičić a Tošić (2008, s. 69) jedná se o nejpoužívanější / nejtradičnější / nejpoblíbenější koncepci výuky programování.

Tato koncepce výuky programování klade ve výuce programování důraz na práci s proměnnou, datovými typy, výrazy a základními řídicími konstrukcemi. Mezi základní řídicí konstrukce patří sekvence, větvení a cyklus (iterace). Aby programovaný kód nebyl moc dlouhý a byl přehledný, program se rozděluje do tzv. podprogramů: procedur a funkcí, které se nazývají v OO programech metody, viz kapitola 3.1.2. Mezi další témata této koncepce výuky programování patří např. rekurze nebo práce s polem. (ACM a IEEE-CS, 2001, s. 29; Vujošević-Janičić a Tošić, 2008, s. 69-70)

Teprve po zažití a procvičení příkladů výše uvedených základních pojmů, probíhá výuka zaměřená na OO programové konstrukce OOP. Pro výuku dvou různých, navazujících, paradigmat programování (imperativního paradigmatu programování a OOP) v této koncepci výuky programování imperative-first, může být použito jedno vývojové prostředí s jedním programovacím jazykem. Některé programovací jazyky, které jsou OO, podporují i imperativní paradigma programování, které je jejich součástí. Studenti si nemusejí zvykat na různá vývojová prostředí, jejichž pochopení trvá spoustu času i úsilí. OO konstrukce bývají do této koncepce výuky programování zařazeny na konci výuky předměty nebo v navazujícím předmětu. (ACM a IEEE-CS, 2001, s. 29)



Jak uvádějí ACM a IEEE-CS (2001, s. 29) „*názory v komunitě rozděluje to, jaká koncepce výuky programování by se měla zavést jako první*“.

„...*instituce používající koncepci výuky programování imperative-first budou muset zahrnout dodatečné pokrytí objektově orientovaných konstrukčních principů na středně pokročilé úrovni*“ (ACM a IEEE-CS, 2001, s. 30).

Autoři, kteří se zabývají výzkumy koncepcí výuky programování objects-first vs. imperative-first, nazývají koncepci výuky programování imperative-first různě. Uysal (2012) ji nazývá objects-late, Ehlert a Schulte (2009) ji nazývají objects-later nebo Johnson a Moses (2008) ji nazývají structures-first.

**Pokud bude v disertační práci autorem vymezena koncepce výuky programování imperative-first, bude tato koncepce výuky programování označena jako objects-later. Pro toto označení se autor práce rozhodl z důvodu cíle disertační práce, kde v daných dvou koncepcích výuky programování se u skupin studentů probírají nejen strukturované konstrukce, ale především i OO konstrukce.**

### **Výhody koncepce výuky programování imperative-first**

Ke koncepci výuky programování imperative-first Vujošević-Janičić a Tošić (2008, s. 69) uvádějí, že „...*není složitá na pochopení a v některých jednodušších případech je zřejmé převedení intuitivních algoritmů do kódu*“.

ACM a IEEE-CS (2001, s. 29) uvádějí, že „...*k tradičnímu imperativnímu stylu programování, který zůstává obecně rozšířený a který je nedílnou součástí jakéhokoliv objektově orientovaného jazyku*“.

### **Nevýhody koncepce výuky programování imperative-first**

Podle Pecinovského (2006) „...*takto připravovaní studenti se nesžijí s objektově orientovaným paradigmatickým tak dobře, jako studenti, kteří začali výuku hned prací s objekty*...“.

Podle Listera et al. (2006, s. 148) „*více než čtvrt století, před tím než bylo běžné učit koncepci výuky programování objels-first, se stejná témata<sup>6</sup> objevila v počítačově-vědecké vzdělávací literatuře: studenti nevědí, jak číst programy, nevědí jak navrhovat programy a nevědí jak psát programy*“.

---

<sup>6</sup> Z předchozích informací uvedených v článku se jedná o následující témata: „...*cykly, pole, algoritmické řešení problémů a rozklad programu*...“ (Lister et al., 2006, s. 148).

Autoři dále uvádějí, že „nedávno dvě ITiCSE pracovní skupiny zjistili, že studenti stále považují imperativní programování za obtížné“.

Co je jasné pro vyučujícího, nemusí být jasné pro žáka. Při výuce základních konstrukcí koncepce výuky programování objects-later, mezi které patří např. přiřazování a sekvence, je důležité dbát na to, aby si dané konstrukce žáci pořádně zažili a pochopili je. Při výuce dalších konstrukcí, které na výše dvě zmíněné konstrukce navazují, se může řada studentů setkat s problémy. Proto je důležité, aby vyučující základním stavebním kamenům koncepce výuky programování objects-later, mezi něž patří výše zmíněné přiřazování a sekvence, věnovali dostatečnou pozornost. (Vujošević-Janičić a Tošić, 2008, s. 69-70)

### **Logical-first**

Vujošević-Janičić a Tošić (2008, s. 75) uvádějí, že „tento programovací styl zdůrazňuje deklarativní popis problému spíše než rozklad problému na algoritmickou implementaci“.

Autoři dále uvádějí, že „programovací jazyk Prolog je nejběžnější jazyk užívaný pro představení logického paradigmatu“.

### **Nevýhody koncepce výuky programování logical-first**

Vujošević-Janičić a Tošić (2008, s. 75) uvádějí některé nevýhody této koncepce výuky programování, mezi které patří např. to, že „...studenti musejí pochopit základní matematické koncepty stejně jako principy prováděcího mechanismu Prologu. Tyto dva úkoly jsou obtížné dokonce i pro zkušené programátory...“.

### **Variables-first**

Autory této koncepce výuky programování jsou J. Sajaniemi a Ch. Hu, kterou autoři představili v článku Sajaniemi, Hu (2006) *Teaching Programming: Going beyond „Objects First“*.

Jak strukturované programování, tak i OOP využívají při tvorbě programů proměnné. V každém z těchto rozdílných programování se klade odlišný důraz na práci s proměnnou. U strukturovaného programování se proměnné využívají v souvislosti se základními řídicími konstrukcemi (sekvence, větvení a cyklus) pro uchovávání hodnot v proměnných, podle jejichž hodnoty se vykonávaný program

může chovat určitým způsobem, popřípadě jako parametry procedur a funkcí. U OOP se proměnné využívají ve třídách především jako atributy, respektive datové členy a u metod také jako parametry. Koncepce výuky programování variables-first se nejdříve zaměřuje na proměnné u strukturovaného programování a jeho řídicích struktur a dále, v brzkém přechodu, u atributů v OOP. Zkušenost s proměnnou ze strukturovaného programování nemusí dělat studentům u OOP potíže, např. velkou mírou abstrakce. Základní pojmy OOP se vysvětlují až po pochopení pojmu proměnná. Rozdíl této koncepce výuky programování oproti koncepci výuky programování objects-later je brzký přechod ze strukturovaného programování na OOP, kdy se klade důraz na práci s proměnnými. (Sajaniemi a Hu, 2006, s. 258-259)

### **3.2.2 Koncepce výuky programování s OOP v úvodním předmětu programování**

Mezi koncepce výuky programování, které preferují styl programování OOP na začátku předmětu programování, se dají zařadit následující koncepce výuky programování:

- Architecture First
- Design Patterns First
- model-first
- objects-first

#### **Architecture First**

Autorem této koncepce výuky programování je Rudolf Pecinovský. Autor ji představil např. v článku Pecinovský (2013) *Methodology Architecture First* nebo v článku Pecinovský (2015) *Metodika Architecture First*.

Dále k ní Pecinovský (2013, s. 108) uvádí, že „...byla původně zveřejněna pod názvem *Design Patterns First*, ale v průběhu dalšího vývoje si autor uvědomil, že časný výklad návrhových vzorů je pouze důsledkem celkového zaměření na časný výklad architektury, a proto metodiku přejmenoval“. Dále proto podle autora „...nejprve se studenty probírá základní principy budování architektury objektově orientovaných programů, a teprve poté, co tyto principy studenti vstřebají, pokračuje výkladem způsobu, jak lze navržený program zakódovat“.

Autor výuku této koncepce rozděluje do čtyř etap, které jsou uvedeny v článku Pecinovský (2013).

### **Design Patterns First**

Tato koncepce výuky programování byla představena v některých člancích R. Pecinovského, např. v článku Pecinovský (2008) *Mýty ve výuce programování a metodika Design Patterns First* nebo v článku Pecinovský (2010b) *Metodika Design Patterns First* nebo v článku Pecinovský (2010c) *Metodika Design Patterns First v roce 2010*.

Ke vzniku koncepce výuky programování Pecinovský (2010c) uvádí, že „...*Design Patterns First se narodila v roce 2004 v reakci na nedostatky jiných, v té době používaných koncepcí výuky programování*“. V souvislosti s koncepcí výuky programování objects-first Pecinovský (2008) uvádí, že „...*Design Patterns First proto přebírá od koncepce výuky programování Object First myšlenku začít kurz s vývojovým prostředím, které umožní pracovat z počátku v interaktivním režimu*“.

V člancích Pecinovský (2008) a Pecinovský (2010c) autor uvádí důležité zásady, které by se měly uplatňovat při návrhu programů a které koncepce výuky programování Design Patterns First splňuje.

Pecinovský (2010a, s. 159) uvádí, že „...*Design Patterns First, která seznamuje studenty s rozhraním hned na počátku výkladu a prakticky od počátku kurzu také zařazuje témata přibližující použití (a tím i význam) návrhových vzorů*“.

Detailní popis jednotlivých kroků (fází) výuky výše uvedené koncepce výuky programování je uvedeno např. v člancích Pecinovský (Pecinovský, 2008; Pecinovský, 2010c).

### **Model-first**

Jak uvádí Rýdlo (2012, s. 12), tato koncepce výuky programování „...*rozšiřuje koncepci výuky programování objects-first a na samotný začátek klade modelování problému*“.

Na začátku této koncepce výuky programování je využíván modelovací jazyk, např. UML. Tento modelovací jazyk obsahuje několik typů diagramů, které mají různé využití. Ve vývojovém programovacím prostředí BlueJ je při návrhu architektury programu (viz např. koncepce výuky programování Architecture First)

v programu zobrazován diagram tříd. Tento, popřípadě i jiné diagramy, se dají na začátku výuky využít při modelování různých problémů, se kterými se studenti mohou dále setkávat při návrhu programu. Více o modelování lze najít v různých publikacích, které jsou k dispozici na českém trhu. Teprve po návrhu programu pomocí různých diagramů přichází výklad základních OO konstrukcí. Tato koncepce výuky programování po úvodním modelování přechází do koncepce výuky programování objects-first, protože po OO konstrukcích se dále vykládají řídicí konstrukce (viz koncepce výuky programování Objects-first). (Rýdlo, 2012, s. 12)

### **Výhody koncepce výuky programování model-first**

Rýdlo (2012, s. 12) uvádí k této koncepci výuky programování, že *„...je vhodná díky důrazu na fázi návrhu, takže studenti mohou při samotném programování vycházet z připraveného modelu. To podporuje systematické řešení úkolu a analytické myšlení“*.

### **Objects-first**

Ke vzniku této koncepce výuky programování Pecinovský (2010a, s. 158) uvádí, že vznikla *„kolem roku 2000...“* Dále autor uvádí, že *„...prosazovala zahájení výuky přímo výkladem objektů a prací s nimi“*.

Podle společností ACM a IEEE-CS (2001, s. 30) je zásadní, že tato koncepce výuky programování *„...od samého počátku klade důraz na principy objektově orientovaného programování a návrhu“*. Podobné vymezení uvádí i např. Uysal (2012) nebo výše Pecinovský (2010a).

K obsahové náplni této koncepce výuky programování ACM a IEEE-CS (2001, s. 30) dále uvádějí, že *„úvodní předmět v jednom sledu ihned začíná představením objektů a dědičnosti, aby se studenti s těmito myšlenkami brzy setkali“*.

K další návaznosti společnosti také uvádějí, že *„...předmět dále pokračuje představením tradičních řídicích struktur, ale vždy v kontextu zaměření na objektově orientovaný návrh“*.

Oproti koncepci výuky programování objects-later podle Pecinovského (2010a, s. 159) *„klasické konstrukce strukturovaného programování (podmínky, cykly, ...) doporučovala vykládat až poté, co si studenti osvojili základy práce s objekty bez použití těchto konstrukcí“*.

Ehlert a Schulte (2009, s. 16) uvádějí seznam témat, která by měla být zahrnuta v koncepci výuky programování objects-first:

- *„Přehled OOP, tříd a objektů*
- *Proměnné/atributy, konstanty, primitivní datové typy*
- *Řídící struktury: iterace a výběr*
- *Podprogramy: procedury/metody*
- *Neprimitivní datové typy*
- *Dědičnost*
- *Asociace“.*

### **Výhody koncepce výuky programování objects-first**

ACM a IEEE-CS (2001, s. 30) uvádějí, že *„největší výhodou v koncepci výuky programování objects-first je brzká orientace k objektově orientovanému programování, které nabývá na důležitosti v akademickém světě i průmyslu“.*

### **Nevýhody koncepce výuky programování objects-first**

Pecinovský (2010a, s. 159) uvádí, že *„...Object First se však soustředila pouze na objekty jako základní konstrukce, avšak konstrukci rozhraní nechávala až na konec základního výkladu a o návrhových vzorech se v základním kurzu prakticky nezmínila“.*

Při programování může být pro řadu studentů náročné naučit se nejen syntaxi konkrétního programovacího jazyka, ale i naučit se dostatečně ovládat příslušné vývojové prostředí. Často se ve výuce programování používají profesionální vývojová prostředí, která se uplatňují v praxi. Jejich osvojení může studentům zabrat příliš mnoho úsilí. Proto je při výuce programování pomocí koncepce objects-first důležité vzít v potaz výše uvedené možné těžkosti. Je důležité zvolit obsahovou náplň předmětu pomocí koncepce výuky programování objects-first tak, aby studenti nebyli zahlceni příliš velkým množstvím nových informací a množstvím požadavků na jejich dovednosti a schopnosti. (ACM a IEEE-CS, 2001, s. 30)

Vystavěl (2008, s. 191) uvádí, že *„za problémem koncepce výuky programování „objects first“ považují vyšší míru abstrakce a delší čas potřebný k tomu, aby se studenti dostali k zajímavým aplikacím“.*

### 3.3 Výhody/nevýhody OOP a strukturovaného programování v úvodních předmětech programování

V předchozí kapitole byly v podkapitolách u koncepcí výuky programování objects-first a objects-later uvedeny jejich výhody a nevýhody. Někteří autoři uvádějí výhody a nevýhody paradigmat programování, která se vyučují v prvních předmětech programování. Daná paradigmata programování nemusejí mít, jako podle předpokladu koncepce výuky programování, další návaznost na jiná paradigmata programování. Proto mohou být uvedené výhody/nevýhody daných paradigmat programování brány z jiného úhlu pohledu, než jak bylo uvedeno u výhod a nevýhod v příslušných podkapitolách daných koncepcí výuky programování. V této podkapitole jsou uvedeny výhody a nevýhody OOP a strukturovaného programování<sup>7</sup> v úvodních předmětech programování.

Vujošević-Janičić a Tošić (2008, s. 72) uvádějí ke zvolení OOP v úvodním předmětu programování, že „*prvním a hlavním důvodem je velká důležitost a oblíbenost tohoto paradigmatu*“.

Další výhodou OOP v úvodním předmětu programování je minimalizace problémů při přechodu studentů ve výuce programování z jiných a odlišných paradigmat programování, např. strukturovaného programování. Zažití odlišných návyků v různých paradigmatech programování může působit studentům velké obtíže při učení se novému paradigmatu. Nemusejí využít výhod toho konkrétního paradigmatu, protože mohou být ovlivněni předchozím paradigmatem programování. (Vujošević-Janičić a Tošić, 2008, s. 72)

Donchev a Todorova (2008, s. 160) tento důvod, kdy je nejdříve vyučuje OOP místo dříve hojně vyučovaného strukturovaného programování, nazývají „...*paradigmatickým posunem*“.

---

<sup>7</sup> Někteří autoři vymezují výhody a nevýhody imperativního paradigmatu programování. V práci nebude předpokládáno, že by autoři u imperativního paradigmatu programování měli na mysli vymezení vzhledem k naivnímu paradigmatu nebo nestrukturovanému paradigmatu programování, jak bylo uvedeno v kapitole Paradigmata programování, ale že mají na mysli především strukturované programování. Proto se bude dále v této kapitole pro lepší srozumitelnost textu u citovaných nebo parafrázovaných informací používat u imperativního paradigmatu programování označení strukturované programování.

Vujošević-Janičić a Tošić (2008) uvádějí, že „jedním z hlavních problémů je, že toto programování<sup>8</sup> vyžaduje analýzu a návrhářské schopnosti před kódováním programu“.

Analýza a objektový návrh, s níž se studenti seznámí nejdříve v předmětu programování, má také řadu úskalí. Studenti při této výuce tolik neprogramují v kódu. Navrhují objektový návrh programu. Navrhují objekty, které budou v návrhu potřebné a jejich vzájemný vztah. Specializují se a jsou z nich dobří objektoví návrháři. Při vlastní implementaci pak mívají problémy s naprogramováním funkčnosti daného objektového návrhu. Nemusejí mít dobré programátorské návyky a schopnosti pro programování v konkrétním programovacím jazyku. Při objektovém návrhu programu je důležité, aby se studenti seznámili i s příslušnou implementací při programování funkčnosti. Návrh objektové architektury by měl být vhodně doplněn i o programátorské hledisko. (Vujošević-Janičić a Tošić, 2008, s. 72)

Donchev a Todorova (2008, s. 160) uvádějí, že „...zdá se být nevhodné vyučování objektově orientovaného návrhu (OOD) bez předchozích zkušeností s používáním tříd a jejich hierarchie“.

Podle Listera et al. (2006, s. 149) „ačkoliv výuka objektově orientovaného programování<sup>9</sup> nedělá studium programování jednodušší, literatura nepodporuje tvrzení, že ho dělá obtížnějším. Tam jsou důkazy, že objektově orientovaný styl usnadňuje začátečnickům mapování mezi programem a jeho doménou“.

Vujošević-Janičić a Tošić (2008, s. 69) uvádějí, že „strukturované programování není obtížné na pochopení a v některých jednodušších případech je jednoduché převést intuitivní algoritmy do kódu“.

U strukturovaného programování mohou začínajícím programátorům působit problémy některé jeho základní konstrukce. K tomu Dehnadi a Bornat (2006, s. 5) uvádějí, že „ze zkušeností je zřejmé, že existují tři hlavní sémantické překážky, kde začátečník udělá chybu v imperativních programech. Popořadě jsou to:

- přiřazování a sekvence
- rekurze a iterace
- souběžnost“.

---

<sup>8</sup> Autoři místo pojmu programování uvádějí v článku pojem přístup. Pro lepší srozumitelnost textu bude u citovaných nebo parafrázovaných informací používáno označení programování.

<sup>9</sup> Stejně jako u předchozí poznámky, viz poznámka pod čarou 8



Pro studenty bývají základní programovací konstrukce, např. sekvence nebo symbol přiřazení u strukturovaného programování, těžké na pochopení. Přitom na první pohled se jedná o celkem pochopitelné pojmy, se kterými se mohou studenti běžně setkávat v běžném životě, např. při přípravě pokrmu apod. V programování u přiřazování hodnot do proměnných si obvykle studenti neumějí představit, jak pracuje paměť a jak se do ní ukládají hodnoty. Při nemožnosti názorné představy mohou přiřazování používat v programech automaticky, bez potřebného rozmyslu. Což může vést k chybám a může se projevit problémy např. i při příkladech se sekvencí příkazů. Je důležité, aby se v předmětech věnovala těmto konceptům zvýšená pozornost. Studentům by se mělo předložit dostatečné množství názorných příkladů, protože správné pochopení základním konceptům strukturovaného programování je důležité z hlediska dalších probíraných programátorských konceptů. (Vujošević-Janičić a Tošić, 2008, s. 69-70)

Vujošević-Janičić a Tošić (2008, s. 72) dále uvádějí, že *„zejména, mít více zkušeností ze strukturovaného programování, dělá obtížnější změnu k objektově orientovanému paradigmatu“*.

Lister et al. (2006, s. 148-149) uvádějí, že *„v roce 2004 Leeds Working group testovala schopnosti více než 500 studentů z vysokých škol v několika zemích na sérii otázek s výběrem odpovědi... V této studii studenti nepsali kód. Místo toho pouze museli kód přečíst a ukázat, že mu rozumí. Všechny otázky se zaměřovaly na iterativní procesy polí a jednalo se tedy o tradiční imperativní kód, přesto čtvrtina studentů předvedla úroveň odpovídající náhodnému tipování“*.

### **3.4 Výzkumy koncepcí výuky programování objects-first vs. objects-later**

V této části jsou uvedeny výzkumy autorů, které se týkají koncepcí výuky programování objects-first vs. objects-later<sup>10</sup>. Výzkumy uvedených koncepcí výuky programování se zabývali autoři Johnson a Moses (2008) v článku *„Objects-First vs. Structures-First Approaches to OO Programming education: An Empirical Study“*,

---

<sup>10</sup> Autoři výzkumů uvedených v této kapitole označují koncepcie výuky programování objects-first a objects-later různými termíny. Pro lepší srozumitelnost textu bude u citovaných nebo parafrázovaných informací používáno označení koncepcí výuky programování objects-first a objects-later.

Ehlert a Schulte (2009) v článku „*Empirical Comparison of Objects-First and Objects-Later*“ a Uysal (2012) v článku „*The Effects of Objects-First and Objects-Late Methods on Achievements of OOP Learners*“.

### **3.4.1 Výzkumné otázky a hypotézy ve výzkumu koncepcí výuky programování**

Johnson a Moses (2008, s. 245) formulovali výzkumnou otázku ve své studii následovně: „*Jaký výsledek dělá vyučování s koncepcí výuky programování objects-first (protiklad výuky s koncepcí výuky programování objects-later) na to, jak dobře studenti úvodu do programování chápou OO koncepty a píší OO programy?*“ Z výzkumné otázky byla autory stanovena zkoumaná hypotéza: „...*ve výkonu studentů úvodu do programování není žádný rozdíl, pokud jde o koncepci výuky programování objects-first nebo objects-later k OO programování.*“

Ehlert a Schulte (2009, s. 17) formulovali výzkumnou otázku (autoři formulovali výzkumné otázky dvě, ale je uvedena pouze výzkumná otázka, která se týká problematiky v této práci) ve své studii ve znění: „*Liší se skutečně tyto dvě koncepce výuky programování<sup>11</sup>, pokud jde o výsledek učení?*“ Autoři dále neuvádějí výzkumnou hypotézu, která by byla stanovena na základě výzkumné otázky.

Uysal (2012, s. 817) stanovil výzkumnou hypotézu ve znění: „*Úspěchy studentů se nebudou podle koncepce výuky OOP významně lišit*“.

Z výše uvedených výzkumných otázek a výzkumných hypotéz vyplývá, že autoři Ehlert a Schulte (2009) zjišťují celkové výsledky dvou koncepcí výuky programování bez ohledu na jednotlivé konstrukce, které se v daných koncepcích výuky programování používají. Autoři Johnson a Moses (2008) i autor Uysal (2012) zkoumají celkové výsledky obou skupin studentů vzhledem k OO konstrukcím. U autora Uysala (2012) je to zřejmější, než z výzkumné hypotézy, z dalších informací, které jsou uvedeny v jeho výzkumné studii.

### **3.4.2 Rozdělení studentů do dvou skupin podle různých kritérií**

Výzkumná studie Johnsona a Mosese (2008, s. 245) probíhala u vysokoškolských studentů. Autoři dále uvádějí, že „...*každý z autorů této studie*

---

<sup>11</sup> Autoři mají na mysli koncepcí výuky programování objects-first a objects-later

vyučoval jednu skupinu v úvodu OO programování (CIS 260) s účastí asi 25 studentů během semestru 2007 na podzim na Missouri State University (MSU)<sup>12</sup>.

K rozdělení studentů autoři uvádějí, že „*studenti absolvovali průzkum, kterým byla získána demografická data jako pohlaví, věk, ročník, hlavní a vedlejší oblasti studia, příprava s jazykem Java a dalšími programovacími jazyky, znalosti o používání počítačů a touze naučit se počítačové programování*“. Tato data posloužila autorům studie, „...*aby zjistili, že obě skupiny (OF a SF<sup>12</sup>) mají podobné znalosti a schopnosti*“.

Autoři Ehlert a Schulte (2009, s. 18) uvádějí, že „*studie byla provedena na škole Oberstufenzentrum Informations- und Medizintechnik (OSZ IMT) v Berlíně v letech 2006/07 (předběžná studie) a 2007/08 (hlavní studie)*“.

Autoři dále uvádějí, že „*během hlavní studie měli studenti v průměru 17 let*“.

K rozdělení studentů autoři uvádějí na straně 17, že „*pokud jde o studenty, před experimentem byli všichni studenti rozděleni do podskupin tak, aby byly obě skupiny co nejpodobnější s ohledem na hlavní ovlivňující proměnné. Za takové proměnné jsme zvolili pohlaví, věk, předchozí známky ze zkoušek z německého jazyka a matematiky*“.

K dalším sledovaným předmětům autoři na téže straně uvádějí, že „*bylo tak možné přidat dodatečnou kontrolu: obě třídy byly přiděleny stejnému učiteli matematiky. Vzhledem ke stejným dřívějším známám z matematiky jsme očekávali, že budou mít obě třídy podobné známky z matematiky i na konci školního roku*“.

Počet žáků, který byl v daných dvou skupinách, autoři neuvádějí.

Uysal (2012, s. 817) uvádí, že „*dvacet OOP vysokoškolsky vzdělaných studentů se účastnilo této studie. Studenti získali bakalářský titul v oboru systémového inženýrství. Studenti měli základní vědomosti ze strukturovaného programování v programovacím jazyku Java a dříve používali JCreator LE jako nástroj pro vývoj softwaru*“.

Podle informací o výsledcích jednotlivých účastníků, které autor uvádí, bylo v každé skupině deset studentů.

Podle jakých kritérií byli studenti do dvou skupin rozděleni, autor neuvádí.

---

<sup>12</sup> Objects-first a structures-first

Srovnatelnost skupin, respektive výsledků v daných dvou skupinách, jak se dá z kontextu článku pochopit, byla na straně 817 podle autora dána tak, že *„studenti byli testováni zkouškami před a po absolvování kurzu a skupina 2 byla považována za experimentální. Zkoušky před a po absolvování kurzu byly podány jako otevřené vysokoškolské výkonnostní zkoušky aby zjistily, jaké znalosti OOP a dovednosti studenti získali“*.

### **3.4.3 Postup řešení výzkumu**

Postup řešení výzkumu lze realizovat např. následujícími prostředky: náplní předmětu programování, hodinovou dotací, zvoleným programovacím jazykem a průběhem výuky předmětu programování.

Ve výše uvedených výzkumných studiích se jednotlivá obsahová i časová náplň předmětu programování lišila. Lišil se i způsob vyučování daného předmětu programování u jednotlivých autorů studií.

Jak již bylo uvedeno v předcházející podkapitole, u studie Johnsona a Mosese (2008, s. 245) každou skupinu vyučoval jiný vyučující. *„...nebylo možné, aby stejný instruktor učil obě dvě skupiny kvůli konfliktu harmonogramu“*.

Autoři dále uvádějí, že *„aby byly rozdíly v předmětu co nejmenší, vybrali si učitelé dva různé texty napsané stejným autorem...“*.

Obsahovou náplň předmětu programování ve studii autoři neuvádějí. Z kontextu studie autorů se dá předpokládat, že obsahovou náplní předmětu byly základy OOP: práce s objekty a třídami, které autoři uvádějí jako obsahovou náplň předmětu v učebním textu.

Hodinová dotace vyučování ve studii také není uvedena. Autoři na straně 247 dále uvádějí, že programovacím jazykem předmětu programování byla Java.

Ehlert a Schulte (2009, s. 18) o rolích vyučujících dodávají: *„Samozřejmě je možné, že různé charakteristiky zapojených učitelů ovlivnily studii – aby se toto co nejvíce kontrolovalo, konaly se pravidelné schůzky a otázky týkající se učitele se zahrnují do dotazníků, jež studenti vyplňovali po každém modulu“*.

Autoři dále uvádějí, že *„oba učitelé také spolupracovali na tom, aby měl každý modul stejnou délku (maximální rozdíl byl 3 hodiny)“*.

Obsahová náplň předmětu hlavní studie autorů Ehlerta a Schulteho (2009) je uvedena v tabulce 1.

**Tabulka 1** – „*Sekvence témat dosud vyučovaných ve skupině koncepcí výuky programování objects-first a objects-later*“ (Ehlert a Schulte, 2009, s. 18), tabulka převzata z Ehlert a Schulte (2009, s. 18)

	Objects-first (témata)	Objects-later (témata)
1	T1: Úvod do OOP s BlueJ (úvod do pojmů, použití objektů)	T2: Úvod do IDE, proměnná, konstanta, datové typy, sekvence (zde bez souvislosti s atributy)
2	T2: Úvod do IDE, proměnná, konstanta, datové typy, sekvence (zde v souvislosti s atributy)	T5: Výběry včetně diagramů Nassiho a Shneidermana
3	T3: Třída a objekt (implementace tříd a instance objektů)	T6: Smyčky včetně diagramů Nassiho a Shneidermana
4	T4: Metody (v rámci tříd)	T7: Pole a řetězce (zde: řetězce se zavedly jako speciální datový typ, nikoliv jako třída)
5	T5: Výběry včetně diagramů Nassiho a Shneidermana	T1: Úvod do OOP s BlueJ
6	T6: Smyčky včetně diagramů Nassiho a Shneidermana	T3: Třída a objekt
7	T7: Pole a řetězce	T4: Metody
8	T8: Dědičnost, abstraktní metody	T8: Dědičnost, abstraktní metody
9	T9: Asociace, statické členy	T9: Asociace, statické členy

„V obou skupinách byl stejný čas výuky, zkoušky, programovací jazyk a nástroje. Jednotlivé úkoly, domácí úkoly a použité příklady se však musely lišit...“ (Ehlert a Schulte, 2009, s. 17).

Jak autoři dále uvádějí na straně 18, tak „*cílem úvodní sekvence je zavést do objektově orientovaného paradigmatu a do programovů základy jazyka Java*“.

K časové náplni předmětu a k vyučujícímu výzkumné studie Uysal (2012, s. 818) uvádí, že „*tento kurz poskytl studentům základy konceptů OOP. Sestával se ze dvou přednáškových a dvou laboratorních hodin týdně a probíhal 13 týdnů v rámci semestru. Skupiny 1 i 2 vyučoval tentýž lektor, výuka probíhala v různých dnech týdne*“.

K programovacímu jazyku autor na straně 817 uvádí, že „*obojí účastníci ve skupině 1 i skupině 2 používali BlueJ IDE*“.

K obsahové náplni koncepce výuky programování objects-later autor na straně 818 dodává, že „... zpočátku následuje klasický výukový vzor, který byl podobný tomu jako ve většině kurzů strukturovaného programování v Javě...“. K tomu autor dále dodává, že „účastníci začali s textovým rozhraním BlueJ“.

Ke studentům koncepce výuky programování objects-later autor na straně 819 dále uvádí, že „studenti skupiny 1 využívali vizuální podporu BlueJ pro OO konstrukce (design) a podporu interakce s objekty až v posledních týdnech (v 8. až 13. týdnu)“.

Oproti tomu, jak uvádí autor na straně 816 „studenti objects-first využívali všechny vizuální funkce BlueJ IDE“. Obsahová náplň předmětu studie autora je uvedena v tabulce 2.

**Tabulka 2 – „Sekvence a struktura probíraných témat kurzu pro koncepcí výuky programování objects-first a objects-later“ (Uysal, 2012, s. 818), tabulka převzata z Uysala (2012, s. 818)**

Týdny	Objects-first (témata)	Objects-later (témata)
1	Seznámení s prostředím BlueJ IDE, paradigmatickým OOP, objektem, třídou a metodou.	Úvod do BlueJ IDE, představení třídy, metod a atributů na začátečnické úrovni.
2	Objekty, třídy, metody, parametry, pole, stavy objektů, prezentace a aplikace těchto konceptů v prostředí BlueJ.	Datové typy, proměnné, konstanty. Příkazy přiřazení. Odkazování na objekty a třídy.
3	Příkazy přiřazení, konstruktor, datové typy, prezentace a aplikace v prostředí BlueJ.	Použití operátorů a podmíňovacích příkazů v příkladech aplikací.
4	Programování, ladění a zkoušení principů v prostředí BlueJ.	Smyčky a struktury opakování s příklady aplikací.
5	Koncepty abstrakce, dědičnost a polymorfizmu v prostředí BlueJ.	Editace a modifikace existujících tříd. Vytváření nových tříd se členy a metodami, pozorování chování objektů a stavů v příkladech aplikací.
6	Základní principy objektově orientovaného designu diagramů s UML.	Zacházení s událostmi v aplikacích Javy.
7	Seznámení s operátory, podmíňovacími	Práce s grafikou (tj. kreslení čar,

	příkazy, smyčkami a opakováním v BlueJ s jednoduchými příklady.	obdélníků a kruhů). Představení AWT.
8	Editace a modifikace existujících tříd. Vytváření nových tříd se členy a metodami, pozorování chování objektů a stavů v příkladech aplikací.	Projekt „Shapes“: vytváření jednotlivých tříd kruhů, čtverců a trojúhelníků, které lze přemísťovat a upravovat co do velikosti a barvy.
9	Projekt „Shapes“: vytváření jednoduchých tříd kruhů, čtverců a trojúhelníků, které lze přemísťovat a upravovat co do velikosti a barvy.	Představení abstrakce, dědičnost a polymorfizmu se základními diagramy UML.
10	Principy stavby velkých projektů, interakce několika objektů pro dokončení programovacího úkolu.	Principy stavby velkých projektů, interakce několika objektů pro dokončení programovacího úkolu.
11	Projekt „Picture“: vytváření třídy obraz kombinací tvarů pro vykreslení jednoduchých obrázků.	Projekt „Picture“: vytváření třídy obraz kombinací tvarů pro vykreslení obrázku.
12	Úvod do softwarového inženýrství, procesy softwarového rozvoje a principy objektově orientované konstrukce.	Úvod do softwarového inženýrství, principy objektově orientované konstrukce, procesy softwarového rozvoje.
13	Závěrečná zkouška	Závěrečná zkouška

### 3.4.4 Způsob testování studentů

Ve studii Johnsona a Mosese (2008, s. 245) „každá skupina studentů vykonala v průběhu semestru tři zkoušky (zkouška 1, zkouška 2 a závěrečná zkouška). Každá zkouška se skládala z 25 otázek s možností výběru z několika odpovědí, které pokrývaly programovací koncepty a jeden programovací problém“.

Autoři dále na straně 246 dodávají, že „...pouze první a poslední zkouška byly využity v této studii, neboť pouze tyto dvě zkoušky byly stejné pro obě skupiny. Zkouška 1 byla pro obě skupiny stejná, protože zahrnovala první dvě kapitoly v obou textech (základní programování v Javě zahrnující proměnné, jednoduché algoritmy a sekvenční struktura)“.

K závěrečné zkoušce autoři uvedli, že „závěrečná zkouška byla pro obě skupiny stejná a skládala se z otázek s možností výběru z několika odpovědí a týkala se OO konceptů a OO programovacího problému“.

Ehlert a Schulte (2009, s. 17) k testování studentů uvádějí, že „v našem případě to znamenalo test na konci jednoho školního roku a druhý test po letních prázdninách pro kontrolu dlouhodobých účinků. Test získaných znalostí zahrnuje veškerá vyučovaná témata“.

K jednotlivým testům autoři dále uvádějí, že „celkově lze říci, že závěrečné a navazující testy celkově zahrnují jedenáct částí (devět částí pro každé téma výuky a dvě další témata týkající se OOM<sup>13</sup> a dynamického OOP). Každá část obsahuje otázku jako ukazatel získané znalosti specifické pro modul. Každá z těchto devíti částí byla klasifikována jako buď objektově orientovaná, nebo procedurální – takže se témata použila k definování různých sekvencí verzí kurzu OOP-first a OOP-later a k definování témat pro závěrečné testy“.

K časové náplni autoři uvádějí, že „studenti dostali 100 minut pro zodpovězení otázek 9 témat a dalších 30 minut na zodpovězení otázek dvou dalších dotazů týkajících se modelování a dynamického OOP“.

K obsahové náplni testů autoři uvádějí, že „témata v testech související i nesouvisející s OOP byly zastoupeny rovnoměrně (přibližně 50 % od každých). Některé otázky byly podány jako otevřené, některé jako otázky s více odpověďmi (ne více než 30 % ...)“.

Ve studii Uysala (2012), jak bylo uvedeno v podkapitole 3.4.2 studenti absolvovali zkoušku před zahájením předmětu a závěrečnou zkoušku na konci kurzu.

„Pretest a posttest byly dány ke stanovení, zda byly získány znalosti a dovednosti<sup>14</sup> OOP“ (Uysal, 2012, s. 819).

K obsahové náplni testů autor na stranách 819 a 820 uvádí, že „studenti byli požádáni, aby v rámci výkonnostních testů uváděli správný popis OOP pojmů a v Javě napsali kompletní aplikaci. Úmyslem bylo změřeni jejich porozumění a nabytí dovedností OOP studentů, zda naplnili výsledky učení“.

---

<sup>13</sup> Objektově orientované modelování

<sup>14</sup> Autor má na mysli znalosti a dovednosti studentů



### 3.4.5 Výsledky výzkumné studie

Výsledky zkoušek výzkumné studie Johnsona a Mosese jsou uvedeny v jejich studii, viz (Johnson a Moses, 2008, s. 247).

Z výsledků autorů je patrné, že především v praktické úloze v závěrečné zkoušce dopadli lépe studenti koncepce vyučování objects-first.

Autoři k výsledkům na straně 247 uvádějí, že *„i když tyto výsledky by mohly být vysvětleny určitými změnami v kurzech vyučovaných dvěma rozdílnými učiteli, taková změna byla minimalizována opatřeními, která zajistila vysoký stupeň shody. Je mnohem pravděpodobnější, že vliv na vyšší skóre skupiny OF<sup>15</sup> v závěrečné zkoušce měla koncepce výuky programování objects-first“*.

Podle autorů na stranách 247-248 je možné, že *„...v průběhu času by skupina koncepce výuky programování objects-later mohla zvládnout OO programování jako skupina koncepce výuky programování objects-first“*.

Ehlert a Schulte (2009, s. 19) uvádějí výsledky závěrečného testu studie i navazujícího testu.

Autoři Ehlert a Schulte (2009, s. 19) uvádějí, že *„...obě třídy dosáhly obecně srovnatelných výsledků, ve dvou tématech byla jedna třída výrazně lepší, ale zde se jednalo o skupinu objects-first namísto skupiny objects-later“*. Autoři dále uvádějí, že *„kromě toho výsledky v T4 (metody) jsou pozoruhodné, i když se významně neliší“*.

Podle autorů *„ukázalo se, že skupina objects-first měla lepší (intenzivnější) přípravu na závěrečný test než druhá skupina“*.

K výsledkům navazujícího testu, které jsou uvedeny v tabulce ve studii autorů, viz Ehlert a Schulte (2009, s. 19), autoři uvádějí, že *„zdá se spíše zajímavé, že v navazujícím testu zmizely všechny významné rozdíly“*.

Autoři také dále uvádějí, že *„pokud jde o celkový výsledek i jednotlivé výsledky v jednotlivých tématech, nedokázali jsme potvrdit konzistentní rozdíl mezi těmito dvěma koncepcemi výuky programování“*.

Výsledky závěrečné zkoušky studie Uysala jsou uvedeny v Uysal (2012, s. 821). Autor ke statistickému zpracování dat studijní úspěšnosti použil Mann-Whitney test. Jeho výsledky jsou uvedeny v tabulce 3.

---

<sup>15</sup> Objects-first

**Tabulka 3 – „Výsledky studijní úspěšnosti podle testu Mann-Whitney“ (Uysal, 2012), tabulka převzata z Uysala (2012, s. 821)**

Skupina	n	Průměrné pořadí	Součet pořadí	z	p
Objects- Later	10	6,89	62,00	-2,290	0,022
Objects-First	10	12,80	128,00		

K daným výsledkům a nulové hypotéze autor na straně 820 uvádí, že „*nicméně, jak je patrné z tabulky, je tam mezi experimentálními skupinami statisticky významný rozdíl a nulovou hypotézu odmítneme ( $z = -2.290$ ,  $p < 0.05$ ). Je možné říci, že studenti vyučovaní koncepcí výuky programování objects-first dosáhli lepších výsledků*“.

Autor dále v článku uvádí několik důvodů, proč je, dle jeho názoru, koncepce výuky programování objects-first výhodnější než koncepce výuky programování objects-later. Důvody jsou uvedeny ve studii autora, viz Uysal (2012, s. 820-821).

### **3.5 Zhodnocení výzkumů koncepcí výuky programování**

Autoři výše uvedených studií v předcházející kapitole 3.4.5 dospěli k zajímavým a v některých případech i k odlišným výsledkům. Výsledky daných studií mohou být ovlivněny rozdělením studentů podle určitých kritérií, obsahovou náplní předmětu, vyučujícím, který daný předmět vyučoval, způsobem testování i dalšími faktory, které budou uvedeny v další části této kapitoly. Jednotliví autoři řešili výzkumné studie odlišným způsobem, např. jinou formulací nulové hypotézy nebo jinou přípravou skupin na závěrečný test, proto zobecnění získaných výsledků není možné. Dají se ovšem vysledovat určité společné indicie prezentovaných výsledků.

Vliv na porovnání výsledků dvou koncepcí výuky programování objects-first vs. objects-later má např. způsob testování studentů nebo sledované programátorské konstrukce v daných testech.

V teoretických částech výzkumných studií u Johnsona a Mosese (2008) nebo u Ehlertha a Schulteho (2009) dosáhli studenti koncepce výuky programování objects-first lepších výsledků než studenti koncepce výuky programování

objects-later. U Ehlerta a Schulteho (2009, s. 19) se lepší výsledek skupiny žáků koncepce výuky programování objects-first týkal OO témat (u strukturálních témat byl výsledek jiný) a podle autorů „*zdá se spíše zajímavé, že v navazujícím testu zmizely všechny významné rozdíly*“. Podle informací, které byly zmíněny ve studii Uysala (2012, s. 820), i v jeho výzkumu „*je možné říci, že studenti vyučovaní koncepcí výuky programování objects-first dosáhli lepších výsledků učení*“. Strukturální témata byla sledována pouze ve studii Ehlerta a Schulteho (2009) a v hlavní studii závěrečného testu o něco lépe dopadli studenti koncepce výuky programování objects-first.

U autorů Johnsona a Mosese (2008) a Uysala (2012), jak bylo uvedeno v kapitole 3.4.1, kteří ve svém výzkumu vymezili podobně výzkumnou otázku, která se týkala zkoumání OO konstrukcí, dopadli lépe v části vyžadující naprogramování praktické úlohy studenti koncepce výuky programování objects-first. Autoři Ehlert a Schulte (2009) ve své výzkumné studii praktickou úlohu studentům nepředložili.

Významné rozdíly mezi danými dvěma skupinami koncepcí výuky programování, které byly u autorů Johnsona a Mosese (2008) a Uysala (2012) se týkaly studentů, kteří již navštěvovali vysokou školu. Ve výzkumné studii Ehlerta a Schulteho (2009), ve které mezi danými dvěma skupinami koncepcí výuky programování nebyly významné rozdíly, byli žáci, kteří navštěvovali střední školu. Navíc, žáci střední školy mohou mít méně zkušeností s programováním, než studenti vysoké školy. Předchozí programátorské znalosti mohou mít na celkové výsledky v koncepcích výuky programování objects-first a objects-later vliv. Vliv na výsledek může mít i mentální vyspělost zkoumaných studentů. Vysokoškolští studenti by měli být mentálně vyspělejší o 3 až 4 roky než mladší žáci ze střední školy. Tento faktor by měl být, při interpretaci získaných výsledků, brán v úvahu.

## 4 METODY ZPRACOVÁNÍ A ZPŮSOB ŘEŠENÍ

Výzkumný problém a jeho formulace ve formě výzkumných otázek byl vymezen v kapitole Úvod. Stanovení hlavního cíle a dílčích cílů na základě výzkumných otázek a následného stanovení výzkumných hypotéz zamýšleného výzkumu bylo vymezeno v kapitole 2 Cíl disertační práce. V této kapitole bude nejdříve stanovena strategie zamýšleného výzkumu s následným uvedením výzkumných metod a technik, dále bude uveden způsob měření získaných dat, budou uvedeny metody zpracování získaných dat a budou uvedeny možnosti testování studentů v předmětu programování.

### 4.1 Strategie výzkumu

Pelikán (1998, s. 69) uvádí, že „...součástí každého výzkumu je i vytvoření toho, co bychom mohli nazvat výzkumnou strategií. Ta je základem každého dobrého projektu výzkumu“.

Dále autor uvádí „...dva základní typy výzkumů. Jde o tzv. **ex post facto výzkum** a o **experimentální výzkum**“. Podle autora „jde tedy o typ výzkumu, v němž v podstatě hledáme příčinu relativně známého stavu“.

„**Ex post facto výzkum** lze definovat jako „výzkum, ve kterém se nezávisle proměnná či proměnné už objevily a ve kterém badatel začíná s pozorováním závisle proměnné či proměnných. Studuje pak nezávisle proměnné retrospektivně pro jejich možné vztahy a účinky na nezávisle proměnnou či proměnné““ (Pelikán, 1998, s. 70).

„Naproti tomu **experimentální výzkum** volí zcela opačný postup. Tento přístup ověřuje, zda předpokládaná intervenující proměnná vyvolává v případě, kdy ji experimentátor mění, adekvátní změny v proměnné, kterou badatel považuje za proměnnou na ní závislou“ (Pelikán, 1998, s. 70).

**Vzhledem k povaze zamýšleného vlastního výzkumu této disertační práce, bude typem výzkumu v této disertační práci experimentální výzkum.**

## 4.2 Výzkumné metody a techniky

Podle Pelikána (1998, s. 95) „výzkumná metoda je, podle našeho názoru, obecným metodologickým nástrojem k získání a zpracování dat, vymezujícím širší a komplexnější úhel pohledu na šetřenou problematiku“.

Autor dále uvádí, že „výzkumné metody lze rozdělit do tří kategorií: Metody empirického výzkumu, zaměřené na sběr a analýzu výzkumných dat, metody teoretické analýzy a metody historicko-srovnávacího výzkumu“.

Autor také dále uvádí na straně 97 „metody a techniky používané v pedagogickém empirickém výzkumu...“.

Vzhledem k zamýšlenému výzkumu a přehledu metod a technik uvedených v Pelikánovi (1998, s. 97-98) byly vybrány následující metody a techniky (seřazené abecedně):

- Explorační metoda
  - dotazník
- Experimentální metoda
  - přirozený, neboli terénní experiment
- Testy

### 4.2.1 Dotazník

Podle Pelikána (1998, s. 104) „dotazník je snad nejpoužívanější pedagogickou výzkumnou technikou vůbec“.

„Dotazníkem lze současně a stejnou formou, což je pro zpracování důležité, oslovit i velký počet respondentů“ (Pelikán, 1998, s. 105).

Podle autora existují i nevýhody, kdy ale „musíme počítat především se subjektivitou výpovědí...“.

Jednotlivé typy otázek používaných v dotazníku jsou uvedeny v Pelikánovi (1998, s. 107-112).

Ve vlastním výzkumu v disertační práci dostali na konci předmětu studenti k vyplnění dotazníkové šetření. Více o dotazníkovém šetření je uvedeno v kapitole 4.2.3 Testy.

#### 4.2.2 Přirozený experiment

*„Specifikou tohoto typu experimentu je využití přirozené životní situace k tomu, aby při potřebném stupni kontroly mohly být zaváděny a ovlivňovány proměnné, které považuje experimentátor za důležité pro změnu konečného stavu“ (Pelikán, 1998, s. 227).*

Mezi přednosti přirozeného experimentu autor na straně 229 uvádí *„...jeho blízkost přirozenému životu“*.

Mezi nevýhody přirozeného experimentu autor uvádí, *„...že v přirozeném prostředí nedokážeme vždy mít pod kontrolou možné vnější vlivy, které mohou vstoupit do experimentu“*.

Přirozený experiment ve výzkumu této disertační práce je detailně uveden v kapitole 5 Průběh pedagogického experimentu.

#### 4.2.3 Testy

Pelikán (1998, s. 153) uvádí, že *„testy patří mezi velice často užívané výzkumné techniky nejen v pedagogice, ale zejména v psychologii a některých blízkých oborech“*.

*„Dalším problémem je otázka validity a reliability testů. I u renomovaných testů se nezdá, že měří něco poněkud jiného, než co být měřeno má“ (Pelikán, 1998, s. 154).*

Podle autora *„v pedagogickém empirickém výzkumu se setkáváme se třemi základními skupinami testů...“*. Z vymezení autora na stranách 154-155 budou dále uvedeny pouze testy didaktické.

#### Didaktické testy

*„Didaktické testy jsou konstruovány jako poměrně přesný měřicí nástroj, který byl navržen, ověřen a používán ke zjištění (měření) výsledků výuky. Kromě toho má každý standardizovaný test stanovené i normy, podle nichž se vyhodnocují výsledky testu, takže výstupy jsou srovnatelné a kvantifikovatelné“ (Pelikán, 1998, s. 172).*

Autor dále klasifikuje didaktické testy podle určitých kritérií, které jsou uvedeny na stranách 172-175.

Podle zaměření výzkumu disertační práce budou dále vymezeny CR testy, které byly při výzkumu ověřování znalostí studentů použity.

### **CR testy**

Pelikán (1998, s. 175) uvádí, že, „...CR testy jsou testy, které zjišťují výkon žáka nebo studenta v dané oblasti. Tyto testy měří, jak student zvládl určitý úsek učiva. ...CR testy mohou, ale nemusí mít stanoveny normy, podle nichž je test vyhodnocován. Spíše se častěji vymezuje hranice minimálního výkonu...“.

V každém didaktickém testu mohou být, podle způsobu odpovědi dotazovaného, různé typy úloh. Pelikán (1998, s. 175) je klasifikuje jako „*druhy testových úloh*“. Níže je uvedeno základní rozdělení druhů testových úloh podle autora s jejich základní charakteristikou.

### **Otevřené úlohy**

„...testovaný odpovídá sám, bez možnosti výběru určitých variant odpovědí...“ (Pelikán, 1998, s. 175).

Dále autor na straně 176 dělí otevřené úlohy:

- „*otevřené široké úlohy*,
- *otevřené úlohy se stručnou odpovědí*“.

### **Uzavřené úlohy**

„...kde je mu<sup>16</sup> nabídnuto několik možností odpovědí“ (Pelikán, 1998, s. 175).

Dále autor dělí na straně 176 uzavřené úlohy:

- „*dichotomické úlohy*,
- *úlohy s výběrem více odpovědí*,
- *úlohy přiřazovací*,
- *úlohy uspořádací*“.

Testování studentů ve vlastním výzkumu disertační práce se skládá ze dvou částí, části praktické a části teoretické.

V praktické části, je možné ji označit také jako část programovou, respektive programátorskou, jsou použity otevřené široké úlohy, kde studenti musí naprogramovat zadanou úlohu.

---

<sup>16</sup> Testovanému

V teoretické části jsou typy otevřených i uzavřených úloh. U otevřených odpovědí se jedná o typ otevřených úloh se stručnou odpovědí (doplňovací úlohy). U uzavřených odpovědí se jedná o úlohy s výběrem odpovědí, úlohy přiřazovací a úlohy uspořádací.

Více o testování studentů ve vlastním výzkumu disertační práce je uvedeno v kapitole 5.1.4 Testy.

Způsob testování studentů v programování je uveden v kapitole 4.4 Možnosti ověřování studentských programátorských znalostí. Protože se může jednat o specifické možnosti testování studentů, je této problematice vyhrazena speciální kapitola.

### 4.3 Způsob měření získaných dat v testech

*„Jestliže chceme při studiu pedagogické reality uplatňovat vědecký přístup, je třeba, abychom u každého studovaného jevu dokázali vedle postižení jeho kvality zachytit i jeho kvantitu, tj. zachytit jeho velikost nebo množství měřením“* (Chráška, 2003, s. 33-34).

Chráška (2003, s. 34) cituje Kerligeru (1972), že měření je *„přiřazování čísel předmětům nebo jevům podle pravidel“*.

*„...jde o to, nalézt funkci (pravidlo) pro přiřazování prvků množiny měřených objektů k prvkům množiny čísel“* (Chráška, 2003, s. 34).

Podle autora se může stát, že *„...proměnná, kterou chceme zachytit, není přímo měřitelná...“*. Autor dále uvádí, že *„...se často uchylujeme k měření tzv. indikátorů (ukazatelů), tj. jiných jevů, které s velkou pravděpodobností s danou proměnnou souvisejí“*.

Autor také rozlišuje na straně 35 *„...čtyři úrovně měření: měření nominální (klasifikace), měření ordinální (pořadové), měření intervalové a měření poměrové“*.

K měření výsledků z testů, výzkumu v této disertační práci, bylo zvoleno **měření intervalové**.

Při intervalovém měření *„...přiřazujeme čísla tak, že vyjadřují, jak velké jsou mezi nimi rozdíly. Tento druh měření však nemá přirozený nulový bod...“* (Chráška, 2003, s. 36).



Autor dále uvádí, že „*příkladem intervalového měření je např. měření úrovně vědomostí didaktickým testem (platí přibližně)*“.

Podle autora „...*je možné používat Studentův t-test, párový t-test ....*“.

V testech v disertační práci je jednotlivým praktickým nebo teoretickým úlohám přiřazeno číslo z intervalu podle míry splnění zadané úlohy.

„*Při posuzování vlastností měření nás obvykle nejvíce zajímá jeho validita, reliabilita a praktičnost*“ (Chráska, 2003, s. 37).

Autor uvádí, že „*českým ekvivalentem pojmu validita je platnost. Měření má dobrou validitu tehdy, jestliže měří skutečně to, co podle předpokladu měřit má*“.

Autor také dále uvádí na stranách 37-38, že „*podle toho, k čemu se validita vztahuje, lze rozlišit validitu:*

- *obsahovou (posuzujeme, do jaké míry se měří stanovený obsah),*
- *souběžnou (posuzujeme, do jaké míry se měření shoduje s jiným měřením týchž objektů),*
- *predikční (posuzujeme, do jaké míry provedené měření vypovídá o budoucím vývoji objektů),*
- *konstruktovou (pojmovou, teoretickou), u které posuzujeme, do jaké míry ovlivňuje výsledky provedeného měření nějaký faktor – konstrukt*“.

Validita měření v testech ve výzkumu disertační práce bude dodržena hodnocením základních konstrukcí typických v daných koncepcích výuky programování. V testech a jejich vyhodnocení je rozlišena validita obsahová a souběžná.

Validita měření v dotazníkovém šetření, ve vlastním výzkumu v disertační práci, bude dodržena hodnocením studentů možného vlivu vyučujícího na průběh výuky, možného dopadu na výsledky studentů jejich samostudiem, popřípadě studiem jiných materiálů, než byly k dispozici ve výuce.

„*Pojem reliabilita se často nahrazuje pojmy spolehlivost, stabilita, homogenita, přesnost, konzistence nebo stálost, avšak žádný z nich pojem reliability plně nevystihuje. Aby měření bylo reliabilní, je třeba, aby při opakování za stejných*

*podmínek poskytovalo stejné (zhruba stejné) výsledky. Tento aspekt reliability je možno označit jako spolehlivost měření“ (Chráška, 2003, s. 38).*

*Podle autora „dostatečně vysoká reliabilita je nutnou podmínkou dobré validity měření, vysoká reliabilita avšak ještě nezaručuje dobrou validitu“.*

S ohledem na minimalizaci chyb a přesnost měření budou výsledky testů vyhodnoceny jedním vyučujícím. Dané testy studentů budou hodnoceny v jedné časové jednotce, aby byly minimalizovány chyby případného hodnocení ohledně jiného bodování stejných sledovaných konstrukcí.

#### **4.4 Možnosti ověřování studentských programátorských znalostí**

Možnosti ověřování studentských programátorských znalostí v této kapitole 4.4 byly částečně publikovány v článku Hubálovský a Kořínek (2015b). Pokud nebude v této kapitole uvedeno jinak, možnosti ověřování studentských programátorských znalostí byly stanoveny autorem disertační práce na základě jeho znalostí a zkušeností, které získal jako student nebo vyučující v předmětech programování.

I když způsobů, jakým se dají ověřit znalosti studentů v předmětech programování je velké množství, velice důležité je i jejich následné vyhodnocení.

Vyhodnocení studentských programátorských znalostí je možno následujícími způsoby:

- vyučujícím
- automatickým vyhodnocením

V následujících podkapitolách jsou uvedeny nejčastější možnosti ověřování studentských programátorských znalostí/úloh podle možnosti vyhodnocení (vyučujícím/automaticky).

##### **4.4.1 Vyhodnocování vyučujícím**

Mezi nejčastější ověřovací úlohy v programování s následným vyhodnocením vyučujícím patří naprogramování zadané úlohy v konkrétním programovacím jazyku, napsání řešení zadané úlohy/části úlohy na papír nebo ověření znalostí

testem, ve kterém mohou být doplňující otázky nebo otázky s výběrem z několika možností.

### **Naprogramování úlohy v programovacím jazyce**

Nejčastějšími ověřovacími úlohami bývají úlohy na naprogramování programu v příslušném programovacím jazyku. Zkontrolování naprogramované úlohy zabere učiteli poměrně hodně času, protože musí často ověřit funkčnost odevzdané úlohy. Tím to pro něj ovšem zdaleka nekončí. Důležitou součástí výsledné klasifikace bývá i způsob naprogramování, jakým student úlohu naprogramoval. Úloha, která vypisuje možná i přesvědčivá data, nemusí být naprogramována optimálně. Pro dostatečnou zpětnou vazbu studenta musí vyučující projít detailně kód. Detailní procházení kódu stojí vyučujícího další čas. Při opravování velkého množství úloh může dojít k neobjektivitě. Ta může být způsobena únavou nebo stereotypem dlouhého opravování. Tento způsob ověřování znalostí však dává studentovi kvalitní zpětnou vazbu a může si z ní vzít ponaučení.

### **Ověřování programátorských znalostí psaných na papír**

Dalším typem ověřovacích úloh mohou být programátorské konstrukce psané na papír. K tomuto způsobu ověřování může vést vyučujícího několik důvodů. Vyučující může chtít zjistit, jak se studenti naučili základní příkazy. Nemusí chtít, aby se studenti rozptylovali vývojovým prostředím. K tomuto typu ověřování může vést i typ vyučovací hodiny, kdy při výuce programování nemusejí studenti pracovat na počítačích, ale může se jednat o hodinu teorie ve třídě. Při tomto ověřování znalostí nemusí vyučující danou úlohu spouštět a odlaďovat ve vývojovém prostředí. Na druhé straně více práce a času vyučujícímu zabere ověřování kódu. Navíc může nějaké chyby nebo chybné konstrukce přehlédnout a náchylnost k chybám je tedy daleko větší. Mezi studenty obvykle nebývá toto ověřování znalostí moc oblíbené.

### **Ověřování programátorských znalostí pomocí testu**

Dalším typem ověřovacích úloh bývají programátorské úlohy ve formě testu. Test může nejčastěji obsahovat doplňující otázky nebo otázky s výběrem odpovědi z několika možností.

### **Výběr z několika možností**

Výběr správné odpovědi z několika možností nemusí mít vypovídající hodnotu, protože tento typ testu nezjistí, jak student rozumí dané problematice. Při zaškrtnutí správné odpovědi u otázky vyučující nepozná studentovo uvažování při sestavování programu. U této možnosti testování záleží i na typu otázek, kterých může být více druhů. Výběr jedné z několika možných odpovědí má svá úskalí. Mezi největší nedostatek této možnosti ověřování znalostí je studentem náhodné uhodnutí správné odpovědi. Výběr správné odpovědi lze navíc provést vyloučením ostatních možností. Možností, jak náhodnému tipování zabránit je např. odečítáním bodů za chybnou odpověď, což ale může být pro studenty demotivující. Může se tím ale eliminovat náhodné uhodnutí správné odpovědi.

### **Doplňující otázky**

V tomto typu příkladů mají studenti za úkol doplnit chybějící části úlohy. Doplnující otázky mohou mít větší vypovídající hodnotu než otázky s výběrem z několika možností. Studenti mohou doplňovat:

- části definic
- hodnoty proměnných po vykonání kódu programu/algoritmu
- části kódu programu/algoritmu

Z výše uvedených možností mají větší vypovídající hodnotu poslední dvě možnosti. Proměnná je základním stavebním prvkem programování. Postupné vykonávání programu/algoritmu s přiřazováním proměnných různé hodnoty a následné vypisování hodnot proměnných v částech programu prověří studentovo chápání programových konstrukcí.

Doplnění části kódu programu/algoritmu má nejvyšší vypovídající hodnotu z úloh o doplňování. Tento typ úloh může dát vyučujícímu informace o tom, jaké má student znalosti a jaký má způsob uvažování. Student musí pochopit podstatu problému. Teprve po pochopení může doplnit chybějící části předloženého kódu. Tento typ ověření má ovšem také několik úskalí. Pokud jsou části kódu psané na papír, tak si student nemůže vyzkoušet daný kód, což může vést k chybnému zapsání některých příkazů. Pokud studenti dostanou úlohu částečně naprogramovanou v programovacím jazyku, výše uvedený problém je tím minimalizován. Při

programování úloh není pouze jediné řešení správné. Jeden problém může být vyřešen různými způsoby. Některé způsoby mohou být navíc daleko efektivnější než ostatní. Zadání konkrétní části programu směřuje studenta k jednomu konkrétnímu řešení. On by ale příklad řešil jiným způsobem. Předložené řešení v otestování ho může limitovat. Na druhé straně ale může dobře ověřit jeho schopnost myšlení a uvažování při programování.

#### **4.4.2 Automatické vyhodnocování**

Pro automatické vyhodnocení testových úloh existují různá prostředí. Mezi nejčastěji automaticky vyhodnocované testové úlohy patří testové úlohy s doplňujícími otázkami nebo otázkami s výběrem z několika kategorií.

Automatické vyhodnocení studentských programátorských znalostí pomocí vývojového prostředí je možné také pomocí koncepce výuky programování Design Patterns First, jak uvádí Pecinovský (2007a).

Dále v kapitole jsou ukázány možnosti testování pomocí univerzálního testovacího prostředí, které se používalo v teoretické části testů pro vyhodnocování studentských úloh na Přírodovědecké fakultě UHK v předmětu Programování 1 v akademických letech 2013/2014 a 2014/2015.

#### **Vyhodnocení metodou výuky programování Design Patterns First**

Podle Pecinovského (2007a) „*automatické vyhodnocení odevzdaných řešení se však výrazně zjednoduší ve chvíli, kdy jedním z požadavků úlohy bude implementace nějakého rozhraní. Pak totiž nebudeme muset řadu věcí kontrolovat my, protože je za nás zkontroluje překladač, a my se budeme moci soustředit pouze na vyhodnocení vlastního řešení.*“

#### **Univerzální testovací prostředí**

Vytvoření univerzálního testovacího prostředí s dalšími informacemi i možnostmi testování je uvedeno v disertační práci autora systému Petra Voborníka, viz Voborník (2012a).

Určité informace o univerzálním testovacím prostředí jsou uvedeny i na webových stránkách autora systému, viz Voborník (2012b).

Pomocí univerzálního testovacího prostředí lze vytvořit škálu otázek různých typů. Některé z těchto typů otázek byly použity v teoretické části testů studentů ve výzkumu uvedeném v této disertační práci. Autor disertační práce se také seznámil s vytvářením otázek pomocí univerzálního testovacího prostředí.

Typy jednotlivých otázek jsou uvedeny např. ve Voborníkovi (2012b).

V testovacím prostředí mohou být vytvořeny doplňovací otázky, otázky na výběr z několika možností. Platí pro ně stejné výhody a nevýhody, jaké byly uvedeny v příslušných podkapitolách kapitoly 4.4.1 Vyhodnocování vyučujícím. Výhoda testovacího prostředí spočívá především v okamžitém vyhodnocení vyplněných odpovědí. Student se tak může velmi brzy seznámit s chybami, kterých se v testu dopustil a ve velmi krátkém čase se z nich ponaučit. U tohoto typu testování odpadá také možná neobjektivita hodnocení vyučujícím. Navíc pro vyučujícího odpadá i časová náročnost opravy. Zadání úloh testu může být vybíráno zcela náhodně univerzálním prostředím. Zkoušející do univerzálního prostředí pouze nastaví parametry testování: Kolik a jaké otázky se budou náhodně vybírat z daných okruhů. Každý ze studentů může mít samostatné zadání testu. Jak již bylo uvedeno, studenti si mohou ihned po ukončení testování prohlédnout rozbor testu. (Voborník, 2012b)

## 4.5 Metody zpracování dat v pedagogickém výzkumu

V této podkapitole budou uvedeny metody (způsob statistického zpracování) zpracování dat získaných pomocí testů v pedagogickém výzkumu. Získaná data pro výzkum disertační práce byla zpracována programem pro statistiku Statistica 12.

Chráška (2003, s. 41) uvádí „nejběžnější metody zpracování dat v pedagogických výzkumech“.

Podle autora na straně 78 „mezi statistickými postupy používanými při ověřování platnosti (testování) hypotéz mají velký význam statistické testy významnosti. Jsou to postupy (procedury), pomocí nichž ověřujeme, zda mezi proměnnými existuje vztah (závislost, rozdíl)“.

„Rozhodování ve statistických testech významnosti má vždy pravděpodobnostní charakter (nikdy si nejsme svým rozhodnutím beze zbytku jisti). Pravděpodobnost (riziko), že neoprávněně odmítneme nulovou hypotézu (a tudíž nesprávně přijmeme alternativní hypotézu), se nazývá hladina významnosti

(signifikance)“ (Chráska, 2003, s. 78). Ve statistickém zpracování testů v disertační práci byla různými metodami volena její hodnota (hladiny významnosti) 5%, tj.  $\alpha = 0.05$ .

Autor na straně 79 rozděluje statistické testy významnosti na testy parametrické a testy neparametrické.

K parametrickým testům autor uvádí, že „*parametrické testy vyžadují splnění řady předběžných podmínek a požadavků... Vyžaduje se např., aby rozdělení náhodné veličiny bylo určitého typu (nejčastěji se požaduje, aby rozdělení bylo normální)*“. Autor dále uvádí, že „*statistické testy významnosti, u nichž se nulová hypotéza netýká parametrů rozdělení, nýbrž nějaké jiné, obecné vlastnosti rozdělení, označujeme jako **testy neparametrické***“. K nim autor dodává, že „*větší univerzálnost neparametrických testů významnosti je však vykoupena jejich menší statistickou účinností*“.

Podle rozdělení náhodné veličiny byl ke statistickému zpracování v disertační práci použit buď parametrický test (Studentův t-test) nebo neparametrický test (U-test Manna a Whitneyho), pro opakované testování studentů byl v disertační práci použit Wilcoxonův test.

Rozdělení náhodné veličiny bylo zjišťováno pomocí grafu typu Q-Q v programu Statistica 12, „*protože jsou tyto grafy významným a hojně využívaným pomocníkem pro vizuální kontrolu předpokladů na rozdělení zkoumané náhodné veličiny...*“ (Jak se pozná, 2013).

Grafy typu Q-Q jsou „*...založené na porovnání kvantilů teoretického rozdělení a naměřených kvantilů...*“ (Jak se pozná, 2013).

Pro vlastní výzkum, jak spolu těsně souvisejí dva jevy (výsledky studentů z pretestu a jednotlivých částí testů), byl stanoven Spearmanův koeficient pořadové korelace.

#### **4.5.1 Wilcoxonův test**

„*Tento statistický test významnosti se používá v podobných situacích jako znaménkový test, tedy v případě opakovaných měření týchž objektů. Podmínkou pro jeho použití je, že data, se kterými se pracuje, musí být alespoň ordinální (pořadová)*“ (Chráska, 2003, s. 119).

#### 4.5.2 U-test Manna a Whitneyho

*„Je to velmi vydatný neparametrický test, který lze použít v případech, kdy máme rozhodnout, zda dva výběry mají zhruba stejné rozdělení četností, zda tedy mohou pocházet ze stejného základního souboru“ (Chráska, 2003, s. 121).*

#### 4.5.3 Studentův t-test

*„...je jedním z nejznámějších statistických testů významnosti pro metrická data. Pomocí Studentova t-testu můžeme rozhodnout, zda dva soubory dat, získané měřeními na dvou různých souborech objektů (např. žáků), mají stejný aritmetický průměr“ (Chráska, 2003, s. 149).*

Autor dále uvádí na straně 155, že *„studentův t-test patří mezi parametrické testy významnosti“*.

#### 4.5.4 Spearmanův koeficient pořadové korelace

*„Tento statistický postup lze využít v případech, kdy máme rozhodnout, jak těsně spolu souvisí dva jevy, které byly zachyceny (změřeny) pomocí ordinálního měření. Často např. můžeme u skupiny žáků pozorovat vlastnosti, které nám umožňují seřadit je podle míry této vlastnosti“ (Chráska, 2003, s. 108).*

Autor dále uvádí, že *„...spearmanův koeficient pořadové korelace umožňuje stanovit kvantitativně, jak dalece jsou si dvě vytvořená pořadí podobná, a tím vlastně určit, jak těsná je souvislost mezi jevy, na základě nichž byla pořadí vytvořena“*.

Autor také dále uvádí na straně 110, že *„koeficient  $r_s$  může nabývat hodnot od 0 do +1 nebo -1. Hodnota 0 vypovídá o tom, že mezi srovnávanými jevy není žádný vztah. Čím více se vypočítaná hodnota koeficientu korelace blíží hodnotě 1 (nebo -1), tím těsnější je vztah mezi jevy, které srovnáváme. Kladný výsledek vypovídá o tom, že vyšším hodnotám u jednoho měřeného jevu odpovídají také spíše vyšší hodnoty u druhého jevu a zároveň nižším hodnotám u prvního jevu odpovídají také nižší hodnoty u jevu druhého. Jestliže je koeficient korelace záporný, znamená to, že mezi jevy, které srovnáváme, je negativní (opačný) vztah...“*.

*„Přibližná interpretace hodnot korelačního koeficientu“ (Chráska, 2003, s. 111) je uvedena v tabulce 4.*



**Tabulka 4 – „Přibližná interpretace hodnot korelačního koeficientu (Chráška, 2003, s. 111)“**  
**Tabulka převzata z (Chráška, 2003, s. 111)**

<b>Koeficient korelace</b>	<b>Interpretace</b>
$r = 1$	naprostá závislost (funkční závislost)
$1,00 > r \geq 0,90$	velmi vysoká závislost
$0,90 > r \geq 0,70$	vysoká závislost
$0,70 > r \geq 0,40$	střední (značná) závislost
$0,40 > r \geq 0,20$	nízká závislost
$0,20 > r \geq 0,00$	velmi slabá závislost
$r = 0$	naprostá nezávislost

## 5 PRŮBĚH PEDAGOGICKÉHO EXPERIMENTU

V této kapitole je podrobně popsán průběh pedagogického experimentu v předmětu Programování 1. Budou zde vysvětleny fáze pedagogického experimentu, mezi které patří:

- kritéria, podle kterých byli studenti rozděleni do dvou skupin
- obsahová náplň předmětu programování zkoumaných dvou skupin studentů (použití vývojového prostředí, programovací jazyk, způsob výuky ve vývojovém prostředí, zvolené koncepce výuky programování, vyučující v daných skupinách)
- způsob testování studentů

Pedagogický experiment probíhal v akademických letech 2013/2014 a 2014/2015. V tabulce 5 je uvedeno podrobné načasování etap pedagogického experimentu v předmětu Programování 1.

Částečný popis pedagogického experimentu v předmětu Programování 1 a ukázka možného zadání posttestu bylo publikováno v článku Hubálovský a Kořínek (2015a) nebo v článku Hubálovský a Kořínek (2015c).

Pedagogický experiment v disertační práci se od výzkumů autorů různých studií, které byly uvedeny v kapitole 3.4, liší. V této práci je rozdělení studentů do skupin podle různých kritérií odlišné. Bylo zvoleno i jiné vývojové prostředí a programovací jazyk. Studenti byli zkoušeni komplexně pomocí různých typů úloh, které jsou uvedeny v dalších částech této kapitoly.

S ohledem na výzkumy různých autorů koncepcí výuky programování objects-first a objects-later, které byly uvedeny v kapitole 3.4. Obsahová náplň předmětu Programování 1 vycházela ze studie autorů Ehlerta a Schulteho (2009) s určitými odlišnostmi. V předmětu Programování 1 nebyla vyučována dědičnost, ani nebyly vyučovány abstraktní metody a asociace. Základní konstrukce byly vyučovány s jednorozměrným a vícerozměrným polem. Detailnější popis koncepcí výuky programování objects-first a objects-later, jejichž částečný popis byl uveden např. v článku Hubálovský a Kořínek (2015c), jsou uvedeny v následujících podkapitolách.

Tabulka 5 – Přehled načasování etap pedagogického experimentu

Akademický rok	Pretest	Koncepte výuky programová ní (skupina studentů)	Týdny v semestru (letní semestr) – předmět Programování 1					
			1	2-6	7	8-12	13	Zkouškové období
2013/2014	Zkouškové období zimního semestru 2013/2014	objects-first	Úvodní seznámení s předmětem	Témata koncepte výuky programování objects-first	Posttest	Společně probíraná témata	Retest 1	Retest 2
		objects-later		Témata koncepte výuky programování objects-later				
2014/2015	Zkouškové období zimního semestru 2014/2015	objects-first	Úvodní seznámení s předmětem	Témata koncepte výuky programování objects-first	Posttest	Společně probíraná témata	Retest 1	Retest 2
		objects-later		Témata koncepte výuky programování objects-later				

## 5.1 Pedagogický experiment - pretest

Studenti byli rozděleni do dvou skupin podle výsledků ze zkouškového testu z předmětu ALGDS, který je pretestem pedagogického experimentu.

V předmětu ALGDS, jehož popis byl částečně uveden např. v článku Hubálovský a Kořínek (2015a), studenti na PŘF UHK procvičovali algoritmické úlohy se základními algoritmickými konstrukcemi. Tyto konstrukce se uplatňují v koncepcích výuky programování jak objects-later tak i objects-first.

Výuka v předmětu ALGDS probíhala v pseudokódu jazyku Pascal. Důraz byl na začátku předmětu kladen na algoritmy využívající jednoduchou proměnnou. Po algoritmech využívajících jednoduchou proměnnou byl kladen důraz na algoritmy se strukturovanou proměnnou, polem.

V předmětu se studenti seznámili pouze s několika základními příkazy, které při vytváření algoritmů využívali. Práce s jednoduchou proměnnou byla demonstrována na základních algoritmických konstrukcích (sekvence, větvení a cyklus). I při práci s proměnnou typu pole se využívaly základní algoritmické konstrukce, které by měli mít již studenti osvojené z příkladů s jednoduchou proměnnou.

Pro výuku algoritmů, která byla uvedena v kapitole 3, platí výhody i nevýhody, které byly uvedeny v uvedené kapitole. V předmětu ALGDS byla menší atraktivita a nemožnost si vyzkoušet dané algoritmy minimalizována využíváním programu Algoritmy.

V předmětu studenti psali 2 testy: zápočtový a zkouškový. Zápočtový test obsahoval pouze příklady na práci s jednoduchou proměnnou. Zkouškový test obsahoval příklady na práci se strukturovanou proměnnou typu pole. K úspěšnému získání zápočtu studenti potřebovali získat 60% možných bodů ze zápočtového testu. Zkouškový test mohli psát pouze ti studenti, kteří uspěli v zápočtovém testu. Ve zkouškovém testu byla úspěšnost stanovena také na 60% možných bodů z testu.

Testy obsahovaly několik typů úloh: vytvoření algoritmu, dopsání příkazů neúplného algoritmu a napsání hodnot daných proměnných po vykonání předloženého algoritmu.

Výhodou výuky základních algoritmických konstrukcí je srovnání počáteční úrovně studentů. Ti na vysokou školu přichází s různými znalostmi a dovednostmi.

Studenti byli do dvou skupin v pedagogickém experimentu rozděleni podle výsledků ze zkouškového testu z předmětu ALGDS. V tomto předmětu byli studenti zkoušeni komplexně, protože museli uspět v obou testech a v ústní zkoušce. Srovnatelnost skupin v pedagogickém experimentu podle výsledků zkouškového testu z předmětu ALGDS byla dána také brzkou návazností předmětů ALGDS a Programování 1.

### 5.1.1 Rozdělení studentů do skupin

Jak bylo uvedeno v předchozí kapitole, studenti byli náhodně rozděleni do dvou skupin. Na začátku semestru byl v obou skupinách stejný počet studentů. Vzhledem k nedisciplinovanosti některých studentů, kteří na předmět přestali docházet a neabsolvovali testy nebo v průběhu semestru ukončili studium, není v akademických letech v obou skupinách stejný počet studentů.

První skupina studentů, experimentální skupina, se vyučovala podle koncepce výuky programování objects-first. Druhá skupina studentů, kontrolní skupina, se vyučovala podle koncepce výuky programování objects-later. Počet studentů ve skupinách v jednotlivých akademických letech je uveden v tabulce 6.

**Tabulka 6 – Počty studentů ve skupinách v různých akademických letech**

<b>Akademický rok</b>	<b>Počet studentů – skupina koncepce výuky programování objects-first</b>	<b>Počet studentů – skupina koncepce výuky programování objects-later</b>
2013/2014	14	14
2014/2015	16	18

Jednotlivé skupiny vyučovali různí vyučující. Vyučující měli pravidelné konzultace ohledně obsahové náplně předmětu, použití vyučovacích metod v hodinách a struktury kurzu s ohledem na studijní materiály. Materiály, mezi které patřily např. výukové prezentace, příklady ze cvičení a domácí úkoly, byly studentům k dispozici v prostředí Moodle a byly strukturovány podle probíraných témat v jednotlivých vyučovacích hodinách. Jedním z vyučujících byl autor disertační práce, druhým vyučujícím byl Ing. Petr Voborník, Ph.D. V akademickém roce 2013/2014 skupinu studentů vyučovaných podle koncepce výuky programování

objects-first vyučoval autor disertační práce, skupinu studentů vyučovaných podle koncepce výuky programování objects-later vyučoval Ing. Petr Voborník, Ph.D. V následujícím akademickém roce si vyučující jednotlivé skupiny prohodili, tj. autor disertační práce vyučoval skupinu studentů vyučovaných podle koncepce výuky programování objects-later a Ing. Petr Voborník, Ph.D. vyučoval skupinu studentů vyučovaných podle koncepce výuky programování objects-first.

Každá vyučovací jednotka v jednotlivých týdnech se skládala ze tří vyučovacích hodin, kde byly dvě hodiny cvičení a jedna hodina teorie. Teorie se probírala v rámci cvičení s brzkou návazností na praktické programátorské příklady.

### **5.1.2 Zvolené vývojové prostředí a programovací jazyk**

Zvolené vývojové prostředí k výuce programování v předmětu Programování 1 bylo Visual Studio a programovacím jazykem byl zvolen C#.

Výuka ve vývojovém prostředí probíhala v tzv. konzolové aplikaci. Zobrazení v konzolové aplikaci je textové. K volbě nás vedly následující předpoklady:

- studenti se nemusejí rozptylovat grafickým rozhraním programu a různými nabídkám
- při programování v textovém režimu se studenti mohou zaměřit na programátorské konstrukce
- s ohledem na zkušenosti studentů z předmětu ALGDS bylo výhodné některé konstrukce a typy příkladů, se kterými se již studenti setkali, názorně demonstrovat v programovacím jazyku v textovém prostředí

### **5.1.3 Obsahová náplň jednotlivých vyučovaných skupin studentů**

Jak již bylo uvedeno, první skupina studentů se vyučovala podle koncepce výuky programování objects-first. Druhá skupina studentů se vyučovala podle koncepce výuky programování objects-later. Náplň obou skupin studentů se lišila jiným pořadím probíraných úvodních témat. Studenti koncepce výuky programování objects-first se více zabývali OO konstrukcemi. Studenti koncepce výuky programování objects-later se více zabývali strukturovanými konstrukcemi. Příklady v obou koncepcích výuky programování byly zaměřené na práci s jednorozměrným

a vícerozměrným polem. Na cvičení studenti programovali příklady s konstrukcemi využívajícími pole, se kterými se již setkali v předmětu ALGDS. V obou skupinách byl studentům demonstrován praktický příklad ve formě jednoduchého bludiště s braním předmětů. Tento praktický příklad demonstroval využití vícerozměrného pole.

Společně probíraná témata obou koncepcí výuky programování jsou označena v tabulce 5 stejnou barvou, odlišná témata jsou vyznačena jinou barvou. Jiná obsahová náplň koncepcí výuky programování objects-first a objects-later byla náplní předmětu Programování 1 v semestru od druhého do šestého týdne. Od osmého týdne byla obsahová náplň v předmětu v obou skupinách stejná. Seznam probíraných témat v obou skupinách je uveden v tabulce 7 i s barevným vyznačením, které bylo uvedeno v tabulce 5.

**Tabulka 7 – Obsahová náplň kontrolní skupiny studentů (objects-later) a experimentální skupiny studentů (objects-first)**

Skupina objects-later	Skupina objects-first
Základní příkazy	Základy OOP v C#
Jednorozměrné pole	Základní příkazy
Matice	Jednorozměrné pole
Čtvercová matice/ Základy OOP v C#	Matice
Základy OOP v C#	Čtvercová matice
Řazení Soubory Binární soubory, souborový systém Parametry programu	

Obsahová náplň předmětu byla vytvořena vyučujícími předmětu Programování 1. Vyučující se na náplni předmětu Programování 1 podíleli stejnou měrou, ať již ve formě příkladů na cvičení nebo teorie ve formě prezentací.

### **Koncepce výuky programování objects-first**

Oproti vymezení koncepce výuky programování objects-first v kapitole 3 nebyla v pedagogickém experimentu v předmětu Programování 1 vyučována dědičnost ani asociace. Při vymezení obsahové náplně skupiny studentů, kteří se vyučovali podle koncepce výuky programování objects-first, byla brána v potaz definice OOP Drbala

(1994, s. 3), která byla uvedena jako první definicí v kapitole **Definice OOP** v disertační práci na straně 25.

Důraz v koncepci výuky programování objects-first byl kladen především na následující OO konstrukce:

- Vytváření tříd
- Vytváření instancí
- Definice konstruktoru
- Definice datových položek
- Definice metod
- Definice přístupových metod pomocí „getterů“ a „setterů“
- Volání metod
- Definování typu návratové hodnoty
- Definice vstupních parametrů
- Přetěžování
- Využívání a dodržování principu zapouzdření

V daných konstrukcích je také důležité, aby je dokázali studenti definovat ve správném pořadí podle programátorských zvyklostí a aby některé konstrukce neprogramovali do jedné metody, ale vytvořili si různé metody, které daný problém řeší.

Studenti vyučování pomocí této koncepce výuky programování procvičovali příklady se strukturovanými konstrukcemi až po zvládnutí OO konstrukcí. Oproti druhé skupině, ve které studenti programovali kód do hlavní metody v programu, studenti vyučování podle koncepce výuky programování objects-first programovali kód do vytvořené třídy. Strukturované konstrukce procvičovali při OO návrhu programu s příklady jednorozměrného a vícerozměrného pole. Dále je v této podkapitole detailněji uvedena obsahová náplň koncepce výuky programování objects-first. Jednotlivá cvičení jsou očíslována postupně za sebou.

#### 1. Základy OOP v C#

Vysvětlení základních OO konstrukcí uvedených výše v úvodu podkapitoly, demonstrace OO konstrukcí na názorných příkladech.



## 2. Základní příkazy

Srovnání základních příkazů s příkazy uvedenými v předmětu ALGDS, pravidla a příkazy programovacího jazyku C#, příklady na práci s OO konstrukcemi.

## 3. Jednorozměrné pole

Práce s jednorozměrným polem pomocí OO návrhu programu s využitím OO konstrukcí: Vytvoření a naplnění pole, vypsání pole, počet hodnot s určitou vlastností, součet a průměr členů posloupnosti, nalezení minimální hodnoty a jejich počet, index prvního/posledního výskytu minima, odebrání prvku, vložení prvku, cyklický posun, zkopírování sudých členů z pole A do seznamu B, metoda-práce se seznamem.

## 4. Matice

Práce s dvourozměrným polem pomocí OO návrhu programu s využitím OO konstrukcí: Vytvoření a naplnění matice, výpis matice, součet a průměr členů matice, počet sudých členů, počet minimálních hodnot, průměr hodnot na R-tém řádku, záměna prvního a posledního řádku, součin prvků z S-tého sloupce, průměry ze všech sloupců.

## 5. Čtvercové matice

Práce s dvourozměrným polem s čtvercovou maticí pomocí OO návrhu programu s využitím OO konstrukcí: Vytvoření čtvercové matice pomocí konstruktora, naplnění matice, hlavní diagonála (nad, pod), vedlejší diagonála (nad, pod), trojúhelníky vymezené oběma diagonálami (nahore, dole, vpravo, vlevo).

### **Koncepce výuky programování objects-later**

Obsahová náplň koncepce výuky programování objects-later v pedagogickém experimentu se neliší od vymezení koncepce výuky, která byla uvedena v kapitole 3.

Studenti této skupiny vyučovaní podle koncepce výuky programování objects-later programovali nejdříve příklady se strukturovanými konstrukcemi. Oproti druhé skupině studenti psali kód v hlavní metodě programu. Strukturované konstrukce programovali s příklady jednorozměrného a vícerozměrného pole. U příkladů s polem využívali statických metod. Teprve po procvičení příkladů

s jednorozměrným a vícerozměrným polem byli studenti seznámeni s konstrukcemi OOP. Jednotlivá cvičení jsou očíslována postupně za sebou. Jednotlivá cvičení jsou očíslována také postupně za sebou.

#### 1. Základní příkazy

Vysvětlení základních příkazů (datové typy, sekvence, větvení, cyklus), srovnání základních příkazů s příkazy uvedenými v předmětu ALGDS, pravidla a příkazy programovacího jazyku C#, vše definováno v hlavní metodě main

#### 2. Jednorozměrné pole

Práce s jednorozměrným polem pomocí statických metod v hlavní části programu: Vytvoření a naplnění pole, vypsání pole, počet hodnot s určitou vlastností, součet a průměr členů posloupnosti, nalezení minimální hodnoty a jejich počet, index prvního/posledního výskytu minima, odebrání prvku, vložení prvku, cyklický posun, zkopírování sudých členů z pole A do seznamu B, metoda-práce se seznamem.

#### 3. Matice

Práce s dvourozměrným polem pomocí statických metod v hlavní části programu: Vytvoření a naplnění matice, výpis matice, součet a průměr členů matice, počet sudých členů, počet minimálních hodnot, průměr hodnot na R-tém řádku, záměna prvního a posledního řádku, součin prvků z S-tého sloupce, průměry ze všech sloupců.

#### 4. Čtvercová matice

Práce s dvourozměrným polem pomocí statických metod v hlavní části programu: Vytvoření čtvercové matice, naplnění matice, vypsání matice (metodou), hlavní diagonála (nad, pod), vedlejší diagonála (nad, pod), trojúhelníky vymezené oběma diagonálami (nahore, dole, vpravo, vlevo), základy OOP: třída, konstruktor, instance, datová položka.

#### 5. Základy OOP v C#

Procvičování pojmů probíraných v minulé jednotce: třída, konstruktor, instance, datová položka, přístupové metody pomocí „getterů“ a „setterů“, přetěžování, využívání a dodržování principu

zapouzdření, metody v OOP (metody, parametry již znají ze statických metod).

### **Společná probíraná témata obou koncepcí výuky programování**

Jak bylo uvedeno výše, po posttestu, který studenti absolvovali zhruba v polovině semestru, se v obou dvou skupinách studentů vyučovala stejná témata. Témata jsou uvedena v tabulce 7 a jsou zvýrazněna zelenou barvou. Níže je uvedena jejich detailnější obsahová náplň.

#### 6. Řazení

Ve výukové jednotce byly studentům ukázány principy základních řadících algoritmů: Bubblesort, Insertionsort, Selectionsort a Quicksort. Všechny řadící algoritmy byly demonstrovány pomocí OO konstrukcí na příkladech, které využívaly třídy, konstruktor, datové položky, metody, objektové typy aj. U řadících algoritmů bylo ukázáno řazení vzestupně i řazení sestupně.

#### 7. Soubory

Ve výukové jednotce byli studenti seznámeni s principem ukládání dat do souborů. V této výukové jednotce se jednalo především o práci s textovými soubory. V příkladech se jednalo o následující metody/algoritmy práce se souborem: Čtení souboru po řádcích, zápis do souboru, zjištění existence souboru, informace o souboru, kopie souboru, přesun souboru, smazání souboru, informace o adresářích, vypsaní souborů v adresáři, zpracování informací, které jsou uloženy v souboru, rozparsování souboru s příponou CSV a vyhledávání slov v souboru. Jednotlivé metody/algoritmy práce se soubory uvedenými výše byly demonstrovány pomocí OO konstrukcí (konstruktor, datové položky, přístupové metody, metody).

#### 8. Binární soubory, souborový systém

Ve výukové jednotce byli studenti seznámeni s principem zpracování dat v binárních souborech. V příkladech se jednalo o následující metody/algoritmy práce se souborem: zápis čísla do binárního souboru, přečtení čísla z binárního souboru, zápis textu do binárního souboru, přečtení textu z binárního souboru. Dále byly probírány

příklady na hledání souborů a hledání duplicit v souborech. Jednotlivé metody/algoritmy práce se soubory uvedenými výše byly opět demonstrovány pomocí OO konstrukcí.

#### 9. Parametry programu

Ve výukové jednotce byly probírány následující tři tematické oblasti:

- Parametry metod
- Parametry aplikace
- Vypiš soubor

V tematickém okruhu Parametry metod byly probírány metody s proměnným počtem parametrů, kde byly pomocí statické funkce vyhledávány znaky a kde byly použity konstrukce např. na zmenšení velikosti pole, cyklus foreach apod.

V tematickém okruhu Parametry aplikace byly zpracovávány argumenty zadané ve vývojovém prostředí, a kde byly jednotlivé argumenty skládány do jednoho argumentu, byl ověřován jejich počet, byla zobrazena nápověda pro zpracování parametrů a nakonec byly argumenty zpracovány.

V posledním tematickém okruhu Vypiš soubor byl vytvořen program, který načel vstupní parametry aplikace ve formě celé cesty k souboru a vypsal velikost souboru i jeho obsah. Pokud byla parametrem místo souboru složka, pak se vypsala velikost souboru i jeho obsah pro všechny soubory ve složce.

#### 5.1.4 Testy

V pedagogickém experimentu v předmětu Programování 1, jak je zřejmé z tabulky 5, byli studenti testováni pomocí následujících testů: posttest, retest 1 a retest 2. V předmětu Programování 1:

- posttest odpovídal zápočtovému testu
- retest 1 odpovídal opravnému zápočtovému testu
- retest 2 odpovídal zkouškovému testu

Studenti v předmětu mohli psát ještě 2. opravný zápočtový test. S ohledem na malý počet studentů nejsou výsledky tohoto testu uváděny. U zkouškového testu

měli studenti možnosti ještě dalších dvou oprav. Další opravné testy obsahovaly příklady nejen na práci se soubory, ale také na práci s jednorozměrným a vícerozměrným polem. S ohledem na malý počet studentů a podobnost příkladů z předchozích testů nejsou výsledky těchto testů uváděny.

Vzhledem k možnému ovlivnění výsledků studentů z testů způsobené výukou různým vyučujícím v jednotlivých skupinách, studenti na konci semestru vyplnili anonymní dotazníkové šetření s parametrickými otázkami, které obsahovalo otázky na zjištění:

- možného vlivu vyučujícího
- dostatečnosti materiálů ve výuce
- jiných materiálů, které studenti mohli využívat při studiu problematiky

Ukázka dotazníkového šetření je uvedena v příloze A.

Posttest studenti absolvovali zhruba v polovině semestru, retest 1 studenti absolvovali na konci semestru a retest 2 studenti absolvovali ve zkouškovém období. Podobná struktura byla u posttestu a retestu 1. Retest 2 obsahoval tematické okruhy společně probíraných tematických okruhů v obou skupinách studentů.

K úspěšnému splnění předmětu museli studenti získat z testů 60% možných bodů ze dvou částí. Testy se skládaly z části praktické (programátorské, programovací), která byla rozdělena na část objektových konstrukcí a část strukturovaných konstrukcí a části teoretické. V praktické části měli studenti za úkol naprogramovat zadané úlohy v programovacím jazyku C#. V části teoretické byli studenti testováni v univerzálním testovacím prostředí. Výsledný procentuální výsledek studentů byl vypočítán z váženého průměru jednotlivých částí testů:

$$(\text{Teoretická část} + (\text{Objektová část} * 2 + \text{Strukturovaná část} * 3) / 5) / 2$$

Vzorec pro výpočet procentuální úspěšnosti výsledků studentů byl stanoven domluvou mezi vyučujícími předmětu Programování 1.

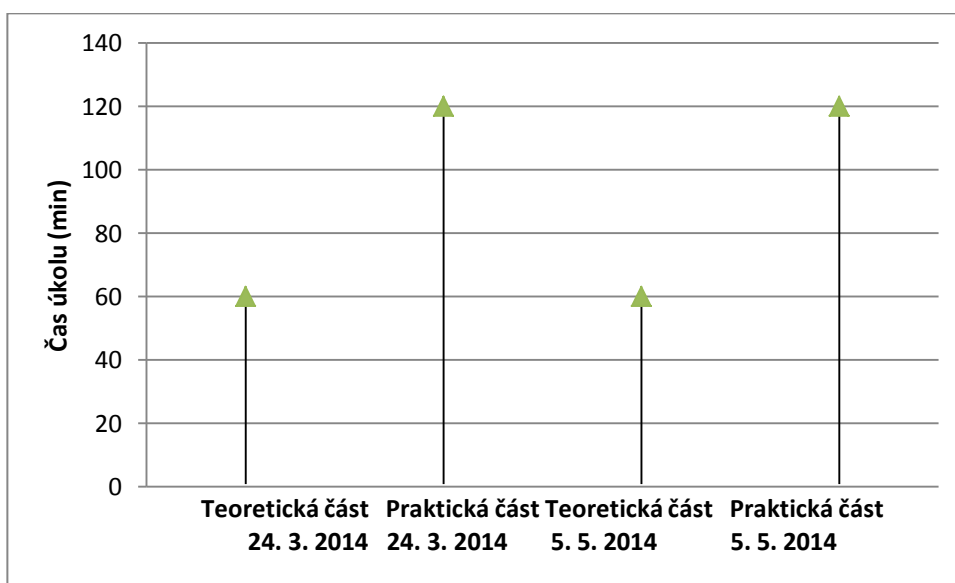
Autorem zadání testů jak teoretické, tak praktické části byl vyučující Ing. Petr Voborník Ph.D. Zadání praktické části testů bylo mezi vyučujícími konzultováno.

### **Posttest a retest 1**

Jak bylo uvedeno v předchozí části, testy posttest a retest 1 mají podobnou strukturu. Oba dva testy absolvovali všichni studenti. Retest 1 absolvovali i ti

studenti, kteří již uspěli v posttestu. Otestování úspěšných studentů retestem 1 bylo zvoleno s ohledem na možnou změnu programátorských schopností v obou skupinách. Posttest obsahuje úlohy na jednorozměrné pole. Retest 1 obsahuje úlohy na dvourozměrné pole. Oba dva testy vycházely z obsahové náplně předmětu, která byla probírána ve druhém až šestém týdnu semestru výuky předmětu Programování 1.

Teoretická část testu trvala 60 minut. Praktická část testu trvala 120 minut. Nejdříve studenti absolvovali teoretickou část. Po teoretické části studenti absolvovali praktickou část. Obě dvě části studenti psali v jednom dnu a v jednom bloku. Ukázka struktury testování studentů z akademického roku (pouze jedné skupiny studentů) 2013/2014 je ukázána na obrázku 1. Druhá skupina studentů byla testována analogicky, pouze v jiném dnu v daném týdnu. První dva grafy v levé části obrázku označují testování v posttestu (datum 24. 3. 2014), dva grafy v pravé části obrázku označují testování v retestu 1 (datum 5. 5. 2014). V akademickém roce 2014/2015 bylo testování studentů identické, pouze s jinými daty testů. V další části kapitoly jsou představeny jednotlivé části (praktická a teoretická část) testů.



Obrázek 1 – Ukázka testování studentů v akademickém roce 2013/2014 s časovým vymezením jednotlivých částí

### Praktická část

V praktické části měli studenti za úkol naprogramovat třídu s konstruktorem, atributy, vlastnostmi, metodami a měli správně vytvořit instanci třídy a volat metody

instance. Metody obsahovaly algoritmy strukturovaných konstrukcí. Zvláště byla hodnocena část objektových konstrukcí a část strukturovaných konstrukcí.

Část objektových konstrukcí se dá rozdělit na tři dílčí části:

1. Vytvoření konstruktora a inicializace datových položek
2. Vytvoření metod: správně zadané vstupní a výstupní parametry, návratové hodnoty
3. Vytvoření instance, volání metod

Část strukturovaných konstrukcí se dá rozdělit na dvě dílčí části:

1. Definice, naplnění a výpis posloupnosti
2. Strukturované konstrukce/algoritmy v metodách

Ukázka zadání posttestu je uvedena v příloze B. Zadání posttestu bylo publikováno v článcích Hubálovského a Kořínka (Hubálovský a Kořínek, 2015a; Hubálovský a Kořínek, 2015c). Ukázka zadání retestu 1 je uvedena v příloze C. Část teoretické části uvedené v následující podkapitole **Teoretická část** možností ověřování studentských programátorských znalostí i s ukázkami možného typu zadání úloh byla publikována v článku Hubálovský a Kořínek (2015b).

### **Teoretická část**

Teoretickou část testů lze rozdělit celkem do třech částí, kde z každé části bylo univerzálním testovacím prostředím náhodně na základě nastavení vyučujícího vybráno několik otázek. Teoretická část se celkem skládala z 10 otázek. Níže jsou podrobněji uvedeny jednotlivé její dílčí části.

1. V první části byly úlohy na doplňování kódu.
2. Ve druhé části Interaktivní prvky bylo několik typů úloh. Byly zde úlohy na následující části:
  - roztřídění položek do příslušné kategorie
  - vybrání jedné nebo několika odpovědí z většího výběru možností
  - přiřazení správných odpovědí k jednotlivým možnostem
3. Ve třetí části testu byly úlohy na doplnění hodnot do proměnných po vykonání příslušného kódu programu/algoritmu.

S první a třetí částí, s doplňováním částí kódu v úlohách a doplnění hodnot do proměnných, se studenti již setkali v zápočtovém a zkouškovém testu v předmětu

ALGDS. V další části této podkapitoly je ukázka jednoho možného zadání s jeho vysvětlením z každé části teoretické části. Obrázky příkladů jsou vytvořeny pomocí univerzálního testovacího prostředí. Autorem zadání příkladů je Ing. Petr Voborník, Ph.D.

### Doplňování kódu

Univerzálním testovacím prostředím jsou podle požadavků vyučujícího z této části vybrány studentům dvě otázky ze sady úloh, kde každá úloha má stejnou váhu. To znamená, že by měla být stejně náročná. Studenti mají za úkol doplnit kód příslušné metody. Studentům se vždy vygeneruje různé zadání na doplnění kódu. V sadě úloh jsou úlohy na následující tematické okruhy: čtvercové matice, matice a posloupnosti. Ukázka úlohy doplnění kódu v matici je na obrázku 2. Sada úloh je zaměřena hlavně na základní pojmy strukturovaného programování (sekvence, podmínka a proměnná) a základní konstrukce programovacího jazyku C#.

Zadání úlohy: Doplněte kód metody tak, aby ze vstupní matice pro každý řádek spočítala součet prvků se sudou hodnotou a tyto součty uložila do výstupní posloupnosti.

```
private int[] Funkce(int[,] vstup)
{
     posloupnost = new int[vstup.GetLength()];
    for(int i = 0; i < vstup.GetLength(0); i++)
    {
        int sum = ;
        for(int j = 0; j < vstup.GetLength(1); j++)
            if (vstup[]  2 == )
                sum += ;
         = sum;
    }
    return ;
}
```

Obrázek 2 – Úloha na doplňování kódu

### Interaktivní prvky

V této části jsou otázky, které využívají interaktivní prvky. Jsou zde otázky např. na přetahování položek odpovědí nebo označování kódu myší. Do každé



uvedené kategorie patří různá sada otázek. Každá z otázek má nastavenou jinou váhu. Celkové skóre studentů z této sady otázek bylo vypočítáno váhovým průměrem. Do testu bylo vybráno šest otázek ze sady otázek. Otázky byly zaměřeny na základní pojmy z OOP. Jsou v ní i otázky ze strukturovaného programování: větvení a hodnoty booleovských výrazů. Ukázka otázky na přetahování položek odpovědí je na obrázku 3.

Zadání úlohy: Zařaďte výrazy pod hodnotu, která je jejich výsledkem.

true	false
$!(9 == 7)$	$6 * 2 == 13 - 2$
$2 * 3 == 4 + 2$	
	$50 + 5 == 65$
	$10 == 10 \ \&\& \ 6 \ != \ 6$
$42 == 21 * 2$	
$100 - 25 \ != \ 50$	$0 == 8 \    \ 4 \ != \ 4$

Obrázek 3 – Úloha na využití interaktivních prvků

### Doplnění hodnot do proměnných

Univerzálním testovacím prostředím podle požadavků vyučujícího jsou z této části vybrány studentům náhodně dvě otázky. Studenti mají za úkol na základě vstupních hodnot doplnit hodnoty výstupní po provedení příslušné metody. Na výběr byly úlohy na cyklické posuny a jiné posuny. Každý typ otázky z této sady měl opět stejnou váhu i stejnou pravděpodobnost, že bude vybrán. Ukázka úlohy na doplnění hodnot do proměnných je na obrázku 4.

Zadání úlohy: Na základě vstupních hodnot posloupnosti A určete, jaké v ní budou hodnoty výstupní, po proběhnutí následující metody.

```

public void Operace(int[] a)
{
    int e = a[a.Length / 2];
    for (int i = 1; i <= a.Length-2; i++)
        if (a[i] <= e)
            a[i] = a[i-1] + a[i];
}

```

Prvek pole	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
Vstupní hodnoty	2	2	2	4	4	6
Výstupní hodnoty	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Obrázek 4 – Doplnění hodnot do proměnných

## Retest 2

Retest 2 mohou psát pouze ti studenti, kteří uspěli buď v posttestu nebo v retestu 1, popřípadě ještě v jednom opravném testu, jak již bylo uvedeno výše v kapitole. Retest 2 se opět skládal z části teoretické a části praktické. Zkouškové okruhy vycházejí z obsahové náplně předmětu Programování 1, která byla vyučována po posttestu zhruba po polovině semestru. Obsahová náplň předmětu již byla shodná pro obě dvě skupiny studentů. Zadání praktické části retestu 2 se liší od předchozích testů (posttest, retest 1).

## Praktická část

Praktická část obsahuje čtyři příklady na práci se soubory. Studenti si mohli libovolně vybrat tři příklady a ty naprogramovat.

V praktické části retestu 2, stejně jak bylo uvedeno u posttestu a retestu 1, měli studenti za úkol naprogramovat třídu s metodami a správně vytvořit instanci třídy a volat metody instance (i když není uvedeno, měli by používat i konstruktor, atributy, vlastnosti). Metody algoritmy/strukturované konstrukce jsou zaměřeny na konstrukce využívající úlohy se soubory. Zvlášť je hodnocena část objektových konstrukcí a část strukturovaných konstrukcí.

Část objektových konstrukcí se dá rozdělit na dílčí dvě části:

1. Vytvoření metod: správně zadané vstupní a výstupní parametry, vrácení parametrů
2. Vytvoření instance, volání metod

Část algoritmických/strukturovaných konstrukcí se skládá ze strukturovaných konstrukcí/algoritmů v metodách.

Hodnocení příkladů je stejné, jak bylo uvedeno v kapitole Posttest a retest 1 a jak bylo uvedeno výše. Ukázka zadání retestu 2 je v příloze D.

### **Teoretická část**

Teoretická část retestu 2 testu obsahovala 10 otázek v univerzálním testovacím prostředí. Test lze rozdělit do následujících částí:

1. Opakovací úlohy z postestu a retestu 1
  - a. Doplnění kódu
  - b. Interaktivní prvky (OOP)
  - c. Doplnění hodnot proměnných
2. Řazení
  - a. Doplnění kódu
  - b. Teoretická východiska
3. Soubory
  - a. Doplnění kódu
  - b. Základní pojmy

Teoretická část retestu 2 navíc obsahovala oproti posttestu a retestu 1 tematické celky 2. Řazení a 3. Soubory. Vzhledem ke stanoveným hypotézám uvedených v kapitole 2 nejsou výsledky těchto tematických celků zahrnuty do vlastního výzkumu v disertační práci.

### **Opakovací úlohy z postestu a retestu 1**

Z této kategorie byly vybrány celkem čtyři otázky. Kategorie otázek byly stejné, jak bylo uvedeno v podkapitole Posttest a retest 1. Jedna otázka byla vybrána na doplnění kódu, dvě otázky byly vybrány na interaktivní prvky a jedna otázka byla vybrána na doplnění hodnot proměnných.

## 6 VÝSLEDKY VLASTNÍHO VÝZKUMU

V této kapitole budou představeny výsledky vlastního výzkumu pedagogického experimentu. Částečné výsledky výzkumu různých částí testů (dílčí strukturovaná část, dílčí teoretická část a dílčí objektová část) byly prezentovány v článcích Hubálovského a Koříneka (Hubálovský a Kořínek, 2015a; Hubálovský a Kořínek, 2015b; Hubálovský a Kořínek, 2015c).

Experimentální skupina studentů byla vyučována pomocí koncepce výuky programování objects-first, kontrolní skupina studentů byla vyučována pomocí koncepce výuky programování objects-later.

Kapitola je rozdělena na několik částí. V úvodních částech kapitoly jsou prezentovány výsledky výzkumu studentů z posttestu, retestu 1 a retestu 2 z akademických let 2013/2014 a 2014/2015. Tyto úvodní části mají podobnou strukturu a jsou rozděleny podle akademických let. V každé úvodní části (úvodní část posttestu, dále úvodní část retestu 1 a nakonec úvodní část retestu 2) je nejdříve uvedeno rozdělení studentů do skupin (experimentální a kontrolní skupina) podle jejich výsledků z pretestu. Po rozdělení studentů do skupin následují v každé části výsledky studentů hlavní části v daném testu (v posttestu nebo retestu 1 nebo retestu 2, dále bude souhrnně označeno jako testu). Hlavní část testu je rozdělena na část celkových výsledků, část praktickou (programátorskou) a část teoretickou. Praktická část testu je ještě dále rozdělena na část objektových konstrukcí a část strukturovaných konstrukcí. Vybrané hlavní části testu (část teoretická, část objektových konstrukcí a část strukturovaných konstrukcí) se dají rozdělit na dílčí části testu.

Teoretická část testu je rozdělena na tři dílčí části (u posttestu a retestu 1) nebo na sedm částí<sup>17</sup>(retest 2).

Objektová část testu je rozdělena na tři dílčí části (u posttestu nebo retestu 1) nebo na dvě dílčí části (retest 2).

Strukturovaná část testu je rozdělena na dvě dílčí části (u posttestu nebo retestu 1). U retestu 2 není strukturovaná část rozdělena na dílčí části.

---

<sup>17</sup> Ve výsledcích vlastního výzkumu jsou uvedeny pouze ty části retestu 2, které byly společné jak pro posttest a retest 1, tak i pro retest 2.

Výsledky jednotlivých částí testu studentů v daných skupinách uvedených výše jsou uvedeny v příslušných tabulkách v daných částech kapitoly.

Úvodní tabulka testu vždy obsahuje výsledky hlavních částí testu (celková část výsledků, praktická část výsledků, teoretická část výsledků, objektová část výsledků a strukturovaná část výsledků). Další tabulky testu obsahují podrobnější výsledky teoretické části, objektové části a strukturované části. Sledovanými údaji, v jednotlivých tabulkách testu, jsou:

- Statistické zpracování výsledků neparametrickým testem, uvedena hodnota  $p$
- Průměrné skóre výsledků studentů příslušných částí testu dosažených v jednotlivých skupinách (experimentální a kontrolní)
- Počet úspěšných studentů v jednotlivých skupinách (úspěšnost je stanovena na 60% dosažených bodů z daných částí)
- Počet studentů, kteří z daných částí dosáhli 0% bodů (neúspěšných studentů)

Statistické zpracování bylo provedeno pomocí neparametrického U-testu Manna a Whitneyho. Zároveň bylo provedeno statistické zpracování také pomocí Studentova t-testu u všech částí. Vizualní normalita byla zjišťována pomocí grafu typu Q-Q. Graf typu Q-Q v jednotlivých částech testů je uveden pouze v případě, pokud se lišily výsledky neparametrického a parametrického testu. Stejně jako výsledky Studentova t-testu jsou uvedeny v tabulkách pouze v případě, pokud se lišily výsledky neparametrického a parametrického testu.

Počet úspěšných a neúspěšných studentů je uveden pouze pro názornost, protože stanovená úspěšnost studentů v testech je dána vzorcem, který byl uveden v předcházející kapitole. Výsledky neúspěšných studentů mohou ovlivnit celkové výsledky studentů v jednotlivých skupinách.

U každé části testů byl zjišťován Spearmanův koeficient pořadové korelace vzhledem k výsledkům studentů z pretestu. Spearmanův koeficient pořadové korelace je zjišťován s ohledem jak na všechny studenty v předmětu Programování 1, tak i na studenty v jednotlivých skupinách. Výsledky Spearmanova koeficientu jsou uvedeny v příloze E.

Na konci každé části je uvedena interpretace dosažených výsledků studentů z jednotlivých testů a z jejich částí, které byly uvedeny v daných tabulkách každé části, popřípadě v tabulkách jiných částí nebo v tabulkách přílohy G u studentů, které vyučoval stejný vyučující.

Na konci kapitol jsou uvedeny výsledky dotazníkového šetření, které je přiloženo v příloze F. Je uvedena jeho analýza, která zjišťuje možné ovlivnění výsledků studentů ve skupinách vyučovaných různými vyučujícími.

Pokud je v tabulkách u jednotlivých částí testu mezi skupinami studentů v jednotlivých částech statisticky významný rozdíl (nulová hypotéza odmítnuta), je hodnota  $p$  v testu vyznačena červeně.

V jednotlivých částech kapitol jsou také uvedeny výsledky studentů z posttestu, kteří absolvovali i retest 1. Výsledky pouze těch studentů, kteří absolvovali jak posttest tak i retest 1, je důležité uvést vzhledem k pozdějšímu srovnání výsledků studentů v posttestu i v retestu 1 v jednotlivých skupinách.

V poslední části kapitoly jsou srovnány výsledky studentů experimentální a kontrolní skupiny z různých akademických let, u kterých předmět vyučoval pouze stejný vyučující. Srovnání výsledků studentů vyučovaných stejným vyučujícím by mělo dokreslit možné rozdíly mezi experimentální a kontrolní skupinou studentů. Struktura této části je stejná jako u předchozích částí podkapitol.

## 6.1 Výsledky studentů z akademického roku 2013/2014

### 6.1.1 Výsledky studentů z posttestu

V akademickém roce 2013/2014 předmět Programování 1, respektive posttest, absolvovalo celkem 28 studentů. V experimentální i v kontrolní skupině byl stejný počet studentů. Rozdělení studentů do skupin je uvedeno v tabulce 8.

Tabulka 8 – Rozdělení studentů do dvou skupin podle výsledku z pretestu

Známka z pretestu	Experimentální skupina studentů	Kontrolní skupina studentů
Výborně	5	3
Velmi dobře	1	2
Dobře	5	5
Neuspěl	3	4
<b>Celkem studentů</b>	14	14

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 9 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 10 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 11 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 12 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

V tabulce 103 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí posttestu. V tabulce 104 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu. V tabulce 105 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu.

Tabulka 9 – Výsledky studentů z hlavních částí posttestu

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	0,077	
	Průměr	46,31%	29,8%
	Úspěšní studenti	5	1
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,023	
	Průměr	40,56%	11,85%
	Úspěšní studenti	5	1
	Neúspěšní studenti	3	8
<b>Teoretická část</b>	Hodnota p	0,566	
	Průměr	52,06%	47,75%
	Úspěšní studenti	5	5
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,023	
	Průměr	44,18%	10,95%
	Úspěšní studenti	5	1
	Neúspěšní studenti	4	8
<b>Strukturovaná část</b>	Hodnota p	0,046	
	Průměr	38,15%	12,44%
	Úspěšní studenti	4	1
	Neúspěšní studenti	4	9

Tabulka 10 – Výsledky studentů z dílčích částí teoretické části posttestu

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,206	
	Průměr	36,61%	26,32%
	Úspěšní studenti	2	3
	Neúspěšní studenti	0	4
<b>Interaktivní prvky</b>	Hodnota p	0,872	
	Průměr	68,96%	66,93%



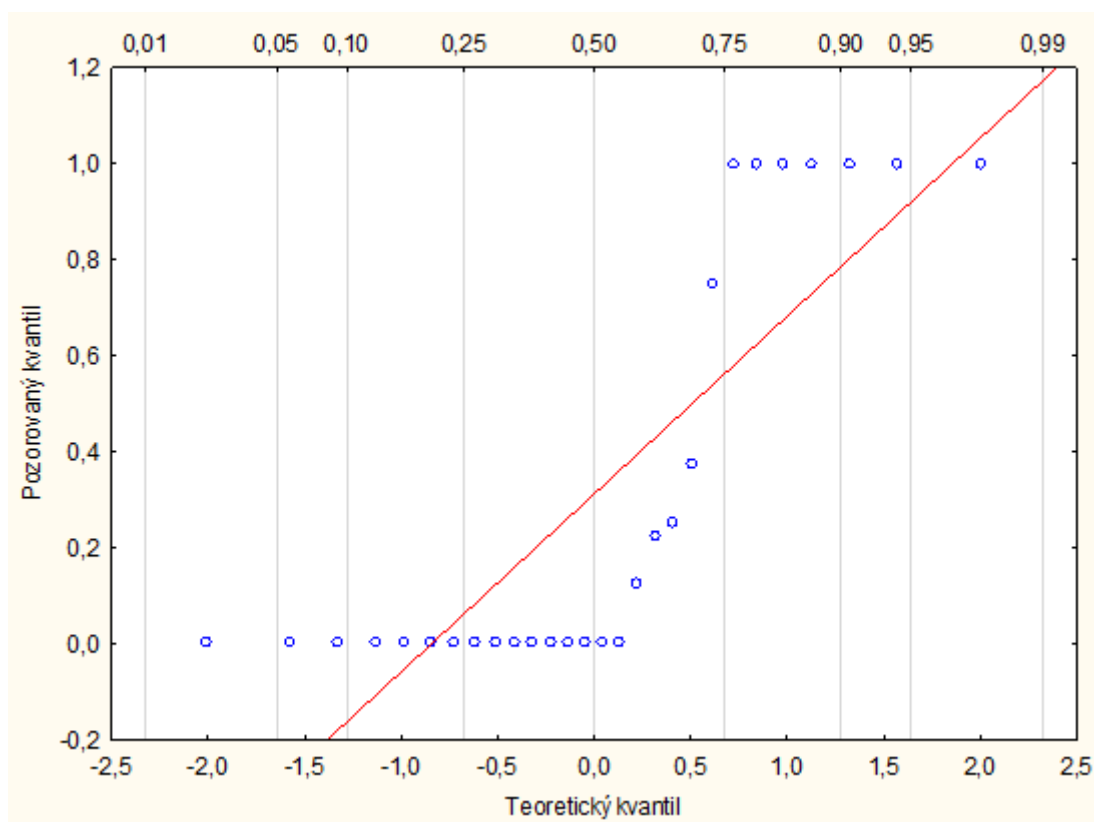
	Úspěšní studenti	9	11
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,730	
	Průměr	36,90%	36,31%
	Úspěšní studenti	2	2
	Neúspěšní studenti	3	3

Tabulka 11 – Výsledky studentů z dílčích částí objektové části posttestu

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,023	
	Průměr	55,36%	16,25%
	Úspěšní studenti	7	1
	Neúspěšní studenti	4	9
<b>Metody a vlastnosti</b>	Hodnota p	0,033	
	Průměr	35,97%	8,87%
	Úspěšní studenti	5	1
	Neúspěšní studenti	5	11
<b>Instance a volání metod</b>	Hodnota p	0,077	
	Hodnota p (Studentův t-test)	0,013	
	Graf normality	Obrázek 5	
	Normální rozdělení	Ne	
	Průměr	50,89%	11,43%
	Úspěšní studenti	7	1
	Neúspěšní studenti	6	10

Tabulka 12 – Výsledky studentů z dílčích částí strukturované části posttestu

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
Naplnění a výpis posloupnosti	Hodnota p	0,041	
	Průměr	55,36	19,82%
	Úspěšní studenti	9	3
	Neúspěšní studenti	4	9
Návrh algoritmů metod	Hodnota p	0,094	
	Průměr	29,55%	8,75%
	Úspěšní studenti	3	0
	Neúspěšní studenti	6	11



Obrázek 5 – Graf normality objektové části (vytvoření instance a volání metod) studentů z posttestu z akademického roku 2013/2014, obrázek byl vytvořen v programu Statistica 12

## Interpretace výsledků

Mezi hodnotou skóre výsledků studentů experimentální skupiny a kontrolní skupiny je v hlavních částí posttestu statisticky významný rozdíl v praktické, objektové a strukturované části. Výrazně vyšší hodnoty skóre výsledků dosáhli studenti experimentální skupiny i v celkových výsledcích. Pouze v teoretické části testu byly hodnoty skóre výsledků obou skupin studentů podobné.

Vyšší hodnoty skóre výsledků v teoretické části posttestu i ve všech jejích dílčích částech dosáhli opět studenti experimentální skupiny, kde výrazně vyšší hodnoty skóre výsledků získali v dílčí části Doplnování kódu.

V objektové části byl mezi hodnotou skóre výsledků studentů v daných skupinách statisticky významný rozdíl v dílčích částech Definice konstrukturu a datových položek a Metody a vlastnosti. U dílčí části instance a volání metod se hodnota skóre výsledků studentů ke statisticky významnému rozdílu blížila.

Ve strukturované části byl mezi hodnotou skóre výsledků studentů v daných skupinách statisticky významný rozdíl v dílčí části Naplnění a výpis posloupnosti. I v dílčí části strukturované části Návrh algoritmů metod však dosáhli studenti experimentální skupiny výrazně vyšší hodnoty skóre výsledků.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování I probírali dříve než studenti druhé skupiny.

Studenti experimentální skupiny i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí posttestu studentů) u všech studentů značná až vysoká negativní závislost, tj. nižší známka z pretestu odpovídá vyšší hodnotě skóre v jednotlivých částech posttestu. Na této značně vysoké až vysoké negativní závislosti mají vliv výsledky studentů hlavně experimentální skupiny, ve které je spíše nízká závislost pouze v dílčí teoretické části posttestu Doplnování kódu<sup>18</sup>.

---

<sup>18</sup> Spearmanův koeficient dílčí části posttestu, v tomto případě dílčí části teoretické části Doplnování kódu je stanoven mezi srovnávanými jevy známkou z pretestu studentů a jejich výsledku výše uvedené dílčí teoretické části posttestu. I když není mezi srovnávanými jevy v textu uveden jev známka z pretestu, vždy je myšlen mezi srovnávanými jevy výsledek z pretestu studentů a příslušná dílčí část testu. Tato formulace platí i pro další kapitoly pro výpočet Spearmanova koeficientu.

### 6.1.2 Výsledky studentů z posttestu, kteří absolvovali i retest 1

V akademickém roce 2013/2014 předmět Programování 1, respektive posttest a zároveň retest 1, absolvovalo celkem 19 studentů. V experimentální skupině studentů oba testy, tj. posttest a retest 1, absolvovalo celkem 10 studentů. V kontrolní skupině studentů oba testy, tj. posttest a retest 1, absolvovalo celkem 9 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 13.

**Tabulka 13 – Rozdělení studentů, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	4	2
Velmi dobře	0	1
Dobře	4	3
Neuspěl	2	3
<b>Celkem studentů</b>	10	9

Oproti posttestu retest 1 v experimentální skupině studentů neabsolvovali čtyři studenti (v každé ze skupin podle známky z pretestu po jednom studentovi).

Oproti posttestu retest 1 v kontrolní skupině studentů neabsolvovalo pět studentů (v každé ze skupin podle známky z pretestu po jednom studentovi, kromě dvou studentů, kteří dostali z pretestu známku dobře).

Výsledky všech studentů z posttestu v obou skupinách i výsledky z posttestu pouze těch studentů, kteří absolvovali i retest 1, by měly být podobné s ohledem na složení obou skupin studentů.

Na základě výpočtu váženého průměru opět o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 14 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 15 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 16 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 17 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

V tabulce 106 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí posttestu. V tabulce 107 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu. V tabulce 108 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu.

**Tabulka 14 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z hlavních částí**

<b>Část posttestu</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Celkové výsledky</b>	Hodnota p	0,111	
	Průměr	50,02%	30,84%
	Úspěšní studenti	4	1
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,022	
	Průměr	45,88%	12,87%
	Úspěšní studenti	4	1
	Neúspěšní studenti	1	5
<b>Teoretická část</b>	Hodnota p	0,713	
	Průměr	54,16%	49,22%
	Úspěšní studenti	4	4
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,037	
	Průměr	49,33%	11,85%
	Úspěšní studenti	4	1
	Neúspěšní studenti	2	5
<b>Strukturovaná část</b>	Hodnota p	0,045	
	Průměr	43,58%	12,87%
	Úspěšní studenti	4	1
	Neúspěšní studenti	2	6

Tabulka 15 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z dílčích částí teoretické části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,540	
	Průměr	37,59%	32,13%
	Úspěšní studenti	2	3
	Neúspěšní studenti	0	2
<b>Interaktivní prvky</b>	Hodnota p	0,838	
	Průměr	70,23%	66,62%
	Úspěšní studenti	6	6
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,327	
	Průměr	41,67%	36,11%
	Úspěšní studenti	1	1
	Neúspěšní studenti	1	2

Tabulka 16 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z dílčích částí objektové části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,050	
	Průměr	59,75%	19,72%
	Úspěšní studenti	5	1
	Neúspěšní studenti	2	6
<b>Metody a vlastnosti</b>	Hodnota p	0,033	
	Průměr	38,75%	6,94%
	Úspěšní studenti	4	1
	Neúspěšní studenti	3	8
<b>Instance a volání metod</b>	Hodnota p	0,153	

	Průměr	60%	15,28%
	Úspěšní studenti	6	1
	Neúspěšní studenti	4	6

**Tabulka 17 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z dílčích částí strukturované části**

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,045	
	Průměr	61,25%	19,17%
	Úspěšní studenti	7	2
	Neúspěšní studenti	2	6
<b>Návrh algoritmů metod</b>	Hodnota p	0,131	
	Průměr	34,75%	9,72%
	Úspěšní studenti	3	0
	Neúspěšní studenti	4	7

### **Interpretace výsledků**

Hodnoty skóre výsledků studentů v posttestu, kteří absolvovali i retest 1, v hlavních částech byla vyšší v obou skupinách studentů než hodnoty skóre výsledků studentů, kteří absolvovali pouze posttest. Studenti kontrolní skupiny, kteří absolvovali i retest 1, dosáhli nižší hodnoty skóre výsledků z posttestu než studenti, kteří absolvovali pouze posttest, v dílčích částech teoretické části Interaktivní prvky a Doplnění hodnot proměnných, v dílčí části objektové části Metody a vlastnosti a v dílčí části strukturované části Naplnění a výpis posloupnosti. Studenti experimentální skupiny, kteří absolvovali i retest 1, dosáhli ve všech dílčích částech posttestu vyšší hodnoty skóre výsledků než studenti, kteří absolvovali pouze posttest.

V hlavních částech posttestu i v jeho jednotlivých dílčích částech dosáhli studenti experimentální skupiny vyšší hodnoty skóre výsledků než studenti kontrolní skupiny.

Statisticky významné rozdíly mezi kontrolní a experimentální skupinou studentů jsou stejné v jednotlivých částech posttestu (praktická část, objektová část i její dílčí části Definice konstruktoru a datových položek a Metody a vlastnosti, strukturovaná část i její dílčí část Naplnění a výpis posloupnosti) jako u všech studentů, kteří absolvovali pouze posttest (viz předchozí kapitola).

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Studenti experimentální skupiny dosáhli ještě vysoké hodnoty skóre výsledků v dílčích částech objektové části Definice konstruktoru a datových položek a Instance a volání metod a v dílčí části strukturované části Naplnění a výpis posloupnosti.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí posttestu studentů) u všech studentů značná negativní závislost, tj. nižší známka z pretestu odpovídá vyšší hodnotě skóre jednotlivých částí posttestu. Na této značně vysoké negativní závislosti mezi srovnávanými jevy mají vliv výsledky studentů hlavně experimentální skupiny studentů. V této skupině je nízká závislost mezi jevy pouze v teoretických dílčích částech a v dílčí objektové části Definice konstruktoru a datových položek posttestu. U studentů kontrolní skupiny je mezi jevy negativní závislost střední až nízká, kromě výsledků v dílčí teoretické části a v dílčí objektové části Definice konstruktoru a datových položek posttestu a strukturované části i jejích dílčích částí.

### **6.1.3 Výsledky studentů z retestu 1**

V akademickém roce 2013/2014 předmět Programování 1, respektive retest 1, absolvovalo celkem 19 studentů. V experimentální skupině retest 1 absolvovalo celkem 10 studentů. V kontrolní skupině studentů retest 1 absolvovalo celkem 9 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 18.



**Tabulka 18 – Rozdělení studentů do dvou skupin podle výsledku z pretestu.**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	4	2
Velmi dobře	0	1
Dobře	4	3
Neuspěl	2	3
<b>Celkem studentů</b>	10	9

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 19 je uvedeno zpracování výsledků studentů z hlavních částí retestu 1. V tabulce 20 je uvedeno zpracování výsledků studentů z teoretické části retestu 1. V tabulce 21 je uvedeno zpracování výsledků studentů z objektové části retestu 1. V tabulce 22 je uvedeno zpracování výsledků studentů ze strukturované části retestu 1.

V tabulce 109 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí retestu 1. V tabulce 110 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí retestu 1. V tabulce 111 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí retestu 1.

**Tabulka 19 – Výsledky studentů z retestu 1 z hlavních částí**

<b>Část retestu 1</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Celkové výsledky</b>	Hodnota p	0,270	
	Průměr	49,95%	37,66%
	Úspěšní studenti	4	2
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,540	
	Průměr	39,14%	25,87%
	Úspěšní studenti	3	1

	Neúspěšní studenti	3	4
<b>Teoretická část</b>	Hodnota p	0,270	
	Průměr	60,77%	49,46%
	Úspěšní studenti	5	2
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,683	
	Průměr	38,10%	33,01%
	Úspěšní studenti	4	2
	Neúspěšní studenti	3	4
<b>Strukturovaná část</b>	Hodnota p	0,270	
	Průměr	39,83%	21,11%
	Úspěšní studenti	3	1
	Neúspěšní studenti	3	4

Tabulka 20 – Výsledky studentů z retestu 1 z dílčích částí teoretické části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,307	
	Průměr	51,77%	37,04%
	Úspěšní studenti	4	1
	Neúspěšní studenti	0	1
<b>Interaktivní prvky</b>	Hodnota p	0,838	
	Průměr	73,29%	70,20%
	Úspěšní studenti	8	5
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,142	
	Průměr	47,50%	24,07%
	Úspěšní studenti	4	1
	Neúspěšní studenti	1	4

Tabulka 21 – Výsledky studentů z retestu 1 z dílčích částí objektové části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,934	
	Průměr	42,50%	49,72%
	Úspěšní studenti	5	4
	Neúspěšní studenti	4	5
<b>Metody a vlastnosti</b>	Hodnota p	0,870	
	Průměr	33,26%	31,75%
	Úspěšní studenti	3	2
	Neúspěšní studenti	4	4
<b>Instance a volání metod</b>	Hodnota p	0,540	
	Průměr	45,00%	27,08%
	Úspěšní studenti	5	2
	Neúspěšní studenti	5	6

Tabulka 22 – Výsledky studentů z retestu 1 z dílčích částí strukturované části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,540	
	Průměr	46,75%	35,56%
	Úspěšní studenti	4	4
	Neúspěšní studenti	3	4
<b>Návrh algoritmů metod</b>	Hodnota p	0,236	
	Průměr	36,38%	13,89%
	Úspěšní studenti	2	1
	Neúspěšní studenti	4	5

## **Interpretace výsledků**

I když mezi hodnotami skóre výsledků studentů v daných skupinách není statisticky významný rozdíl, ve všech částech retestu 1 kromě dílčí objektové části Definice konstruktoru a datových položek dosáhli vyšší hodnoty skóre výsledků (výrazněji vyšší hodnoty skóre výsledků) studenti experimentální skupiny.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti experimentální skupiny i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí retestu 1 studentů) u všech studentů značná až vysoká negativní závislost kromě dílčí teoretické části Interaktivní prvky, dílčí objektové části Definice konstruktoru a datových položek a dílčí strukturované části Naplnění a výpis posloupnosti, tj. nižší známka z pretestu studentů odpovídá vyšší hodnotě skóre studentů jednotlivých částí retestu 1.

Na nízké negativní závislosti výsledků mezi jevy mají vliv výsledky studentů hlavně experimentální skupiny, u kterých je tato nízká závislost dosažena v praktických částech retestu 1 i v jejich dílčích částech a v dílčí části teoretické části Interaktivní prvky. U kontrolní skupiny studentů je nízká negativní závislost mezi srovnávanými jevy u dílčí teoretické části Doplnění kódu, dílčí objektové části Definice konstruktoru a datových položek a dílčí strukturované části Naplnění a výpis posloupnosti.

### **6.1.4 Výsledky studentů z retestu 2**

V akademickém roce 2013/2014 předmět Programování 1, respektive retest 2, absolvovalo celkem pouze 8 studentů. V experimentální skupině retest 2 absolvovalo celkem 6 studentů. V kontrolní skupině studentů retest 2 absolvovali celkem 2 studenti. Rozdělení studentů do skupin je uvedeno v tabulce 23.

**Tabulka 23 – Rozdělení studentů do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	5	2
Velmi dobře	0	0
Dobře	0	0
Neuspěl	1	0
<b>Celkem studentů</b>	<b>6</b>	<b>2</b>

Vzhledem k nízkým výsledkům studentů v testech posttest a retest 1, je počet studentů v obou skupinách velice malý. Z tohoto důvodu nebyla použita žádná statistická metoda pro testování nulových hypotéz. V tabulkách níže jsou uvedeny pouze průměrné hodnoty výše skóre studentů v jednotlivých skupinách z jednotlivých částí a počty úspěšných/neúspěšných studentů. Jsou uvedeny pouze výsledky těch částí retestu 2, které jsou shodné s částmi posttestu a retestu 1.

V tabulce 24 je uvedeno zpracování výsledků studentů z hlavních částí retestu 2. V tabulce 25 je uvedeno zpracování výsledků studentů z teoretické části retestu 2. V tabulce 26 je uvedeno zpracování výsledků studentů z objektové části retestu 2.

**Tabulka 24 – Výsledky studentů z retestu 2 z hlavních částí**

<b>Část retestu 2</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Celkové výsledky</b>	Průměr	55,92%	60,93%
	Úspěšní studenti	3	1
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Průměr	41,25%	46,50%
	Úspěšní studenti	1	1
	Neúspěšní studenti	1	0
<b>Teoretická část</b>	Průměr	70,55%	75,37%
	Úspěšní studenti	5	2
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Průměr	51,67%	60,00%

	Úspěšní studenti	3	1
	Neúspěšní studenti	1	0
<b>Strukturovaná část</b>	Průměr	34,38%	37,50%
	Úspěšní studenti	1	1
	Neúspěšní studenti	2	0

Tabulka 25 – Výsledky studentů z retestu 2 z dílčích částí teoretické části

Část retestu 2	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Průměr	57,16%	45,45%
	Úspěšní studenti	2	1
	Neúspěšní studenti	0	0
<b>Interaktivní prvky (OOP)</b>	Průměr	78,58%	87,26%
	Úspěšní studenti	8	2
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Průměr	86,11%	91,67%
	Úspěšní studenti	5	2
	Neúspěšní studenti	0	0

Tabulka 26 – Výsledky studentů z retestu 2 z dílčích částí objektové části

Část retestu 2	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Metody a vlastnosti</b>	Průměr	44,44%	50,00%
	Úspěšní studenti	2	1
	Neúspěšní studenti	1	0
<b>Instance a volání metod</b>	Průměr	62,50%	75,00%
	Úspěšní studenti	4	2
	Neúspěšní studenti	1	0

### 6.1.5 Srovnání výsledků studentů z posttestu a z retestu 1

V této podkapitole budou zvlášť srovnány výsledky studentů experimentální a kontrolní skupiny z posttestu a z retestu 1. Cílem srovnání je sledování změn hodnoty skóre výsledků studentů v obou skupinách. Ke zjištění, jestli je mezi experimentální nebo kontrolní skupinou studentů mezi hodnotami skóre výsledků z posttestu a z retestu 1 statisticky významný rozdíl, byl použit Wilcoxonův neparametrický test.

#### Experimentální skupina studentů

V akademickém roce 2013/2014 předmět Programování 1, respektive posttest a retest 1, absolvovalo v experimentální skupině celkem 10 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 27.

Tabulka 27 – Rozdělení studentů do dvou skupin podle výsledku z pretestu.

Známka z pretestu	Experimentální skupina studentů
Výborně	4
Velmi dobře	0
Dobře	4
Neuspěl	2
<b>Celkem studentů</b>	10

V tabulce 28 je uvedeno zpracování výsledků studentů experimentální skupiny z hlavních částí posttestu a retestu 1. V tabulce 29 je uvedeno zpracování výsledků studentů experimentální skupiny z teoretické části posttestu a retestu 1. V tabulce 30 je uvedeno zpracování výsledků studentů experimentální skupiny z objektové části posttestu a retestu 1. V tabulce 31 je uvedeno zpracování výsledků studentů experimentální skupiny ze strukturované části posttestu a retestu 1.

V tabulce 112 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu a retestu 1.

Tabulka 28 – Výsledky studentů experimentální skupiny z hlavních částí posttestu a retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Celkové výsledky</b>	Hodnota p	0,959	
	Průměr	50,02%	49,95%
	Úspěšní studenti	4	4
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,646	
	Průměr	45,88%	39,14%
	Úspěšní studenti	4	3
	Neúspěšní studenti	1	3
<b>Teoretická část</b>	Hodnota p	0,093	
	Průměr	54,16%	60,77%
	Úspěšní studenti	4	5
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,241	
	Průměr	49,33%	38,10%
	Úspěšní studenti	4	4
	Neúspěšní studenti	2	3
<b>Strukturovaná část</b>	Hodnota p	0,859	
	Průměr	43,58%	39,83%
	Úspěšní studenti	4	3
	Neúspěšní studenti	2	3

Tabulka 29 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z teoretické části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Doplňování kódu</b>	Hodnota p	0,139	
	Průměr	37,59%	51,77%
	Úspěšní studenti	2	4
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,374	
	Průměr	70,23%	73,29%
	Úspěšní studenti	6	8
	Neúspěšní studenti	0	0



<b>Doplnění hodnot proměnných</b>	Hodnota p	0,262	
	Průměr	41,67%	47,50%
	Úspěšní studenti	1	4
	Neúspěšní studenti	1	1

Tabulka 30 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z objektové části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,237	
	Průměr	59,75%	42,50%
	Úspěšní studenti	5	4
	Neúspěšní studenti	2	5
<b>Metody a vlastností</b>	Hodnota p	0,612	
	Průměr	38,75%	33,26%
	Úspěšní studenti	4	3
	Neúspěšní studenti	3	4
<b>Instance a volání metod</b>	Hodnota p	0,109	
	Průměr	60,00%	45,00%
	Úspěšní studenti	6	5
	Neúspěšní studenti	4	5

Tabulka 31 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 ze strukturované části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Naplnění a výpis posloupností</b>	Hodnota p	0,310	
	Průměr	61,25%	46,75%
	Úspěšní studenti	7	4
	Neúspěšní studenti	2	3

<b>Návrh algoritmů metod</b>	Hodnota p	0,932	
	Průměr	34,75%	36,38%
	Úspěšní studenti	3	2
	Neúspěšní studenti	4	4

### Interpretace výsledků

V celkových výsledcích dosáhli studenti experimentální skupiny podobné hodnoty skóre výsledků jak z posttestu tak i z retestu 1. V posttestu studenti dosáhli vyšší hodnoty skóre výsledků ve všech částech kromě teoretické části i všech jejích dílčích částí a dílčí strukturované části Návrh algoritmů metod než v retestu 1.

Příčin těchto výsledků může být více. Studenti mohli podcenit přípravu na retest 1. Při programování společných témat pro obě dvě skupiny studentů po posttestu se v předmětu Programování 1 studenti zabývali jinými typy úloh, než jaké byly v posttestu. Studenti, kteří již uspěli v posttestu, nemuseli retestu 1 přikládat takovou pozornost.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí posttestu a retestu 1 studentů) u všech studentů značná až vysoká negativní závislost u všech částí kromě dílčí teoretické části Interaktivní prvky, dílčích částí objektové části Definice konstruktoru a datových položek a Instance a volání metod.

### Kontrolní skupina studentů

V akademickém roce 2013/2014 předmět Programování 1, respektive posttest a retest 1, absolvovalo v kontrolní skupině celkem 9 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 32.

Tabulka 32 – Rozdělení studentů do dvou skupin podle výsledku z pretestu.

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>
Výborně	2
Velmi dobře	1
Dobře	3
Neuspěl	3
<b>Celkem studentů</b>	<b>9</b>

V tabulce 33 je uvedeno zpracování výsledků studentů kontrolní skupiny z hlavních částí posttestu a retestu 1. V tabulce 34 je uvedeno zpracování výsledků studentů kontrolní skupiny z teoretické části posttestu a retestu 1. V tabulce 35 je uvedeno zpracování výsledků studentů kontrolní skupiny z objektové části posttestu a retestu 1. V tabulce 36 je uvedeno zpracování výsledků studentů kontrolní skupiny ze strukturované části posttestu a retestu 1.

V tabulce 113 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu a retestu 1.

**Tabulka 33 – Výsledky studentů kontrolní skupiny z hlavních částí posttestu a retestu 1**

<b>Část testů</b>	<b>Sledované údaje</b>	<b>Posttest</b>	<b>Retest 1</b>
<b>Celkové výsledky</b>	Hodnota p	0,173	
	Průměr	30,84%	37,66%
	Úspěšní studenti	1	2
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,225	
	Průměr	12,87%	25,87%
	Úspěšní studenti	1	1
	Neúspěšní studenti	5	4
<b>Teoretická část</b>	Hodnota p	0,767	
	Průměr	49,22%	49,46%
	Úspěšní studenti	4	2
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,080	
	Průměr	11,85%	33,01%
	Úspěšní studenti	1	2
	Neúspěšní studenti	5	4
<b>Strukturovaná část</b>	Hodnota p	0,500	
	Průměr	12,87%	21,11%
	Úspěšní studenti	1	1
	Neúspěšní studenti	6	4

Tabulka 34 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z teoretické části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Doplňování kódu</b>	Hodnota p	0,401	
	Průměr	32,13%	37,04%
	Úspěšní studenti	3	1
	Neúspěšní studenti	2	1
<b>Interaktivní prvky</b>	Hodnota p	0,594	
	Průměr	66,62%	70,20%
	Úspěšní studenti	6	5
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,128	
	Průměr	36,11%	24,07%
	Úspěšní studenti	1	1
	Neúspěšní studenti	2	4

Tabulka 35 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z objektové části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,068	
	Průměr	19,72%	49,72%
	Úspěšní studenti	1	5
	Neúspěšní studenti	6	4
<b>Metody a vlastnosti</b>	Hodnota p	0,080	
	Průměr	6,94%	31,75%
	Úspěšní studenti	1	2
	Neúspěšní studenti	8	4
<b>Instance a volání metod</b>	Hodnota p	0,180	
	Průměr	15,28%	27,08%

	Úspěšní studenti	1	2
	Neúspěšní studenti	6	6

**Tabulka 36 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 ze strukturované části**

Část testů	Sledované údaje	Posttest	Retest 1
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,273	
	Průměr	19,17%	35,56%
	Úspěšní studenti	2	4
	Neúspěšní studenti	6	4
<b>Návrh algoritmů metod</b>	Hodnota p	0,787	
	Průměr	9,72%	13,89%
	Úspěšní studenti	0	1
	Neúspěšní studenti	7	5

### **Interpretace výsledků**

V celkových výsledcích dosáhli studenti kontrolní skupiny v retestu 1 vyšší hodnoty skóre výsledků jednotlivých částí kromě dílčí teoretické části Doplnění hodnot proměnných než v posttestu. Především v objektové části a jejích dílčích částech byla hodnota skóre výsledků studentů v retestu 1 výrazně vyšší než v posttestu.

Příčin těchto výsledků může být více. Studenti se mohli více připravit na retest 1 oproti posttestu. Při programování společných témat pro obě skupiny studentů po posttestu se v předmětu Programování 1 zabývali sice jinými typy úloh, než jaké byly v posttestu, ale další procvičování strukturovaných a objektových konstrukcí mělo na studenty kontrolní skupiny vliv. Více se ztotožnili s danými programovými konstrukcemi a nejspíše z tohoto důvodu byly jejich výsledky v retestu 1 vyšší než v posttestu.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí posttestu a retestu 1 studentů) u studentů kontrolní skupiny značná negativní

závislost ve všech částech kromě teoretické části a dílčí části strukturované části Naplnění a výpis posloupnosti, tj. nižší známka z pretestu odpovídá vyšší hodnotě skóre jednotlivých částí posttestu a retestu 1.

### **6.1.6 Výsledky dotazníkového šetření**

V příloze F v tabulce 139 je uvedeno vyhodnocení dotazníkového šetření studentů experimentální a kontrolní skupiny studentů. Podle výsledků dotazníkového šetření u studentů, kteří anonymní dotazník vyplnili, není mezi vyučujícími, materiály nebo přípravou studentů na hodiny předmětu Programování 1 statisticky významný rozdíl. V dotazníkovém šetření zjišťovaly otázky číslo 4 a číslo 11 stejné hledisko. Tyto otázky byly zformulovány trochu odlišným způsobem pro zjištění, jestli studenti vyplňují dotazníkové šetření zodpovědně a jestli nevyplňují otázky náhodně. Korelace výsledků těchto dvou otázek by tudíž měla být vysoká. Korelační koeficient Spearmanovy korelace je u všech studentů 0,871, tj. mezi odpověďmi dvou daných otázek všech studentů je velmi vysoká závislost. V experimentální skupině studentů je koeficient korelace 0,929, v kontrolní skupině studentů je koeficient korelace 0,734. Mezi odpověďmi dvou výše uvedených otázek studentů experimentální skupiny je velmi vysoká závislost. U studentů kontrolní skupiny je mezi odpověďmi dvou výše uvedených otázek vysoká závislost. V obou skupinách je předpoklad, že studenti vyplňovali dotazníkové šetření zodpovědně a ne náhodně.

## **6.2 Výsledky studentů z akademického roku 2014/2015**

### **6.2.1 Výsledky studentů z posttestu**

V akademickém roce 2014/2015 předmět Programování 1, respektive posttest, absolvovalo celkem 34 studentů. V experimentální skupině v první polovině semestru předmět Programování 1 absolvovalo 16 studentů. V kontrolní skupině v první polovině semestru předmět Programování 1 absolvovalo 18 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 37.

**Tabulka 37 – Rozdělení studentů do dvou skupin podle výsledku z pretestu v akademickém roce 2014/2015**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	10	7
Velmi dobře	1	3
Dobře	3	5
Neuspěl	2	3
<b>Celkem studentů</b>	16	18

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 38 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 39 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 40 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 41 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

V tabulce 114 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí posttestu. V tabulce 115 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu. V tabulce 116 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu.

**Tabulka 38 – Výsledky studentů z posttestu z hlavních částí v akademickém roce 2014/2015**

<b>Část posttestu</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Celkové výsledky</b>	Hodnota p	0,417	
	Průměr	41,06%	47,66%
	Úspěšní studenti	4	7
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,569	
	Průměr	29,61%	32,80%

	Úspěšní studenti	4	5
	Neúspěšní studenti	6	3
<b>Teoretická část</b>	Hodnota p	0,184	
	Průměr	52,51%	62,52%
	Úspěšní studenti	7	11
	Neúspěšní studenti	0	0
<b>Objektivá část</b>	Hodnota p	0,931	
	Průměr	35,83%	29,07%
	Úspěšní studenti	5	4
	Neúspěšní studenti	6	5
<b>Strukturovaná část</b>	Hodnota p	0,227	
	Průměr	25,47%	35,28%
	Úspěšní studenti	7	2
	Neúspěšní studenti	8	4

Tabulka 39 – Výsledky studentů z posttestu z dílčích částí teoretické části v akademickém roce 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,352	
	Průměr	38,68%	48,26%
	Úspěšní studenti	2	6
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,109	
	Průměr	68,12%	79,52%
	Úspěšní studenti	10	15
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,605	
	Průměr	39,06%	47,69%
	Úspěšní studenti	5	8
	Neúspěšní studenti	3	2



Tabulka 40 – Výsledky studentů z posttestu z dílčích částí objektové části v akademickém roce 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,692	
	Průměr	29,22%	30,42%
	Úspěšní studenti	5	5
	Neúspěšní studenti	10	9
<b>Metody a vlastnosti</b>	Hodnota p	0,654	
	Průměr	28,85%	31,34%
	Úspěšní studenti	5	5
	Neúspěšní studenti	8	6
<b>Instance a volání metod</b>	Hodnota p	0,129	
	Průměr	49,61%	25,00%
	Úspěšní studenti	9	5
	Neúspěšní studenti	6	12

Tabulka 41 – Výsledky studentů z posttestu z dílčích částí strukturované části v akademickém roce 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,479	
	Průměr	33,28%	40,28%
	Úspěšní studenti	5	8
	Neúspěšní studenti	8	5
<b>Návrh algoritmů metod</b>	Hodnota p	0,352	
	Průměr	21,56%	32,78%
	Úspěšní studenti	2	4
	Neúspěšní studenti	8	7

## **Interpretace výsledků**

Mezi výsledky experimentální a kontrolní skupiny studentů z retestu 1 není mezi skupinami statisticky významný rozdíl. V hlavních částech posttestu kromě objektové části dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny. Ve všech jednotlivých dílčích částech hlavních částí dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny kromě dílčí objektové části Instance a volání metod.

Na základě dosažených výsledků studentů je těžké stanovit, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny, i když se určité spojitosti dají vysledovat. V objektové části dosáhli celkově vyšší hodnoty skóre studenti experimentální skupiny, kteří objektová témata probírali na začátku předmětu Programování 1, i když se jedná pouze o jednu dílčí část objektové části. V ostatních dílčích částech byly výsledky daných skupin podobné, i když o něco vyšší hodnoty skóre dosáhli studenti kontrolní skupiny. Naopak, ve strukturované části i všech jejích dílčích částí dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny, kteří daná témata probírali na začátku předmětu Programování 1.

Studenti experimentální skupiny i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí posttestu studentů) u všech studentů značná až vysoká negativní závislost kromě dílčí teoretické části Doplnění kódu, tj. nižší známka z pretestu odpovídá vyšší hodnotě skóre výsledků jednotlivých částí posttestu studentů. Na této značně vysoké až vysoké negativní závislosti mají vliv výsledky studentů hlavně kontrolní skupiny, kde mezi srovnávanými jevy je značná až vysoká negativní závislost.

### **6.2.2 Výsledky studentů z posttestu, kteří absolvovali i retest 1**

V akademickém roce 2014/2015 předmět Programování 1, respektive posttest a zároveň retest 1, absolvovalo celkem 28 studentů. V experimentální skupině studentů oba testy, tj. posttest a retest 1, absolvovalo celkem 11 studentů. V kontrolní

skupině studentů oba testy, tj. posttest a retest 1, absolvovalo celkem 17 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 42.

**Tabulka 42 – Rozdělení studentů do dvou skupin podle výsledku z pretestu, kteří absolvovali i retest 1**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	6	7
Velmi dobře	1	3
Dobře	3	5
Neuspěl	1	2
<b>Celkem studentů</b>	11	17

Oproti posttestu se v experimentální skupině studentů retestu 1 nezúčastnilo pět studentů (podle výsledků z pretestu po čtyřech studentech se známkou výborně a jednom studentovi, který v pretestu neuspěl). Studenti, kteří z pretestu dosáhli známky výborně, se odmítli zúčastnit retestu 1 z různých důvodů.

Oproti posttestu se v kontrolní skupině studentů retestu 1 nezúčastnil pouze jeden student, který v pretestu neuspěl.

Výsledné hodnoty skóre výsledků studentů z posttestu (kteří absolvovali i retest 1) by mohly být vyšší (výrazněji vyšší) v kontrolní skupině (s ohledem na složení obou skupin studentů) než v experimentální skupině studentů oproti výsledkům studentů, kteří absolvovali v obou skupinách pouze posttest.

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 43 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 44 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 45 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 46 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

V tabulce 117 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí posttestu. V tabulce 118 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny

z výsledku pretestu a jednotlivých částí posttestu. V tabulce 119 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu.

Tabulka 43 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z hlavních částí

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	<b>0,038</b>	
	Průměr	30,75%	49,61%
	Úspěšní studenti	0	7
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	<b>0,048</b>	
	Průměr	14,77%	34,73%
	Úspěšní studenti	0	5
	Neúspěšní studenti	6	2
<b>Teoretická část</b>	Hodnota p	<b>0,021</b>	
	Průměr	46,72%	64,50%
	Úspěšní studenti	3	11
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,240	
	Průměr	19,77%	30,78%
	Úspěšní studenti	1	4
	Neúspěšní studenti	6	4
<b>Strukturovaná část</b>	Hodnota p	<b>0,024</b>	
	Průměr	11,44%	37,35%
	Úspěšní studenti	0	7
	Neúspěšní studenti	7	3

Tabulka 44 – Výsledky studentů posttestu, kteří absolvovali i retest 1, z dílčích částí teoretické části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,145	

	Průměr	36,65%	50,82%
	Úspěšní studenti	1	6
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,008	
	Průměr	61,35%	80,76%
	Úspěšní studenti	6	15
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,196	
	Průměr	31,06%	50,49%
	Úspěšní studenti	2	8
	Neúspěšní studenti	2	1

Tabulka 45 – Výsledky studentů posttestu, kteří absolvovali i retest 1, z dílčích částí objektové části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,126	
	Průměr	13,18%	32,21%
	Úspěšní studenti	2	5
	Neúspěšní studenti	9	8
<b>Metody a vlastností</b>	Hodnota p	0,110	
	Průměr	12,80%	33,19%
	Úspěšní studenti	1	5
	Neúspěšní studenti	7	5
<b>Instance a volání metod</b>	Hodnota p	0,851	
	Průměr	33,52%	26,47%
	Úspěšní studenti	5	5
	Neúspěšní studenti	6	11

**Tabulka 46 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí strukturované části**

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,046	
	Průměr	15,45%	42,65%
	Úspěšní studenti	1	8
	Neúspěšní studenti	7	4
<b>Návrh algoritmů metod</b>	Hodnota p	0,056	
	Průměr	9,43%	34,71%
	Úspěšní studenti	0	4
	Neúspěšní studenti	7	6

### Interpretace výsledků

Studenti v kontrolní skupině, kteří absolvovali retest 1, dosáhli vyšší hodnoty skóre výsledků ve všech částech posttestu než studenti kontrolní skupiny se studentem, který absolvoval pouze posttest.

Oproti tomu studenti v experimentální skupině, kteří absolvovali retest 1, dosáhli nižší hodnoty skóre ve všech částech posttestu než studenti experimentální skupiny se studenty, kteří absolvovali pouze posttest.

Mezi výsledky kontrolní a experimentální skupiny studentů v posttestu, kteří absolvovali i retest 1, oproti výsledkům všech studentů, kteří absolvovali pouze posttest, je statisticky významný rozdíl v následujících částech posttestu: celkové výsledky, praktická část, teoretická část i její dílčí část Interaktivní prvky a strukturovaná část i její dílčí část Naplnění a výpis posloupnosti.

Ve všech částech posttestu dosáhli vyšší hodnoty skóre studenti kontrolní skupiny kromě dílčí objektové části Instance a volání metod než studenti experimentální skupiny.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti experimentální skupiny i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Studenti kontrolní skupiny dosáhli vysoké hodnoty skóre výsledků i v dalších dílčích částech teoretické části.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí posttestu studentů) u všech studentů značná negativní závislost ve všech částech posttestu kromě dílčí teoretické části Doplnování kódu, tj. nižší známka z pretestu odpovídá vyšší hodnotě skóre jednotlivých částí posttestu. Na této značně vysoké až vysoké negativní závislosti mají vliv výsledky studentů hlavně kontrolní skupiny, ve které je nízká závislost pouze v dílčí části teoretické části posttestu Doplnování kódu.

### 6.2.3 Výsledky studentů z retestu 1

V akademickém roce 2014/2015 předmět Programování 1, respektive retest 1, absolvovalo celkem 28 studentů. V experimentální skupině studentů retest 1 absolvovalo celkem 11 studentů. V kontrolní skupině studentů retest 1 absolvovalo celkem 17 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 47.

**Tabulka 47 – Rozdělení studentů do dvou skupin podle výsledku z pretestu, kteří absolvovali retest 1**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	6	7
Velmi dobře	1	3
Dobře	3	5
Neuspěl	1	2
<b>Celkem studentů</b>	11	17

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 48 je uvedeno zpracování výsledků studentů z hlavních částí retestu 1. V tabulce 49 je uvedeno zpracování výsledků studentů z teoretické části retestu 1.

V tabulce 50 je uvedeno zpracování výsledků studentů z objektové části retestu 1. V tabulce 51 je uvedeno zpracování výsledků studentů ze strukturované části retestu 1.

V tabulce 120 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí retestu 1. V tabulce 121 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí retestu 1. V tabulce 122 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí retestu 1.

**Tabulka 48 – Výsledky studentů z retestu 1 z hlavních částí**

<b>Část retestu 1</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Celkové výsledky</b>	Hodnota p	0,541	
	Průměr	49,09%	55,92%
	Úspěšní studenti	4	7
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,832	
	Průměr	41,49%	45,11%
	Úspěšní studenti	4	5
	Neúspěšní studenti	1	3
<b>Teoretická část</b>	Hodnota p	0,132	
	Průměr	56,70%	66,74%
	Úspěšní studenti	4	12
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,925	
	Průměr	47,03%	46,09%
	Úspěšní studenti	5	6
	Neúspěšní studenti	1	3
<b>Strukturovaná část</b>	Hodnota p	0,672	
	Průměr	37,80%	44,46%
	Úspěšní studenti	3	5
	Neúspěšní studenti	2	3



Tabulka 49 – Výsledky studentů z retestu 1 z dílčích částí teoretické části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,017	
	Průměr	34,48%	55,55%
	Úspěšní studenti	1	6
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,221	
	Průměr	72,93%	80,37%
	Úspěšní studenti	8	14
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,944	
	Průměr	56,06%	54,41%
	Úspěšní studenti	5	9
	Neúspěšní studenti	0	1

Tabulka 50 – Výsledky studentů z retestu 1 z dílčích částí objektové části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,944	
	Průměr	42,73%	44,41%
	Úspěšní studenti	4	5
	Neúspěšní studenti	3	4
<b>Metody a vlastnosti</b>	Hodnota p	0,906	
	Průměr	42,01%	43,49%
	Úspěšní studenti	4	7
	Neúspěšní studenti	2	5
<b>Instance a volání metod</b>	Hodnota p	0,707	

	Průměr	57,95%	51,47%
	Úspěšní studenti	7	8
	Neúspěšní studenti	3	5

**Tabulka 51 – Výsledky studentů z retestu 1 z dílčích částí strukturované části**

<b>Část retestu 1</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,832	
	Průměr	54,09%	59,71%
	Úspěšní studenti	7	12
	Neúspěšní studenti	2	3
<b>Návrh algoritmů metod</b>	Hodnota p	0,672	
	Průměr	29,66%	36,84%
	Úspěšní studenti	1	4
	Neúspěšní studenti	4	5

### **Interpretace výsledků**

V hlavních částech retestu 1 není mezi výsledky studentů experimentální a kontrolní skupiny statisticky významný rozdíl. Kromě objektové části dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků v hlavních částech retestu 1 než studenti experimentální skupiny.

V teoretické části je mezi výsledky studentů experimentální a kontrolní skupiny statisticky významný rozdíl v dílčí části Doplnění kódu. V dílčí části teoretické části Doplnění hodnot proměnných dosáhli vyšší hodnoty skóre výsledků studenti experimentální skupiny než studenti kontrolní skupiny. V ostatních dílčích částech dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny.

V objektové části dosáhli studenti experimentální skupiny vyšší hodnoty skóre výsledků v dílčí části Instance a volání metod. V jejích dalších částech dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny. Ve strukturované části dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny.

Na základě dosažených výsledků studentů je těžké stanovit, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny, i když se určité spojitosti dají vysledovat, stejně jako u výsledků studentů z posttestu. V objektové části dosáhli celkově vyšší hodnoty skóre výsledků studenti experimentální skupiny, kteří objektová témata probírali na začátku předmětu Programování 1, i když se jedná pouze o jednu dílčí část objektové části. V ostatních dílčích částech byly výsledky daných skupin studentů podobné, i když o něco vyšší hodnoty skóre výsledků dosáhli studenti kontrolní skupiny. Naopak, ve strukturované části i všech jejích dílčích částech dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny, kteří daná témata probírali na začátku předmětu Programování 1.

Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Další vysoké hodnoty skóre výsledků dosáhli studenti obou skupin v dílčí části objektové části Doplnění hodnot proměnných a v dílčí části strukturované části Naplnění a výpis posloupnosti. Studenti kontrolní skupiny dosáhli vysoké hodnoty skóre výsledků i v další dílčí části teoretické části Doplnění kódu.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí pretestu 1) střední až nízká negativní závislost. U dílčí teoretické části Interaktivní prvky je mezi sledovanými jevy velmi slabá závislost.

U experimentální skupiny studentů je podle výpočtu Spearmanova koeficientu pořadové korelace mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí retestu 1 studentů) střední až nízká negativní závislost kromě teoretické části a její dílčí části Doplnění hodnot proměnných, kde je mezi sledovanými jevy vysoká závislost. U praktické, jak objektové i její dílčí části Metody a vlastnosti, tak strukturované i její dílčí části Návrh algoritmů metod je mezi sledovanými jevy velmi slabá závislost.

U kontrolní skupiny výpočet Spearmanova koeficientu pořadové korelace nepotvrdil střední až nízkou negativní závislost mezi jevy (jako u všech studentů). U hlavních částí retestu 1 (kromě strukturované části) je mezi sledovanými jevy vysoká závislost.

#### 6.2.4 Výsledky studentů z retestu 2

V akademickém roce 2014/2015 předmět Programování 1, respektive retest 2, absolvovalo celkem 24 studentů. V kontrolní skupině studentů retest 2 absolvovalo celkem 11 studentů. V experimentální skupině studentů retest 2 absolvovalo celkem 13 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 52.

Tabulka 52 – Rozdělení studentů do dvou skupin podle výsledku z pretestu, kteří psali retest 2

Známka z pretestu	Experimentální skupina studentů	Kontrolní skupina studentů
Výborně	6	8
Velmi dobře	3	0
Dobře	4	2
Neuspěl	0	1
<b>Celkem studentů</b>	13	11

Na základě výpočtu váženého průměru lepších výsledků z pretestu dosáhli studenti kontrolní skupiny.

V tabulce 53 je uvedeno zpracování výsledků studentů z hlavních částí retestu 2. V tabulce 54 je uvedeno zpracování výsledků studentů z teoretické části retestu 2. V tabulce 55 je uvedeno zpracování výsledků studentů z objektové části retestu 2.

V tabulce 123 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí retestu 2. V tabulce 124 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí retestu 2. V tabulce 125 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí retestu 2.

Tabulka 53 – Výsledky studentů z retestu 2 z hlavních částí

Část retestu 2	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	0,562	
	Průměr	43,18%	50,38%
	Úspěšní studenti	2	3
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,817	
	Průměr	24,66%	31,83%
	Úspěšní studenti	1	1
	Neúspěšní studenti	1	4
<b>Teoretická část</b>	Hodnota p	0,202	
	Průměr	61,70%	68,93%
	Úspěšní studenti	6	11
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,695	
	Průměr	35,64%	45,19%
	Úspěšní studenti	2	6
	Neúspěšní studenti	2	4
<b>Strukturovaná část</b>	Hodnota p	0,908	
	Průměr	17,35%	22,92%
	Úspěšní studenti	1	1
	Neúspěšní studenti	1	4

Tabulka 54 – Výsledky studentů z retestu 2 z dílčích částí teoretické části

Část retestu 2	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,156	
	Průměr	41,69%	57,70%
	Úspěšní studenti	4	6
	Neúspěšní studenti	0	0
<b>Interaktivní prvky (OOP)</b>	Hodnota p	0,885	

	Průměr	86,84%	86,44%
	Úspěšní studenti	11	11
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,354	
	Průměr	65,15%	85,90%
	Úspěšní studenti	7	11
	Neúspěšní studenti	2	0

Tabulka 55 – Výsledky studentů z retestu 2 z dílčích částí objektové části

Část retestu 2	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Metody a vlastnosti</b>	Hodnota p	0,602	
	Průměr	31,44%	41,03%
	Úspěšní studenti	2	4
	Neúspěšní studenti	2	4
<b>Instance a volání metod</b>	Hodnota p	0,791	
	Průměr	41,93%	51,44%
	Úspěšní studenti	3	6
	Neúspěšní studenti	2	4

### Interpretace výsledků

V hlavních částech retestu 2 není mezi výsledky studentů experimentální a kontrolní skupiny statisticky významný rozdíl. Kromě dílčí teoretické části Interaktivní prvky (OOP) dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků ve všech částech retestu 2.

Rozdíly výsledků mezi skupinami studentů mohou být způsobeny jejich rozdělením podle výsledků studentů z pretestu, kde studenti kontrolní skupiny dosáhli lepších výsledků.

Na základě stanovených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti experimentální skupiny i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v jejích dílčích částech Interaktivní prvky a Doplnění hodnot proměnných. Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků v dílčí části objektové části Instance a volání metod.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí retestu 2 daných skupin studentů) střední až nízká negativní závislost kromě strukturované části a dílčí teoretické části Interaktivní prvky (OOP), kde je mezi jevy velmi slabá negativní závislost.

U experimentální skupiny studentů výpočet Spearmanova koeficientu pořadové korelace potvrdil střední až nízkou negativní závislost mezi jevy kromě praktické, objektové a teoretické části, kde je mezi sledovanými jevy velmi slabá závislost.

U kontrolní skupiny výpočet Spearmanova koeficientu pořadové korelace potvrdil střední až nízkou negativní závislost mezi jevy. U dílčí části teoretické části Doplnění hodnot proměnných je mezi srovnávanými jevy vysoká závislost.

### **6.2.5 Srovnání výsledků studentů z posttestu a retestu 1**

V této podkapitole budou zvlášť srovnány výsledky studentů experimentální a kontrolní skupiny z posttestu a retestu 1. Zajímavé bude sledovat, jak se po výuce společných témat změnily výsledky studentů v obou skupinách. Ke zjištění, jestli je mezi jednotlivými skupinami studentů mezi výsledky posttestu a retestu 1 statisticky významný rozdíl, byl opět použit Wilcoxonův neparametrický test.

#### **Experimentální skupina studentů**

V akademickém roce 2014/2015 posttest a retest 1 v předmětu Programování 1 v experimentální skupině studentů absolvovalo celkem 11 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 56.

**Tabulka 56 – Rozdělení studentů, kteří absolvovali posttest i retest 1, do dvou skupin podle výsledku z pretestu,**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>
Výborně	6
Velmi dobře	1
Dobře	3
Neuspěl	1
<b>Celkem studentů</b>	<b>11</b>

V tabulce 57 je uvedeno zpracování výsledků studentů experimentální skupiny z hlavních částí posttestu a retestu 1. V tabulce 58 je uvedeno zpracování výsledků studentů experimentální skupiny z teoretické části posttestu a retestu 1. V tabulce 59 je uvedeno zpracování výsledků studentů experimentální skupiny z objektové části posttestu a retestu 1. V tabulce 60 je uvedeno zpracování výsledků studentů experimentální skupiny ze strukturované části posttestu a retestu 1.

V tabulce 126 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu a retestu 1.

**Tabulka 57 – Výsledky studentů experimentální skupiny z hlavních částí posttestu a retestu 1**

<b>Část testů</b>	<b>Sledované údaje</b>	<b>Posttest</b>	<b>Retest 1</b>
<b>Celkové výsledky</b>	Hodnota p	<b>0,003</b>	
	Průměr	30,75%	49,09%
	Úspěšní studenti	0	4
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	<b>0,022</b>	
	Průměr	14,77%	41,49%
	Úspěšní studenti	0	4
	Neúspěšní studenti	6	1
<b>Teoretická část</b>	Hodnota p	0,091	
	Průměr	46,72%	56,70%
	Úspěšní studenti	3	4
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	<b>0,047</b>	



	Průměr	19,77%	47,03%
	Úspěšní studenti	1	5
	Neúspěšní studenti	6	1
<b>Strukturovaná část</b>	Hodnota p	<b>0,024</b>	
	Průměr	11,44%	37,80%
	Úspěšní studenti	0	3
	Neúspěšní studenti	7	2

Tabulka 58 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z teoretické části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Doplňování kódu</b>	Hodnota p	0,657	
	Průměr	36,65%	34,48%
	Úspěšní studenti	1	1
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,155	
	Průměr	61,35%	72,93%
	Úspěšní studenti	8	6
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	<b>0,021</b>	
	Průměr	31,06%	56,06%
	Úspěšní studenti	2	5
	Neúspěšní studenti	2	0

Tabulka 59 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z objektové části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,069	
	Průměr	13,18%	42,73%
	Úspěšní studenti	2	4
	Neúspěšní studenti	9	3
<b>Metody a vlastnosti</b>	Hodnota p	0,028	
	Průměr	12,80%	42,01%
	Úspěšní studenti	1	4
	Neúspěšní studenti	7	2
<b>Instance a volání metod</b>	Hodnota p	0,091	
	Průměr	33,52%	57,95%
	Úspěšní studenti	5	7
	Neúspěšní studenti	6	3

Tabulka 60 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 ze strukturované části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Naplnění a výpis posloupností</b>	Hodnota p	0,018	
	Průměr	15,45%	54,09%
	Úspěšní studenti	1	7
	Neúspěšní studenti	7	2
<b>Návrh algoritmů metod</b>	Hodnota p	0,042	
	Průměr	9,43%	29,66%
	Úspěšní studenti	0	1
	Neúspěšní studenti	7	4

## Interpretace výsledků

Studenti experimentální skupiny dosáhli oproti výsledkům z posttestu (studenti, kteří absolvovali i retest 1) v retestu 1 výrazně vyšší hodnoty skóre výsledků ve všech hlavních částech. V dílčích částech retestu 1 dosáhli studenti experimentální skupiny výrazně vyšší hodnoty skóre výsledků ve všech dílčích částech kromě dílčí teoretické části Doplnování kódu než v posttestu.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí posttestu a retestu 1) nízká až střední negativní závislost kromě objektové části i její dílčí části Instance a volání metod a dílčí části teoretické části Interaktivní prvky, kde je mezi srovnávanými jevy velmi slabá závislost.

## Kontrolní skupina studentů

V akademickém roce 2014/2015 posttest a retest 1 v předmětu Programování 1 v kontrolní skupině studentů absolvovalo celkem 17 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 61.

**Tabulka 61 – Rozdělení studentů, kteří absolvovali posttest i retest 1, do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Kontrolní skupina studentů</b>
Výborně	7
Velmi dobře	3
Dobře	5
Neuspěl	2
<b>Celkem studentů</b>	<b>17</b>

V tabulce 62 je uvedeno statistické zpracování výsledků studentů kontrolní skupiny z hlavních částí posttestu a retestu 1. V tabulce 63 je uvedeno zpracování výsledků studentů kontrolní skupiny z teoretické části posttestu a retestu 1. V tabulce 64 je uvedeno zpracování výsledků studentů kontrolní skupiny z objektové části posttestu a retestu 1. V tabulce 65 je uvedeno zpracování výsledků studentů kontrolní skupiny ze strukturované části posttestu a retestu 1.

V tabulce 127 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledků pretestu a jednotlivých částí posttestu a retestu 1.

Tabulka 62 – Výsledky studentů kontrolní skupiny z hlavních částí posttestu a retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Celkové výsledky</b>	Hodnota p	0,113	
	Průměr	49,61%	55,92%
	Úspěšní studenti	7	7
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,088	
	Průměr	34,73%	45,11%
	Úspěšní studenti	5	5
	Neúspěšní studenti	2	3
<b>Teoretická část</b>	Hodnota p	0,463	
	Průměr	64,50%	66,74%
	Úspěšní studenti	11	12
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,003	
	Průměr	30,78%	46,09%
	Úspěšní studenti	4	6
	Neúspěšní studenti	4	3
<b>Strukturovaná část</b>	Hodnota p	0,272	
	Průměr	37,35%	44,46%
	Úspěšní studenti	7	5
	Neúspěšní studenti	3	3

Tabulka 63 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z teoretické části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Doplňování kódu</b>	Hodnota p	0,501	
	Průměr	50,82%	55,55%
	Úspěšní studenti	6	6
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,943	
	Průměr	80,76%	80,37%
	Úspěšní studenti	15	14

	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,397	
	Průměr	50,49%	54,41%
	Úspěšní studenti	8	9
	Neúspěšní studenti	1	1

Tabulka 64 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z objektové části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,092	
	Průměr	32,21%	44,41%
	Úspěšní studenti	5	5
	Neúspěšní studenti	8	4
<b>Metody a vlastnosti</b>	Hodnota p	0,109	
	Průměr	33,19%	43,49%
	Úspěšní studenti	5	7
	Neúspěšní studenti	5	5
<b>Instance a volání metod</b>	Hodnota p	0,012	
	Průměr	26,47%	51,47%
	Úspěšní studenti	5	8
	Neúspěšní studenti	11	5

Tabulka 65 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 ze strukturované části

Část testů	Sledované údaje	Posttest	Retest 1
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,084	
	Průměr	42,65%	59,71%

	Úspěšní studenti	8	12
	Neúspěšní studenti	4	3
<b>Návrh algoritmů metod</b>	Hodnota p	0,470	
	Průměr	34,71%	36,84%
	Úspěšní studenti	4	4
	Neúspěšní studenti	6	5

### Interpretace výsledků

Studenti kontrolní skupiny dosáhli oproti výsledkům z posttestu (studenti, kteří absolvovali i retest 1) v retestu 1 vyšší hodnoty skóre výsledků ve všech hlavních částech. Kromě dílčí části teoretické části Interaktivní prvky dosáhli studenti ve všech dílčích částech vyšší hodnoty skóre výsledků v retestu 1 než v posttestu.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí posttestu a retestu 1) značná až vysoká negativní závislost kromě dílčí části teoretické části Doplňování kódu, kde je mezi srovnávanými jevy závislost nízká.

### 6.2.6 Výsledky dotazníkového šetření

V příloze F v tabulce 140 je uvedeno vyhodnocení dotazníkového šetření studentů experimentální a kontrolní skupiny. Podle výsledků dotazníkového šetření u studentů, kteří anonymní dotazník vyplnili, je mezi vyučujícími statisticky významný rozdíl v otázkách číslo 1, 2, 4, 5, 8, 9, 10, 11, 14. Podle četnosti odpovědí v dotazníku a na základě výpočtu váženého průměru u daných otázek hodnotili studenti spíše pozitivně vyučujícího Kořínka. Vliv vyučujícího podle vyplněného dotazníkového šetření mohl mít vliv na výsledky studentů v kontrolní a experimentální skupině. V dotazníkovém šetření opět otázky č. 4 a č. 11 zjišťovaly stejné hledisko, jako v předcházejícím akademickém roce. Korelační koeficient Spearmanovy korelace u otázky č. 4 a č. 11 je u všech studentů 0,696, tj. mezi odpověďmi dvou daných otázek všech studentů je vysoká korelace. V experimentální skupině studentů je koeficient korelace 0,557. V kontrolní skupině je koeficient korelace 0,652. Mezi odpověďmi studentů kontrolní skupiny u otázek č. 4 a č. 11 je

vysoká závislost. U studentů experimentální skupiny u otázek č. 4 a č. 11 je mezi odpověďmi studentů také vysoká závislost. Podle výsledků je možné, že někteří studenti v obou skupinách mohli vyplňovat dotazníkové šetření náhodně. Koeficient korelace u otázek číslo 4 a číslo 11 vyšel u studentů obou skupin podobně jako v předcházejícím akademickém roce, tzn., že vyučující mohl mít vliv na výsledky studentů v jednotlivých skupinách. Statisticky významný rozdíl v dotazníkovém šetření je i u odpovědí studentů na jednu otázku (Otázka 16), která byla zaměřena na materiály, které měli studenti k dispozici v předmětu Programování 1.

### **6.3 Výsledky studentů z akademického roku 2014/2015 - bez studentů opakujících předmět**

V této kapitole jsou uvedeny výsledky studentů z akademického roku 2014/2015, kteří předmět Programování 1 absolvovali poprvé. Výsledky studentů, kteří v předmětu Programování 1 neuspěli v akademickém roce 2013/2014 a předmět Programování 1 absolvovali i v akademickém roce 2014/2015, nejsou uváděny.

#### **6.3.1 Výsledky studentů z posttestu**

Po vynechání studentů, kteří předmět Programování 1 absolvovali v akademickém roce 2013/2014 a v předmětu neuspěli, v akademickém roce 2014/2015 předmět Programování 1, respektive posttest, absolvovalo celkem 25 studentů. V experimentální skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 13 studentů. V kontrolní skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 12 studentů. Celkem tedy v první polovině semestru výuky předmět Programování 1 v akademickém roce 2014/2015 absolvovalo v obou skupinách celkem 9 studentů, kteří předmět absolvovali již v minulém akademickém roce. I když i při rozdělování studentů do skupin na začátku semestru byl brán ohled na to, aby v obou skupinách byli rovnoměrně zastoupení studenti, kteří již předmět Programování 1 absolvovali a měli z pretestu podobný výsledek, jejich nedisciplinovanost při rozdělování studentů do skupin s ohledem na stejný počet opakujících studentů v obou skupinách měla vliv. Rozdělení studentů po vynechání studentů opakujících předmět Programování 1 do skupin je uvedeno v tabulce 66.

**Tabulka 66 – Rozdělení studentů, bez studentů opakujících předmět Programování 1, do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	9	6
Velmi dobře	1	2
Dobře	1	3
Neuspěl	2	1
<b>Celkem studentů</b>	13	12

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V experimentální skupině byli tři studenti (se známkami z pretestu: 1x výborně a 2x dobře), kteří opakovali předmět Programování 1 z minulého akademického roku, viz předchozí kapitola 6.2.

V kontrolní skupině studentů bylo šest studentů (se známkami z pretestu: 1x výborně, 1x chvalitebně, 2x dobře a 2x neuspěl), kteří opakovali předmět Programování 1 z minulého akademického roku, viz předchozí kapitola 6.2.

V tabulce 67 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 68 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 69 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 70 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

V tabulce 128 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí posttestu. V tabulce 129 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu. V tabulce 130 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu.



Tabulka 67 – Výsledky studentů z posttestu z hlavních částí v akademickém roce 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	0,242	
	Průměr	39,20%	50,28%
	Úspěšní studenti	3	4
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,384	
	Průměr	28,49%	36,03%
	Úspěšní studenti	3	4
	Neúspěšní studenti	6	2
<b>Teoretická část</b>	Hodnota p	0,109	
	Průměr	49,91%	64,53%
	Úspěšní studenti	6	8
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,497	
	Průměr	34,39%	37,36%
	Úspěšní studenti	4	4
	Neúspěšní studenti	5	2
<b>Strukturovaná část</b>	Hodnota p	0,277	
	Průměr	24,55%	35,14%
	Úspěšní studenti	1	4
	Neúspěšní studenti	6	2

Tabulka 68 – Výsledky studentů z posttestu z dílčích částí teoretické části v akademickém roce 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,606	
	Průměr	33,90%	53,47%
	Úspěšní studenti	1	5
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,047	
	Průměr	65,25%	81,87%

	Úspěšní studenti	7	10
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,765	
	Průměr	39,10%	45,83%
	Úspěšní studenti	4	5
	Neúspěšní studenti	3	2

Tabulka 69 – Výsledky studentů z posttestu z dílčích částí objektové části v akademickém roce 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,369	
	Průměr	28,65%	42,50%
	Úspěšní studenti	4	5
	Neúspěšní studenti	8	5
<b>Metody a vlastnosti</b>	Hodnota p	0,265	
	Průměr	28,14%	41,11%
	Úspěšní studenti	4	4
	Neúspěšní studenti	6	2
<b>Instance a volání metod</b>	Hodnota p	0,355	
	Průměr	46,63%	29,17%
	Úspěšní studenti	7	4
	Neúspěšní studenti	5	8

Tabulka 70 – Výsledky studentů z posttestu z dílčích částí strukturované části v akademickém roce 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,605	
	Průměr	33,27%	39,17%
	Úspěšní studenti	4	5
	Neúspěšní studenti	6	3
<b>Návrh algoritmů metod</b>	Hodnota p	0,328	
	Průměr	20,19%	33,13%
	Úspěšní studenti	1	2
	Neúspěšní studenti	6	4

### Interpretace výsledků

Po vyfiltrování studentů, kteří předmět Programování 1 opakovali z minulého akademického roku, dosáhli ve všech hlavních částech posttestu vyšší hodnoty skóre výsledků studenti kontrolní skupiny. Oproti tomu ve všech hlavních částech posttestu se všemi studenty (i těmi, kteří předmět Programování 1 opakovali z minulého akademického roku) dosáhli v hlavní objektové části vyšší hodnoty skóre výsledků studenti experimentální skupiny, viz kapitola 6.2.1.

Ve všech dílčích částech kromě dílčí objektové části Instance a volání metod dosáhli vyšší hodnoty skóre výsledků z posttestu studenti kontrolní skupiny. Podobné výsledky dosáhli i všichni studenti (i s těmi, kteří předmět Programování 1 opakovali z minulého akademického roku). Oproti výsledkům se všemi studenty, viz kapitola 6.2.1, v dílčí teoretické části Interaktivní prvky je mezi výsledky studentů statisticky významný rozdíl.

Ve všech hlavních i dílčích částech posttestu dosáhli studenti experimentální skupiny bez studentů opakujících předmět Programování 1 nižší hodnoty skóre výsledků kromě dílčí teoretické části Doplnění hodnot proměnných než se všemi studenty (i se studenty opakujících předmět Programování 1 z minulého akademického roku).

Oproti tomu studenti kontrolní skupiny bez studentů opakujících předmět Programování 1 dosáhli v hlavních částech kromě strukturované části i dílčích částech posttestu kromě dílčí teoretické části Doplnění hodnot proměnných a dílčí strukturované části Naplnění a výpis posloupnosti vyšší hodnoty skóre výsledků než se všemi studenty (i se studenty opakujících předmět Programování 1).

Na základě dosažených výsledků studentů je těžké stanovit, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny. Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků pouze v dílčí části objektové části Instance a volání metod. V dalších částech objektové části dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny a oproti výsledkům v kapitole 6.2.1 (všichni studenti) byly rozdíly mezi skupinami větší.

Studenti experimentální skupiny i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Studenti kontrolní skupiny dosáhli ještě vysoké hodnoty skóre výsledků v dílčí části teoretické části Doplnění kódu.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí posttestu) značná negativní závislost, kromě dílčí teoretické části Doplnění kódu, kde je mezi jevy nízká závislost a dílčí teoretické části Doplnění hodnot proměnných, kde je mezi jevy vysoká závislost, tj. nižší známka z pretestu odpovídá vyšší hodnotě skóre jednotlivých částí z posttestu studentů.

Mezi studenty v kontrolní i experimentální skupině po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí posttestu) značná negativní závislost. U studentů v kontrolní skupině je v celkových výsledcích, teoretické části a její dílčí části Doplnění hodnot proměnných a dílčí objektové části Metody a vlastnosti vysoká závislost.

### **6.3.2 Výsledky studentů z posttestu, kteří absolvovali i retest 1**

V akademickém roce 2014/2015 předmět Programování 1, respektive posttest a zároveň retest 1, absolvovalo celkem 20 studentů. V experimentální skupině

studentů oba testy, tj. posttest a retest 1, absolvovalo celkem 9 studentů. V kontrolní skupině studentů oba testy, tj. posttest a retest 1, absolvovalo celkem 11 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 71.

**Tabulka 71 – Rozdělení studentů, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu**

Známka z pretestu	Experimentální skupina studentů	Kontrolní skupina studentů
Výborně	6	6
Velmi dobře	1	2
Dobře	1	3
Neuspěl	1	0
<b>Celkem studentů</b>	9	11

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 72 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 73 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 74 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 75 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

V tabulce 131 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí posttestu. V tabulce 132 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu. V tabulce 133 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu.

**Tabulka 72 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z hlavních částí**

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	<b>0,026</b>	
	Průměr	30,54%	53,54%
	Úspěšní studenti	0	4

	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,068	
	Průměr	16,76%	39,30%
	Úspěšní studenti	0	4
	Neúspěšní studenti	5	0
<b>Teoretická část</b>	Hodnota p	0,012	
	Průměr	44,33%	67,78%
	Úspěšní studenti	3	8
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,102	
	Průměr	20,93%	40,76%
	Úspěšní studenti	1	4
	Neúspěšní studenti	5	1
<b>Strukturovaná část</b>	Hodnota p	0,068	
	Průměr	13,98%	38,33%
	Úspěšní studenti	0	4
	Neúspěšní studenti	5	1

Tabulka 73 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí teoretické části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,006	
	Průměr	30,30%	57,89%
	Úspěšní studenti	0	5
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,008	
	Průměr	58,47%	84,00%
	Úspěšní studenti	4	10
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,403	
	Průměr	33,33%	50,00%

	Úspěšní studenti	2	5
	Neúspěšní studenti	2	1

Tabulka 74 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí objektové části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,087	
	Průměr	16,11%	46,36%
	Úspěšní studenti	2	5
	Neúspěšní studenti	7	4
<b>Metody a vlastnosti</b>	Hodnota p	0,052	
	Průměr	15,65%	44,85%
	Úspěšní studenti	1	4
	Neúspěšní studenti	5	1
<b>Instance a volání metod</b>	Hodnota p	0,849	
	Průměr	31,25%	31,82%
	Úspěšní studenti	4	4
	Neúspěšní studenti	5	7

Tabulka 75 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí strukturované části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,129	
	Průměr	18,89%	42,73%
	Úspěšní studenti	1	5
	Neúspěšní studenti	5	2
<b>Návrh algoritmů</b>	Hodnota p	0,095	

metod			
	Průměr	11,53%	36,14%
	Úspěšní studenti	0	2
	Neúspěšní studenti	5	3

### Interpretace výsledků

Mezi výsledky kontrolní a experimentální skupiny studentů je statisticky významný rozdíl v celkových výsledcích a v teoretické části i v jejich dílčích částech Doplňování kódu a Interaktivní prvky. Se všemi studenty (i s těmi, kteří opakovali předmět Programování 1) byl statisticky významný rozdíl mezi výsledky kontrolní a experimentální skupiny navíc v praktické části a ve strukturované části i její dílčí části Naplnění a výpis posloupnosti. Statisticky významný rozdíl se všemi studenty (těmi, kteří opakovali předmět Programování 1) nebyl v dílčí části teoretické části Doplňování kódu.

Ve všech částech posttestu dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny než studenti experimentální skupiny.

Studenti kontrolní skupiny dosáhli ve všech částech posttestu kromě dílčí teoretické části Doplnění hodnot proměnných vyšší hodnoty skóre výsledků než studenti kontrolní skupiny i se studenty, kteří opakovali předmět Programování 1.

Studenti experimentální skupiny dosáhli nižší hodnoty skóre výsledků v celkových výsledcích, teoretické části i jejich dílčích částech Doplňování kódu a Interaktivní prvky a v dílčí objektové části Instance a volání metod, než studenti experimentální skupiny i se studenty, kteří opakovali předmět Programování 1.

U výsledků studentů, kteří absolvovali pouze posttest, byl statisticky významný rozdíl pouze v dílčí části teoretické části Interaktivní prvky posttestu než u studentů v posttestu, kteří absolvovali i retest 1. U výsledků studentů, kteří absolvovali pouze posttest, dosáhli v posttestu vyšší hodnoty skóre výsledků studenti experimentální skupiny pouze v dílčí části objektové části Instance a volání metod než studenti kontrolní skupiny. Oproti tomu studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků ve všech dalších částech posttestu než studenti experimentální skupiny, kteří absolvovali i retest 1.



Studenti experimentální skupiny, kteří absolvovali i retest 1, dosáhli nižší hodnoty skóre výsledků v posttestu než všichni studenti, kteří absolvovali pouze posttest v dané skupině.

Oproti tomu studenti kontrolní skupiny, kteří absolvovali i retest 1, dosáhli vyšší hodnoty skóre výsledků v posttestu než všichni studenti, kteří absolvovali pouze posttest v dané skupině.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti experimentální skupiny i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Studenti kontrolní skupiny dosáhli vysoké hodnoty skóre výsledků i v dalších dílčích částech teoretické části.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí posttestu) střední negativní až nízká negativní závislost kromě dílčí teoretické části Doplňování kódu, kde je mezi srovnávanými jevy velmi nízká závislost.

Mezi studenty v kontrolní a experimentální skupině po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí posttestu) střední negativní až nízká závislost. U studentů v kontrolní skupině je v celkových výsledcích a v teoretické části i její dílčí části Doplnění hodnot proměnných mezi srovnávanými jevy vysoká závislost.

### **6.3.3 Výsledky studentů z retestu 1**

V akademickém roce 2014/2015 předmět Programování 1, respektive retest 1, absolvovalo celkem 20 studentů. V experimentální skupině studentů retest 1 absolvovalo celkem 9 studentů. V kontrolní skupině studentů retest 1 absolvovalo celkem 11 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 76.

**Tabulka 76 – Rozdělení studentů, kteří absolvovali retest 1, do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	6	6
Velmi dobře	1	2
Dobře	1	3
Neuspěl	1	0
<b>Celkem studentů</b>	<b>9</b>	<b>11</b>

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti experimentální skupiny.

V tabulce 77 je uvedeno zpracování výsledků studentů z hlavních částí retestu 1. V tabulce 78 je uvedeno zpracování výsledků studentů z teoretické části retestu 1. V tabulce 79 je uvedeno zpracování výsledků studentů z objektové části retestu 1. V tabulce 80 je uvedeno zpracování výsledků studentů ze strukturované části retestu 1.

V tabulce 134 je uveden Spearmanův koeficient pořadové korelace všech studentů z výsledku pretestu a jednotlivých částí retestu 1. V tabulce 135 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí retestu 1. V tabulce 136 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí retestu 1.

**Tabulka 77 – Výsledky studentů z retestu 1 z hlavních částí**

<b>Část retestu 1</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Celkové výsledky</b>	Hodnota p	0,129	
	Průměr	46,31%	62,46%
	Úspěšní studenti	3	6
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,095	
	Průměr	33,91%	55,95%

	Úspěšní studenti	2	4
	Neúspěšní studenti	1	0
<b>Teoretická část</b>	Hodnota p	0,171	
	Průměr	58,70%	68,97%
	Úspěšní studenti	4	8
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,305	
	Průměr	38,12%	55,66%
	Úspěšní studenti	3	5
	Neúspěšní studenti	1	0
<b>Strukturovaná část</b>	Hodnota p	0,068	
	Průměr	31,11%	56,14%
	Úspěšní studenti	1	4
	Neúspěšní studenti	2	0

Tabulka 78 – Výsledky studentů z retestu 1 z dílčích částí teoretické části

<b>Část retestu 1</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Doplňování kódu</b>	Hodnota p	0,087	
	Průměr	35,52%	55,50%
	Úspěšní studenti	1	3
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,149	
	Průměr	73,54%	82,15%
	Úspěšní studenti	7	9
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,939	
	Průměr	62,96%	59,09%
	Úspěšní studenti	5	6
	Neúspěšní studenti	0	0

Tabulka 79 – Výsledky studentů z retestu 1 z dílčích částí objektové části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,254	
	Průměr	33,33%	55,00%
	Úspěšní studenti	2	4
	Neúspěšní studenti	3	1
<b>Metody a vlastnosti</b>	Hodnota p	0,171	
	Průměr	33,49%	53,25%
	Úspěšní studenti	2	6
	Neúspěšní studenti	2	2
<b>Instance a volání metod</b>	Hodnota p	0,470	
	Průměr	48,61%	60,23%
	Úspěšní studenti	5	6
	Neúspěšní studenti	3	2

Tabulka 80 – Výsledky studentů z retestu 1 z dílčích částí strukturované části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,111	
	Průměr	48,06%	75,23%
	Úspěšní studenti	5	9
	Neúspěšní studenti	2	0
<b>Návrh algoritmů metod</b>	Hodnota p	0,081	
	Průměr	22,64%	46,59%
	Úspěšní studenti	0	3
	Neúspěšní studenti	4	1

## Interpretace výsledků

Mezi výsledky kontrolní a experimentální skupiny studentů není statisticky významný rozdíl ve skóre jednotlivých části retestu 1 (oproti statistickému rozdílu v dílčí části teoretické části Doplnění kódu mezi skupinami studentů i se studenty, kteří opakovali předmět Programování 1, viz kapitola 6.2.3). Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků ve všech částech retestu 1 kromě dílčí teoretické části Doplnění hodnot proměnných než studenti experimentální skupiny (oproti vyšší hodnotě skóre v objektové části studentů experimentální skupiny než studentů v kontrolní skupině, kteří opakovali předmět Programování 1, viz kapitola 6.2.3).

Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků v teoretické části retestu 1 i v jeho dílčích částech než studenti experimentální skupiny i se studenty, kteří opakovali předmět Programování 1. V ostatních částech retestu 1 byly výsledky studentů daných skupin s nebo bez studentů, kteří opakovali předmět Programování 1 opačné.

Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků ve všech částech retestu 1 kromě dílčí teoretické části Doplnění hodnot proměnných než studenti kontrolní skupiny i se studenty, kteří opakovali předmět Programování 1.

Na základě dosažených výsledků studentů je těžké stanovit, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti experimentální skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v jejích dílčích částech Interaktivní prvky a Doplnění hodnot proměnných. Studenti kontrolní skupiny dosáhli vysoké hodnoty skóre ve všech částech kromě dílčí teoretické části Návrh algoritmů metod.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí retestu 1) střední negativní až nízká negativní závislost kromě dílčí části teoretické části Interaktivní prvky a dílčí strukturované části Naplnění a výpis posloupnosti, kde je mezi jevy velmi slabá závislost.

Mezi studenty v kontrolní a experimentální skupině po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu

a výsledek jednotlivých částí posttestu) střední negativní až nízká závislost. U studentů v experimentální skupině v teoretické části i její dílčí části Doplnění kódu je mezi srovnávanými jevy vysoká závislost. U dílčí objektové části Definice konstruktoru a datových položek a dílčí teoretické části Interaktivní prvky je mezi srovnávanými jevy velmi malá závislost. U studentů kontrolní skupiny je v dílčí teoretické části Doplnění kódu a v dílčí strukturované části Naplnění a výpis posloupnosti mezi jevy velmi malá závislost.

### 6.3.4 Výsledky studentů z retestu 2

V akademickém roce 2014/2015 žádný student neopakoval retest 2 z předchozího akademického roku 2013/2014.

### 6.3.5 Srovnání výsledků studentů z posttestu a z retestu 1

V této podkapitole budou zvlášť srovnány výsledky studentů experimentální a kontrolní skupiny z posttestu a z retestu 1. Zajímavé bude sledovat, jak se po výuce společných témat změnila výsledky studentů v obou skupinách. Ke zjištění, jestli je mezi experimentální skupinou studentů mezi výsledky posttestu a retestu 1 statisticky významný rozdíl, byl opět použit Wilcoxonův neparametrický test.

#### Experimentální skupina studentů

V akademickém roce 2014/2015 posttest a retest 1 v předmětu Programování 1 v experimentální skupině studentů absolvovalo celkem 9 studentů.

Rozdělení studentů do skupin je uvedeno v tabulce 81.

Tabulka 81 – Rozdělení studentů v experimentální skupině

Známka z pretestu	Experimentální skupina studentů
Výborně	6
Velmi dobře	1
Dobře	1
Neuspěl	1
<b>Celkem studentů</b>	<b>9</b>

V tabulce 82 je uvedeno zpracování výsledků studentů experimentální skupiny z hlavních částí posttestu a retestu 1. V tabulce 83 je uvedeno zpracování výsledků studentů experimentální skupiny z teoretické části posttestu a retestu 1. V tabulce 84 je uvedeno zpracování výsledků studentů experimentální skupiny z objektové části posttestu a retestu 1. V tabulce 85 je uvedeno zpracování výsledků studentů experimentální skupiny ze strukturované části posttestu a retestu 1.

V tabulce 137 je uveden Spearmanův koeficient pořadové korelace studentů experimentální skupiny z výsledku pretestu a jednotlivých částí posttestu a retestu 1.

**Tabulka 82 – Výsledky studentů experimentální skupiny z hlavních částí posttestu a z retestu 1**

Část testů	Sledované údaje	Posttest	Retest 1
<b>Celkové výsledky</b>	Hodnota p	0,008	
	Průměr	30,54%	46,31%
	Úspěšní studenti	0	3
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,069	
	Průměr	16,76%	33,91%
	Úspěšní studenti	0	2
	Neúspěšní studenti	5	1
<b>Teoretická část</b>	Hodnota p	0,038	
	Průměr	44,33%	58,70%
	Úspěšní studenti	3	4
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,161	
	Průměr	20,93%	38,12%
	Úspěšní studenti	1	3
	Neúspěšní studenti	5	1
<b>Strukturovaná část</b>	Hodnota p	0,076	
	Průměr	13,98%	31,11%
	Úspěšní studenti	0	1
	Neúspěšní studenti	5	2

Tabulka 83 – Výsledky studentů experimentální skupiny z dílčích částí teoretické části z posttestu a z retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Doplňování kódu</b>	Hodnota p	0,859	
	Průměr	30,30%	35,52%
	Úspěšní studenti	1	0
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,066	
	Průměr	58,47%	73,54%
	Úspěšní studenti	4	7
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,038	
	Průměr	33,33%	62,96%
	Úspěšní studenti	2	5
	Neúspěšní studenti	2	0

Tabulka 84 – Výsledky studentů experimentální skupiny z dílčích částí objektové části z posttestu a z retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,249	
	Průměr	16,11%	33,33%
	Úspěšní studenti	2	2
	Neúspěšní studenti	7	3
<b>Metody a vlastnosti</b>	Hodnota p	0,091	
	Průměr	15,65%	33,49%
	Úspěšní studenti	1	2
	Neúspěšní studenti	5	2
<b>Instance a volání metod</b>	Hodnota p	0,225	
	Průměr	31,25%	48,61%



	Úspěšní studenti	4	5
	Neúspěšní studenti	5	3

**Tabulka 85 – Výsledky studentů experimentální skupiny z dílčích částí strukturované části z posttestu a z retestu**

Část testů	Sledované údaje	Posttest	Retest 1
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,052	
	Průměr	18,89%	48,06%
	Úspěšní studenti	1	5
	Neúspěšní studenti	5	2
<b>Návrh algoritmů metod</b>	Hodnota p	0,142	
	Průměr	11,53%	22,64%
	Úspěšní studenti	0	0
	Neúspěšní studenti	5	4

### Interpretace výsledků

Mezi výsledky studentů experimentální skupiny z posttestu a z retestu 1 je statisticky významný rozdíl v celkových výsledcích a v teoretické části i v její dílčí části Doplnění hodnot proměnných. Ve všech částech dosáhli studenti experimentální skupiny vyšší hodnoty skóre výsledků z retestu 1 oproti výsledkům z posttestu. Oproti výsledkům studentů experimentální skupiny i se studenty, kteří opakovali předmět Programování 1 z minulého akademického roku, již ve výsledcích studentů není statisticky významný rozdíl mezi praktickou částí i objektovou částí a její dílčí částí Metody a vlastnosti a strukturovanou částí i obou dvou jejích dílčích částí.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí posttestu a retestu 1) střední negativní až nízká negativní závislost kromě dílčí objektové části Definice konstrukturu a datových položek, kde je mezi srovnávanými jevy velmi nízká závislost.

### Kontrolní skupina studentů

V akademickém roce 2014/2015 posttest a retest 1 v předmětu Programování 1 v kontrolní skupině studentů absolvovalo celkem 11 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 86.

Tabulka 86 – Rozdělení studentů v kontrolní skupině

Známka z pretestu	Kontrolní skupina studentů
Výborně	6
Velmi dobře	2
Dobře	3
Neuspěl	0
<b>Celkem studentů</b>	<b>11</b>

V tabulce 87 je uvedeno zpracování výsledků studentů kontrolní skupiny z hlavních částí posttestu a retestu 1. V tabulce 88 je uvedeno zpracování výsledků studentů kontrolní skupiny z teoretické části posttestu a retestu 1. V tabulce 89 je uvedeno zpracování výsledků studentů kontrolní skupiny z objektové části posttestu a retestu 1. V tabulce 90 je uvedeno statistické zpracování výsledků studentů kontrolní skupiny ze strukturované části posttestu a retestu 1.

V tabulce 138 je uveden Spearmanův koeficient pořadové korelace studentů kontrolní skupiny z výsledku pretestu a jednotlivých částí posttestu a retestu 1.

Tabulka 87 – Výsledky studentů kontrolní skupiny z hlavních částí z posttestu a z retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Celkové výsledky</b>	Hodnota p	0,062	
	Průměr	53,54%	62,46%
	Úspěšní studenti	4	6
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,021	
	Průměr	39,30%	55,95%
	Úspěšní studenti	4	4
	Neúspěšní studenti	0	0
<b>Teoretická část</b>	Hodnota p	0,929	

	Průměr	67,78%	68,97%
	Úspěšní studenti	8	8
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,004	
	Průměr	40,76%	55,66%
	Úspěšní studenti	4	5
	Neúspěšní studenti	1	0
<b>Strukturovaná část</b>	Hodnota p	0,037	
	Průměr	38,33%	56,14%
	Úspěšní studenti	4	4
	Neúspěšní studenti	1	0

Tabulka 88 – Výsledky studentů kontrolní skupiny z dílčích částí teoretické části z posttestu a z retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Doplňování kódu</b>	Hodnota p	0,721	
	Průměr	57,89%	55,50%
	Úspěšní studenti	5	3
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,722	
	Průměr	84,00%	82,15%
	Úspěšní studenti	10	9
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,285	
	Průměr	50,00%	59,09%
	Úspěšní studenti	5	6
	Neúspěšní studenti	1	0

Tabulka 89 – Výsledky studentů kontrolní skupiny z dílčích částí objektové části z posttestu a z retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,161	
	Průměr	46,36%	55,00%
	Úspěšní studenti	5	4
	Neúspěšní studenti	4	1
<b>Metody a vlastnosti</b>	Hodnota p	0,477	
	Průměr	44,85%	53,25%
	Úspěšní studenti	4	6
	Neúspěšní studenti	1	2
<b>Instance a volání metod</b>	Hodnota p	0,028	
	Průměr	31,82%	60,23%
	Úspěšní studenti	4	6
	Neúspěšní studenti	7	2

Tabulka 90 – Výsledky studentů kontrolní skupiny z dílčích částí strukturované části z posttestu a z retestu 1

Část testů	Sledované údaje	Posttest	Retest 1
<b>Naplnění a výpis posloupností</b>	Hodnota p	0,008	
	Průměr	42,73%	75,23%
	Úspěšní studenti	5	9
	Neúspěšní studenti	2	0
<b>Návrh algoritmů metod</b>	Hodnota p	0,169	
	Průměr	36,14%	46,59%
	Úspěšní studenti	2	3
	Neúspěšní studenti	3	1

## **Interpretace výsledků**

Mezi výsledky studentů kontrolní skupiny z posttestu a z retestu 1 je statisticky významný rozdíl v praktické části, jak v objektové části i její dílčí části Instance a volání metod tak i ve strukturované části i její dílčí části Naplnění a výpis posloupnosti. Ve všech částech kromě dílčích teoretických částí Doplnění kódu a Interaktivní prvky dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků z retestu 1 oproti výsledkům z posttestu se všemi studenty, tj. i těmi, kteří předmět Programování 1 opakovali z minulého akademického roku, vyšší hodnoty skóre výsledků dosáhli studenti z posttestu pouze v jeho dílčí teoretické části Interaktivní prvky.

Oproti výsledkům studentů kontrolní skupiny i se studenty, kteří opakovali předmět Programování 1 z minulého akademického roku, je ve výsledcích studentů statisticky významný rozdíl v praktické části a ve strukturované části a její dílčí části Naplnění a výpis posloupnosti.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu a výsledek jednotlivých částí z posttestu a z retestu 1) střední negativní závislost kromě dílčí teoretické části Interaktivní prvky a dílčí strukturované části Naplnění a výpis posloupnosti, kde je mezi jevy nízká závislost a u dílčí teoretické části Doplnění kódu je mezi srovnávanými jevy velmi nízká závislost.

## **6.4 Výzkum výsledků studentů vyučovaných stejným vyučujícím**

V předcházejících kapitolách byly prezentovány výsledky výzkumu pedagogického experimentu koncepcí výuky programování objects-first a objects-later. Každou skupinu vyučoval jiný vyučující, což mohlo mít podle výsledků dotazníkového šetření především v akademickém roce 2014/2015 na celkové výsledky studentů vliv. V této kapitole budou prezentovány výsledky výzkumu experimentální skupiny studentů (koncepte výuky programování objects-first) a kontrolní skupiny studentů (koncepte výuky programování objects-later), kterou vyučoval stejný vyučující. V akademických letech 2013/2014 a 2014/2015 vyučoval každý vyučující jinou skupinu. Porovnání výsledků studentů experimentální

a kontrolní skupiny vyučované stejným vyučujícím je tedy z různých akademických let, což může vést k řadě nedostatků. Výsledky mohou ovlivňovat studenti, kteří předmět Programování 1 opakovali z předcházejícího akademického roku, podle výsledků z pretestu mohou být studenti v jednotlivých akademických letech rozdělení do skupin různým způsobem, v každé skupině mohou být rozdílné počty studentů apod. Výsledky tohoto výzkumu jsou uvedeny hlavně pro srovnání a dokreslení výsledků, které byly uvedeny v předcházejících kapitolách.

V tabulce 91 je uvedeno rozdělení studentů do skupin (experimentální a kontrolní skupina) z akademických let 2013/2014 a 2014/2015 u učitele Ing. Petra Voborníka, Ph.D.

**Tabulka 91 – Rozdělení studentů do dvou skupin podle výsledku z pretestu u vyučujícího Voborníka**

<b>Známka z pretestu</b>	<b>Kontrolní skupina studentů (2013/2014)</b>	<b>Experimentální skupina studentů (2014/2015)</b>
Výborně	3	10
Velmi dobře	2	1
Dobře	5	3
Neuspěl	4	2
<b>Celkem studentů</b>	14	16

Z tabulky je patrné, na základě výpočtu aritmetického průměru, že v experimentální skupině dosáhli studenti z pretestu lepších výsledků než v kontrolní skupině. Z tohoto důvodu nejsou uvedeny výsledky výzkumu studentů u výše uvedeného vyučujícího. Z výsledků prezentovaných v předcházejících kapitolách dopadli výrazně lépe studenti experimentální skupiny, což mohlo ovlivnit i rozdělení studentů do skupin podle výsledku z pretestu.

V této kapitole budou uvedeny pouze výsledky výzkumu studentů experimentální a kontrolní skupiny u vyučujícího, který je autorem disertační práce. V každé podkapitole bude uvedeno rozdělení studentů do skupin podle výsledků z pretestu a interpretace výsledků. Výsledky jednotlivých částí testů jsou uvedeny v příloze G. Studenti, kteří opakovali předmět Programování 1, jsou pouze v kontrolní skupině studentů v akademickém roce 2014/2015.

### 6.4.1 Výsledky studentů z posttestu

V akademických letech 2013/2014 a 2014/2015 předmět Programování 1, respektive posttest, absolvovalo celkem 32 studentů. V experimentální skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 14 studentů. V kontrolní skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 18 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 92.

**Tabulka 92 – Rozdělení studentů do dvou skupin podle výsledku z pretestu v akademickém roce 2014/2015**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	5	7
Velmi dobře	1	3
Dobře	5	5
Neuspěl	3	3
<b>Celkem studentů</b>	14	18

Na základě výpočtu váženého průměru o něco lepších výsledků z pretestu dosáhli studenti kontrolní skupiny.

V tabulce 141 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 142 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 143 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 144 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

#### **Interpretace výsledků**

Ve výsledcích mezi experimentální a kontrolní skupinou studentů v jednotlivých částech posttestu není statisticky významný rozdíl. Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků v praktické části jak objektové i ve všech jejích dílčích částech, tak i i strukturované i v její dílčí části Naplnění a výpis posloupnosti. Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků v teoretické části i ve všech jejích dílčích částech a v dílčí části strukturované části Návrh algoritmů metod.

Vzhledem k výsledkům z pretestu uspěli v posttestu lépe studenti experimentální skupiny.

Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Studenti experimentální skupiny dosáhli navíc vysoké hodnoty skóre výsledků i v dílčích částech objektové části Definice konstruktoru a datových položek a Instance a volání metod a v dílčí části strukturované části Naplnění a výpis posloupnosti.

Na základě dosažených výsledků studentů dosáhli vyšší hodnoty skóre výsledků studenti experimentální skupiny v objektové části, tj. v tématech, která studenti Programování 1 probírali dříve, i ve všech jejích dílčích částech než studenti kontrolní skupiny. Ve strukturované části dosáhli vyšší hodnoty skóre výsledků studenti obou skupin v jedné její dílčí části.

#### 6.4.2 Výsledky studentů, kteří absolvovali i retest 1, z posttestu

V akademických letech 2013/2014 a 2014/2015 předmět Programování 1, respektive posttest, absolvovalo celkem 27 studentů, kteří absolvovali i retest 1. V experimentální skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 10 studentů. V kontrolní skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 17 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 93.

**Tabulka 93 – Rozdělení studentů, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu**

Známka z pretestu	Experimentální skupina studentů	Kontrolní skupina studentů
Výborně	4	7
Velmi dobře	0	3
Dobře	4	5
Neuspěl	2	2
<b>Celkem studentů</b>	10	17

Na základě výpočtu váženého průměru o něco lepších výsledků dosáhli studenti kontrolní skupiny.

V tabulce 145 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 146 je uvedeno zpracování výsledků studentů z teoretické části



posttestu. V tabulce 147 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 148 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

### **Interpretace výsledků**

Ve výsledcích mezi experimentální a kontrolní skupinou studentů v jednotlivých částech posttestu není statisticky významný rozdíl. Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků v celkové části, v praktické objektové části i ve všech jejích dílčích částech i v praktické strukturované části a ve všech dílčích částech (u studentů, kteří absolvovali pouze posttest v dílčí strukturované části Návrh algoritmů metod dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny). Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků v teoretické části i ve všech jejích dílčích částech. Vzhledem k výsledkům pretestu uspěli v posttestu lépe studenti experimentální skupiny.

Studenti experimentální i kontrolní skupiny dosáhli ve všech částech posttestu vyšší hodnoty skóre výsledků pouze se studenty, kteří absolvovali i retest 1, než se studenty, kteří absolvovali pouze posttest.

Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Studenti kontrolní skupiny dosáhli i vysoké hodnoty skóre v dílčích částech teoretické části. Studenti experimentální skupiny dosáhli navíc vysoké hodnoty skóre výsledků i v dílčích částech objektové části Definice konstruktoru a datových položek a Instance a volání metod a v dílčí části strukturované části Naplnění a výpis posloupnosti.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

#### **6.4.3 Výsledky studentů z retestu 1**

V akademických letech 2013/2014 a 2014/2015 předmět Programování 1, respektive retest 1, absolvovalo celkem 27 studentů. V experimentální skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 10 studentů.

V kontrolní skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 17 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 94.

**Tabulka 94 – Rozdělení studentů, kteří absolvovali retest 1, do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	4	7
Velmi dobře	0	3
Dobře	4	5
Neuspěl	2	2
<b>Celkem studentů</b>	<b>10</b>	<b>17</b>

Na základě výpočtu váženého průměru o něco lepších výsledků dosáhli studenti kontrolní skupiny.

V tabulce 149 je uvedeno zpracování výsledků studentů z hlavních částí retestu 1. V tabulce 150 je uvedeno zpracování výsledků studentů z teoretické části retestu 1. V tabulce 151 je uvedeno zpracování výsledků studentů z objektové části retestu 1. V tabulce 152 je uvedeno zpracování výsledků studentů ze strukturované části retestu 1.

### **Interpretace výsledků**

Ve výsledcích mezi experimentální a kontrolní skupinou studentů v jednotlivých částech retestu 1 není statisticky významný rozdíl. Oproti výsledkům studentů experimentální a kontrolní skupiny v posttestu, v retestu 1 měli vyšší hodnoty skóre výsledků studenti kontrolní skupiny ve všech hlavních i dílčích částech retestu 1.

V experimentální skupině dosáhli studenti z retestu 1 kromě teoretické části a všech jejích dílčích částí a dílčí strukturované části Návrh algoritmů metod nižší hodnoty skóre výsledků než z posttestu.

Naopak, v kontrolní skupině dosáhli studenti v retestu 1 kromě dílčí teoretické části Interaktivní prvky vyšší hodnoty skóre výsledků než v posttestu.

Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v jejích dílčích částech Interaktivní prvky a Doplnění kódu. Studenti kontrolní skupiny dosáhli i vysoké hodnoty skóre výsledků v dílčí části teoretické části Doplnění hodnot proměnných a dílčí části

objektové části Instance a volání metod a dílčí části strukturované části Naplnění a výpis posloupnosti.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

#### **6.4.4 Výsledky studentů z posttestu bez studentů opakujících předmět Programování 1**

V akademických letech 2013/2014 a 2014/2015 předmět Programování 1, respektive posttest, absolvovalo celkem 26 studentů. V experimentální skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 14 studentů. V kontrolní skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 12 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 95.

**Tabulka 95 – Rozdělení studentů do dvou skupin podle výsledku z pretestu v akademickém roce 2014/2015**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	5	6
Velmi dobře	1	2
Dobře	5	3
Neuspěl	3	1
<b>Celkem studentů</b>	<b>14</b>	<b>12</b>

Na základě výpočtu váženého průměru lepších výsledků z pretestu dosáhli studenti kontrolní skupiny.

V tabulce 153 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 154 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 155 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 156 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

## **Interpretace výsledků**

Ve výsledcích mezi experimentální a kontrolní skupinou studentů v jednotlivých částech posttestu není statisticky významný rozdíl. Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků v praktické části i v části objektové a jejích dílčích částí kromě dílčí části Metody a vlastnosti i strukturované části a její dílčí části Naplnění a výpis posloupnosti. Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků v teoretické části, v celkových výsledcích i ve všech jejích dílčích částech, v dílčí části objektové části Metody a vlastnosti a v dílčí části strukturované části Návrh algoritmů metod.

Studenti v kontrolní skupině bez studentů opakujících předmět Programování 1 dosáhli vyšší hodnoty skóre výsledků v posttestu ve všech hlavních částech kromě strukturované části než se všemi studenty. V dílčích částech dosáhli nižší hodnoty skóre výsledků pouze v teoretické části Doplnění hodnot proměnných a strukturované části Naplnění a výpis posloupnosti.

Studenti, kteří opakovali předmět Programování 1 z minulého akademického roku, měli vliv na výsledné hodnoty skóre výsledků studentů z posttestu v celkově nižších výsledcích skóre výsledků v částech uvedených výše.

Studenti kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v jejích dílčích částech Interaktivní prvky a Doplnění kódu.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

### **6.4.5 Výsledky studentů z posttestu, kteří absolvovali i retest 1 bez studentů opakujících předmět Programování 1**

V akademických letech 2013/2014 a 2014/2015 předmět Programování 1, respektive posttest, absolvovalo celkem 21 studentů, kteří absolvovali i retest 1. V experimentální skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 10 studentů. V kontrolní skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 11 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 96.

**Tabulka 96 – Rozdělení studentů z posttestu, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	4	6
Velmi dobře	0	2
Dobře	4	3
Neuspěl	2	0
<b>Celkem studentů</b>	10	11

Na základě výpočtu váženého průměru výrazně lepších výsledků dosáhli studenti kontrolní skupiny. V tabulce 157 je uvedeno zpracování výsledků studentů z hlavních částí posttestu. V tabulce 158 je uvedeno zpracování výsledků studentů z teoretické části posttestu. V tabulce 159 je uvedeno zpracování výsledků studentů z objektové části posttestu. V tabulce 160 je uvedeno zpracování výsledků studentů ze strukturované části posttestu.

### **Interpretace výsledků**

Ve výsledcích mezi experimentální a kontrolní skupinou studentů v jednotlivých částech posttestu není mezi výsledky studentů statisticky významný rozdíl. Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků v praktické části, v části objektové i jejích dílčích částech kromě dílčí části Metody a vlastnosti i části strukturované kromě dílčí části Návrh algoritmů metod. Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků v teoretické části i ve všech jejích dílčích částech a v dílčích částech uvedených výše.

Studenti v kontrolní skupině, kteří absolvovali i retest 1, dosáhli vyšší hodnoty skóre výsledků v posttestu než studenti, kteří se zúčastnili pouze posttestu.

Studenti v kontrolní skupině bez studentů opakujících předmět Programování 1 dosáhli vyšší hodnoty skóre výsledků ve všech hlavních i dílčích částech kromě dílčí teoretické části Doplnění hodnot proměnných než všichni studenti (i ti, kteří opakovali předmět Programování 1).

Studenti kontrolní skupiny dosáhli nejvyšší hodnoty skóre v teoretické části a především v její dílčí části Interaktivní prvky. I v dalších částech teoretické části dosáhli studenti kontrolní skupiny vysokých hodnot skóre výsledků.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

#### **6.4.6 Výsledky studentů z retestu 1 bez studentů opakujících předmět Programování 1**

V akademických letech 2013/2014 a 2014/2015 předmět Programování 1, respektive retest 1, absolvovalo celkem 21 studentů. V experimentální skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 10 studentů. V kontrolní skupině v první polovině semestru výuku předmětu Programování 1 absolvovalo 11 studentů. Rozdělení studentů do skupin je uvedeno v tabulce 97.

**Tabulka 97 – Rozdělení studentů, kteří absolvovali retest 1, do dvou skupin podle výsledku z pretestu**

<b>Známka z pretestu</b>	<b>Experimentální skupina studentů</b>	<b>Kontrolní skupina studentů</b>
Výborně	4	6
Velmi dobře	0	2
Dobře	4	3
Neuspěl	2	0
<b>Celkem studentů</b>	10	11

Na základě výpočtu váženého průměru výrazně lepších výsledků dosáhli studenti kontrolní skupiny.

V tabulce 161 je uvedeno zpracování výsledků studentů z hlavních částí retestu 1. V tabulce 162 je uvedeno zpracování výsledků studentů z teoretické části retestu 1. V tabulce 163 je uvedeno zpracování výsledků studentů z objektové části retestu 1. V tabulce 164 je uvedeno zpracování výsledků studentů ze strukturované části retestu 1.

#### **Interpretace výsledků**

Ve výsledcích mezi experimentální a kontrolní skupinou studentů v jednotlivých částech retestu 1 není statisticky významný rozdíl. Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků ve všech hlavních i dílčích částech

posttestu než studenti experimentální skupiny. Bez studentů, kteří opakovali předmět Programování 1, dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků ve všech částech retestu 1 kromě dílčí teoretické části Doplnování kódu, než se všemi studenty (těmi, co předmět Programování 1 opakovali).

Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků v retestu 1 než v posttestu kromě dílčích částí teoretické části Doplnování kódu a Interaktivní prvky.

Studenti kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. I v dalších částech teoretické části dosáhli studenti kontrolní skupiny vysokých hodnot skóre výsledků. Vysoké hodnoty skóre dosáhli studenti kontrolní skupiny i v objektové části i všech jejích dílčích částech a v dílčí části teoretické části Naplnění a výpis posloupnosti.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

#### **6.4.7 Výsledky dotazníkového šetření**

V tabulce 165 je uvedeno vyhodnocení dotazníkového šetření studentů experimentální a kontrolní skupiny. Podle výsledků dotazníkového šetření u studentů, kteří anonymní dotazník vyplnili, není mezi jedním vyučujícím, materiály nebo přípravou na hodiny předmětu Programování 1 statisticky významný rozdíl. Korelační koeficient Spearmanovy korelace u všech studentů u otázek číslo 4 a číslo 11 je 0,735, tj. mezi odpověďmi dvou daných otázek všech studentů je vysoká korelace. V obou skupinách je předpoklad, že studenti vyplňovali dotazníkové šetření zodpovědně a ne náhodně.

## **7 SHRNUÍ VÝSLEDKŮ VLASTNÍHO VÝZKUMU**

V předcházející kapitole byly uvedeny výsledky vlastního výzkumu pedagogického experimentu experimentální skupiny studentů (skupina byla vyučována podle koncepce výuky programování objects-first) a kontrolní skupiny studentů (skupina byla vyučována podle koncepce výuky programování objects-later) v předmětu Programování 1 v akademických letech 2013/2014 a 2014/2015. V této kapitole budou dané výsledky shrnuty a budou uvedeny faktory, které mohou mít na výsledky studentů v testech vliv.

Kapitola je rozdělena na podkapitoly. V prvních třech podkapitolách jsou uvedeny tabulky, ve kterých jsou na základě výsledků studentů v jednotlivých částech testů testovány nulové hypotézy, tj. jestli je mezi výsledky studentů v částech testů statisticky významný rozdíl a tudíž jestli se může nulová hypotéza odmítnout nebo jestli se může nulová hypotéza nezamítnout. Hypotézy jsou uvedeny v kapitole 2. V příslušných tabulkách je nezamítnutí nulové hypotézy označeno číslem 1. Odmítnutí nulové hypotézy je v tabulce označeno číslem 0. Pokud nebyly v retestu 2 statisticky zpracovány výsledky studentů z testů, tj. nebylo možné dané hypotézy testovat, je toto v příslušných tabulkách označeno znakem x. Vzhledem k odlišné struktuře retestu 2 oproti posttestu a retestu 1 nejsou v retestu 2 testovány všechny hypotézy, které jsou uvedeny v kapitole 2. Ty hypotézy, které v rámci retestu 2 nebyly testovány, jsou v příslušných tabulkách označeny znakem -.

V podkapitole 7.3 jsou pro názornost uvedeny také tabulky, ve kterých jsou na základě výsledků studentů vyučovaných stejným vyučujícím v jednotlivých částech testů testovány nulové hypotézy, které jsou uvedeny v kapitole 2.

V podkapitole 7.4 jsou všechny výsledky výzkumu ještě komplexně shrnuty a jsou zde uvedeny faktory možného vlivu na výsledky studentů.

### **7.1 Akademický rok 2013/2014**

V této části jsou uvedeny výsledky testování nulových hypotéz podle výsledků studentů v testech v předmětu Programování 1 v akademickém roce 2013/2014.



**Tabulka 98 – Nezamítnutí/odmítnutí nulové hypotézy v testech**

Nulová hypotéza	Posttest	Retest 1	Retest 2
H1 <sub>0</sub>	1	1	x
H2 <sub>0</sub>	0	1	x
H3 <sub>0</sub>	1	1	x
H4 <sub>0</sub>	0	1	x
H5 <sub>0</sub>	0	1	x
H6 <sub>0</sub>	1	1	x
H7 <sub>0</sub>	1	1	x
H8 <sub>0</sub>	1	1	x
H9 <sub>0</sub>	0	1	x
H10 <sub>0</sub>	0	1	x
H11 <sub>0</sub>	1	1	x
H12 <sub>0</sub>	0	1	x
H13 <sub>0</sub>	1	1	x

Z tabulky 98 je patrné na základě dosažených výsledků studentů, že v posttestu byly odmítnuty nulové hypotézy H2<sub>0</sub>, H4<sub>0</sub>, H5<sub>0</sub>, H9<sub>0</sub>, H10<sub>0</sub> a H12<sub>0</sub> a byly přijaty alternativní hypotézy H2<sub>A</sub>, H4<sub>A</sub>, H5<sub>A</sub>, H9<sub>A</sub>, H10<sub>A</sub> a H12<sub>A</sub>.

Studenti koncepce výuky programování objects-first dosáhli odlišných výsledků z posttestu z praktické části, z praktické objektové části, z praktické strukturované části, z praktických objektových částí Definice konstruktora a datových položek a Metody a vlastnosti a z praktické strukturované části Naplnění a výpis posloupnosti než studenti koncepce výuky programování objects-later.

Dále je z tabulky 98 patrné na základě dosažených výsledků studentů, že v retestu 1 ve všech částech byla nulová hypotéza nezamítnuta.

Ve všech částech posttestu dosáhli studenti experimentální skupiny vyšší hodnoty skóre výsledků než studenti kontrolní skupiny. Pouze v teoretické části byly výsledky experimentální a kontrolní skupiny studentů podobné.

Studenti, kteří se již nezúčastnili retestu 1, ovlivnili výsledky studentů v obou skupinách posttestu. Studenti experimentální skupiny bez studentů, kteří již neabsolvovali retest 1, dosáhli ve všech částech posttestu vyšší hodnoty skóre výsledků než všichni studenti v experimentální skupině, kteří absolvovali pouze

posttest. Studenti kontrolní skupiny bez studentů, kteří již neabsolvovali retest 1, dosáhli vyšší hodnoty skóre výsledků ve všech částech posttestu kromě dílčích částí teoretické části Interaktivní prvky a Doplnění hodnot proměnných, dílčí části objektové části Metody a vlastnosti a dílčí části strukturované části Naplnění a výpis posloupnosti než všichni studenti, kteří absolvovali pouze posttest. Bez studentů, kteří absolvovali pouze posttest, v hlavních částech posttestu i v jeho jednotlivých dílčích částech dosáhli studenti experimentální skupiny vyšší hodnoty skóre výsledků než studenti kontrolní skupiny.

V retestu 1 dosáhli studenti experimentální skupiny také vyšší hodnoty skóre výsledků ve všech jeho částech kromě dílčí objektové části Definice konstrukturu a datových položek než studenti kontrolní skupiny.

Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků z posttestu než z retestu 1 kromě teoretické části i všech jejích dílčích částí a dílčí strukturované části Návrh algoritmů metod.

Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků z retestu 1 ve všech částech kromě dílčí teoretické části Doplnění hodnot proměnných než z posttestu.

Vzhledem k nízkému počtu studentů, kteří absolvovali retest 2 v předmětu Programování 1, nebyly výsledky studentů statisticky zpracovány.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti experimentální i kontrolní skupiny dosáhli v jednotlivých testech nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky.

Podle výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z pretestu studentů a výsledek jednotlivých částí posttestu a retestu 1 studentů) spíše značná až vysoká negativní závislost výsledků (v jednotlivých testech se jedná pouze o různé dílčí části), tj. nižší známka z pretestu studentů odpovídá vyšší hodnotě skóre studentů jednotlivých částí testů. Na daných výsledcích mají vliv výsledky hlavně experimentální skupiny studentů.

Podle výsledků dotazníkového šetření, které anonymně vyplňovali studenti, není mezi vyučujícími, materiály nebo přípravou studentů na hodiny předmětu Programování 1 statisticky významný rozdíl. V obou skupinách studentů je navíc předpoklad, že studenti vyplňovali dotazníkové šetření zodpovědně a ne náhodně (podle výpočtu hodnoty korelace podobně formulovaných otázek).

## 7.2 Akademický rok 2014/2015

V této části jsou uvedeny výsledky testování nulových hypotéz podle výsledků studentů v testech v předmětu Programování 1 v akademickém roce 2014/2015. Kapitola se sestává ze dvou podkapitol, kde v první podkapitole jsou uvedeny výsledky testování nulových hypotéz všech studentů a ve druhé kapitole jsou uvedeny výsledky testování nulových hypotéz pouze studentů, kteří neopakovali předmět Programování 1 z předešlého akademického roku.

### 7.2.1 Výsledky všech studentů

Tabulka 99 – Nezamítnutí/odmítnutí nulové hypotézy v testech

Nulová hypotéza	Posttest	Retest 1	Retest 2
H1 <sub>0</sub>	1	1	1
H2 <sub>0</sub>	1	1	1
H3 <sub>0</sub>	1	1	1
H4 <sub>0</sub>	1	1	1
H5 <sub>0</sub>	1	1	1
H6 <sub>0</sub>	1	0	1
H7 <sub>0</sub>	1	1	1
H8 <sub>0</sub>	1	1	1
H9 <sub>0</sub>	1	1	-
H10 <sub>0</sub>	1	1	1
H11 <sub>0</sub>	1	1	1
H12 <sub>0</sub>	1	1	-
H13 <sub>0</sub>	1	1	-

Z tabulky 99 je patrné na základě dosažených výsledků studentů, že v retestu 1 byla odmítnuta nulová hypotéza H6<sub>0</sub> a byla přijata alternativní hypotéza

H6<sub>A</sub>. Studenti koncepce výuky programování objects-first dosáhli odlišných výsledků z retestu 1 z teoretické části Doplnění kódu než studenti koncepce výuky programování objects-later.

Dále je z tabulky 99 patrné na základě dosažených výsledků studentů, že v posttestu i retestu 1 ve všech dalších částech byla nulová hypotéza nezamítnuta. Nepodařilo se prokázat, že ve výsledcích částí testů studentů mezi skupinami je statisticky významný rozdíl.

Ve všech hlavních i dílčích částech posttestu kromě objektové části i její dílčí části Instance a volání metod dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků než studenti experimentální skupiny.

Studenti, kteří se již nezúčastnili retestu 1, ovlivnili výsledky studentů v obou skupinách posttestu. V kontrolní skupině se jednalo o studenta, který neuspěl v pretestu. Studenti kontrolní skupiny (kteří absolvovali i retest 1) dosáhli vyšší hodnoty skóre výsledků ve všech částech posttestu než všichni studenti, kteří absolvovali pouze posttest. V experimentální skupině se retestu 1 nezúčastnili čtyři studenti, kteří z pretestu získali známku výborně. Studenti experimentální skupiny (kteří absolvovali i retest 1) dosáhli nižší (výrazně nižší) hodnoty skóre výsledků ve všech částech posttestu než všichni studenti, kteří absolvovali pouze posttest. Bez studentů, kteří absolvovali pouze posttest, ve všech částech posttestu dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny kromě dílčí objektové části Instance a volání metod než studenti experimentální skupiny.

V retestu 1 ve všech jeho částech kromě objektové části i její dílčí části Instance a volání metod a dílčí části teoretické části Doplnění hodnot proměnných dosáhli studenti kontrolní skupiny vyšší hodnoty skóre než studenti experimentální skupiny.

Studenti experimentální skupiny dosáhli vyšší hodnoty skóre v retestu 1 než v posttestu kromě dílčí části teoretické části Doplnění kódu.

Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre v retestu 1 ve všech hlavních částech kromě dílčí části teoretické části Interaktivní prvky než v posttestu.

Kromě dílčí teoretické části Interaktivní prvky (OOP) dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků ve všech částech retestu 2 než studenti experimentální skupiny. V retestu 2 nezáleží na tom, jestli studenti dané

skupiny dosáhli vyšší hodnoty skóre v tématech, které v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Na základě dosažených výsledků studentů je těžké stanovit, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny. V objektové části dosáhli celkově vyšší hodnoty skóre studenti experimentální skupiny, kteří objektová témata probírali na začátku předmětu Programování 1, i když se jedná pouze o jednu dílčí část objektové části. V ostatních dílčích částech byly výsledky daných skupin podobné, i když o něco vyšší hodnoty skóre výsledků dosáhli studenti kontrolní skupiny. Naopak, ve strukturované části i všech jejích dílčích částech dosáhli vyšší hodnoty skóre výsledků studenti kontrolní skupiny, kteří daná témata probírali na začátku předmětu Programování 1.

Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. V retestu 1 a retestu 2 dosáhli studenti obou skupin vysoké hodnoty skóre i v jiných částech testů.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy:

- známkou z pretestu a výsledků jednotlivých částí posttestu: značná až vysoká negativní závislost kromě dílčí části teoretické části posttestu Doplnování kódu. Na této závislosti mají vliv výsledky studentů hlavně kontrolní skupiny.
- známkou z pretestu a výsledků jednotlivých částí posttestu (pouze u studentů, kteří absolvovali i retest 1): značná negativní závislost ve všech částech posttestu kromě dílčí teoretické části Doplnování kódu. Na této značně vysoké až vysoké negativní závislosti mají vliv výsledky studentů hlavně kontrolní skupiny.
- známkou z pretestu a výsledků jednotlivých částí retestu 1: střední až nízká negativní závislost. U experimentální skupiny studentů je i v některých dílčích částech mezi srovnávanými jevy vysoká závislost nebo i velmi slabá závislost. U kontrolní skupiny studentů korelační koeficient nepotvrdil střední až nízkou negativní závislost

mezi jevy. U hlavních částí retestu 1 (kromě strukturované části) je mezi sledovanými jevy vysoká závislost.

- Známkou z pretestu a výsledků jednotlivých částí retestu 2: střední až nízká negativní závislost kromě strukturované části a dílčí teoretické části Interaktivní prvky (OOP), kde je mezi jevy velmi slabá negativní závislost.

Podle výsledků dotazníkového šetření, které anonymně vyplňovali studenti, je ve většině otázek zaměřené na vliv vyučujícího statisticky významný rozdíl. Vliv vyučujícího podle vyplněného dotazníkového šetření mohl mít vliv na výsledky studentů především v experimentální skupině. Podle výsledků je možné, že někteří studenti v obou skupinách mohli vyplňovat dotazníkové šetření náhodně (podle výpočtu hodnoty korelace podobně formulovaných otázek).

## 7.2.2 Výsledky studentů bez studentů opakujících předmět Programování 1

Tabulka 100 – Nezamítnutí/odmítnutí nulové hypotézy v testech

Nulová hypotéza	Posttest	Retest 1	Retest 2
H1 <sub>0</sub>	1	1	x
H2 <sub>0</sub>	1	1	x
H3 <sub>0</sub>	1	1	x
H4 <sub>0</sub>	1	1	x
H5 <sub>0</sub>	1	1	x
H6 <sub>0</sub>	1	1	x
H7 <sub>0</sub>	0	1	x
H8 <sub>0</sub>	1	1	x
H9 <sub>0</sub>	1	1	x
H10 <sub>0</sub>	1	1	x
H11 <sub>0</sub>	1	1	x
H12 <sub>0</sub>	1	1	x
H13 <sub>0</sub>	1	1	x

Z tabulky 100 je patrné na základě dosažených výsledků studentů, že v posttestu byla odmítnuta nulová hypotéza H7<sub>0</sub> a byla přijata alternativní hypotéza H7<sub>A</sub>. Studenti koncepce výuky programování objects-first dosáhli odlišných výsledků z posttestu z teoretické části Interaktivní prvky než studenti koncepce výuky programování objects-later.

Dále je z tabulky 100 patrné na základě dosažených výsledků studentů, že v posttestu i retestu 1 ve všech dalších částech byla nulová hypotéza nezamítnuta.

Ve všech částech posttestu kromě dílčí objektové části Instance a volání metod dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků než studenti experimentální skupiny.

Studenti kontrolní skupiny (bez studentů opakujících předmět Programování 1) dosáhli ve všech částech posttestu kromě strukturované části i její dílčí části Naplnění a výpis posloupnosti a dílčí teoretické části Doplnění hodnot proměnných vyšší hodnoty skóre výsledků než se studenty opakujících předmět Programování 1.

Studenti experimentální skupiny (bez studentů opakujících předmět Programování 1) dosáhli ve všech částech posttestu kromě dílčí teoretické části Doplnění hodnot proměnných nižší hodnoty skóre výsledků než se studenty opakujících předmět Programování 1.

Studenti, kteří se již nezúčastnili retestu 1, ovlivnili výsledky studentů v obou skupinách posttestu. Studenti kontrolní skupiny dosáhli ve všech částech posttestu vyšší hodnoty skóre výsledků než studenti experimentální skupiny.

Studenti experimentální skupiny bez studentů opakujících předmět Programování 1 (kteří absolvovali i retest 1), dosáhli v celkových výsledcích, teoretické části i v dílčích částech Doplnění kódu a Interaktivní prvky a v dílčí objektové části Instance a volání metod nižší hodnoty skóre výsledků než se studenty, kteří předmět Programování 1 opakovali z minulého akademického roku.

Studenti kontrolní skupiny bez studentů opakujících předmět Programování 1 (kteří absolvovali i retest 1) dosáhli ve všech částech posttestu kromě dílčí teoretické části Doplnění hodnot proměnných vyšší hodnoty skóre výsledků než se studenty, kteří předmět Programování 1 opakovali z minulého akademického roku.

Ve všech částech retestu 1 kromě dílčí teoretické části Doplnění hodnot proměnných dosáhli studenti kontrolní skupiny vyšší hodnoty skóre výsledků než studenti experimentální skupiny (oproti vyšší hodnotě skóre výsledků v objektové části studentů experimentální skupiny než studentů v kontrolní skupině, kteří opakovali předmět Programování 1).

Studenti experimentální skupiny dosáhli vyšší hodnoty skóre výsledků ve všech částech retestu 1 než v posttestu.

Studenti kontrolní skupiny dosáhli ve všech hlavních částech retestu 1 kromě dílčích částí teoretické části Doplnění kódu a Interaktivní prvky vyšší hodnoty skóre než v posttestu.

V akademickém roce 2014/2015 žádný student neopakoval retest 2 z předchozího akademického roku 2013/2014.

Na základě dosažených výsledků studentů v testech je těžké stanovit, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.



Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. Studenti kontrolní skupiny dosahovali i vysokých hodnot skóre výsledků v dalších dílčích částech teoretické části.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy:

- známkou z pretestu a výsledků jednotlivých částí posttestu: střední negativní závislost, kromě dílčí teoretické části posttestu Doplnění kódu, kde je mezi jevy závislost nízká a dílčí teoretické části Doplnění hodnot proměnných, kde je mezi jevy vysoká závislost. U studentů v kontrolní skupině je v celkových výsledcích, teoretické části a její dílčí části Doplnění hodnot proměnných a dílčí objektové části Metody a vlastnosti posttestu vysoká závislost.
- známkou z pretestu a výsledků jednotlivých částí posttestu (pouze u studentů, kteří absolvovali i retest 1): střední negativní až nízká negativní závislost. U dílčí teoretické části Doplnění kódu je mezi srovnávanými jevy velmi nízká závislost. U studentů v kontrolní skupině je v celkových výsledcích a v teoretické části i její dílčí části Doplnění hodnot proměnných mezi srovnávanými jevy vysoká závislost.
- známkou z pretestu a výsledků jednotlivých částí retestu 1: střední negativní až nízká negativní závislost. U dílčí teoretické části Interaktivní prvky a dílčí strukturované části Naplnění a výpis posloupnosti je mezi jevy velmi slabá závislost.

### 7.3 Výsledky studentů, které vyučoval stejný vyučující

V této části jsou uvedeny výsledky testování nulových hypotéz podle výsledků studentů v testech v předmětu Programování 1 v akademických letech 2013/2014 a 2014/2015. Kapitola se sestává ze dvou podkapitol, kde v první podkapitole jsou uvedeny výsledky testování nulových hypotéz všech studentů a ve druhé kapitole jsou uvedeny výsledky testování nulových hypotéz pouze studentů, kteří neopakovali předmět Programování 1 z předešlého akademického roku. Experimentální skupina studentů se vyučovala v akademickém roce 2013/2014 podle koncepce výuky programování objects-first. Kontrolní skupina studentů se vyučovala v akademickém roce 2014/2015 podle koncepce výuky programování objects-later.

#### 7.3.1 Výsledky se studenty, kteří opakovali předmět Programování 1

Tabulka 101 – Nezamítnutí/odmítnutí nulové hypotézy v testech

Nulová hypotéza	Posttest	Retest 1	Retest 2
H1 <sub>0</sub>	1	1	x
H2 <sub>0</sub>	1	1	x
H3 <sub>0</sub>	1	1	x
H4 <sub>0</sub>	1	1	x
H5 <sub>0</sub>	1	1	x
H6 <sub>0</sub>	1	1	x
H7 <sub>0</sub>	1	1	x
H8 <sub>0</sub>	1	1	x
H9 <sub>0</sub>	1	1	x
H10 <sub>0</sub>	1	1	x
H11 <sub>0</sub>	1	1	x
H12 <sub>0</sub>	1	1	x
H13 <sub>0</sub>	1	1	x

Z tabulky 101 je patrné na základě dosažených výsledků studentů, že v posttestu i retestu 1 ve všech částech byla nulová hypotéza nezamítnuta.

Studenti experimentální skupiny dosáhli v praktické objektové části i ve všech jejích dílčích částech a ve strukturované a její dílčí části Naplnění a výpis posloupnosti posttestu vyšší hodnoty skóre výsledků než studenti kontrolní skupiny. Studenti kontrolní skupiny dosáhli v celkových výsledcích, v teoretické části i ve všech jejích dílčích částech a i v dílčí strukturované části Návrh algoritmů metod posttestu vyšší hodnoty skóre výsledků než studenti experimentální skupiny. Vzhledem k výsledkům z pretestu uspěli v posttestu lépe studenti experimentální skupiny.

Studenti experimentální skupiny, kteří absolvovali i retest 1, dosáhli ve všech částech posttestu vyšší hodnoty skóre výsledků než studenti, kteří absolvovali pouze posttest.

Studenti kontrolní skupiny, kteří absolvovali i retest 1, dosáhli ve všech částech posttestu vyšší hodnoty skóre výsledků než studenti, kteří absolvovali pouze posttest.

Oproti výsledkům studentů experimentální a kontrolní skupiny v posttestu, v retestu 1 měli studenti kontrolní skupiny ve všech hlavních i dílčích částech retestu 1 vyšší hodnotu skóre výsledků než studenti experimentální skupiny.

V experimentální skupině dosáhli studenti kromě teoretické části a všech jejích dílčích částí i dílčí strukturované části Návrh algoritmů metod v retestu 1 nižší hodnoty skóre výsledků než v posttestu.

Naopak, v kontrolní skupině dosáhli studenti kromě dílčí teoretické části Interaktivní prvky v retestu 1 vyšší hodnoty skóre výsledků než v posttestu.

Studenti experimentální i kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v teoretické části a především v její dílčí části Interaktivní prvky. V posttestu studenti experimentální skupiny dosáhli navíc vysoké hodnoty skóre výsledků i v dílčích částech objektové části Definice konstruktoru a datových položek a Instance a volání metod a v dílčí části strukturované části Naplnění a výpis posloupnosti. V retestu 1 studenti kontrolní skupiny dosáhli i vysoké hodnoty skóre v dílčí části teoretické části Doplnění hodnot proměnných a dílčí části objektové části Instance a volání metod a dílčí části strukturované části Naplnění a výpis posloupnosti.

Na základě dosažených výsledků studentů je obtížné stanovit, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu

Programování 1 probírali dříve než studenti druhé skupiny. V posttestu dosáhli vyšší hodnoty skóre výsledků studenti experimentální skupiny v objektové části, tj. v tématech, která studenti Programování 1 probírali dříve, i ve všech jejích dílčích částech než studenti kontrolní skupiny.

Podle výsledků dotazníkového šetření, které anonymně vyplňovali studenti, není mezi vyučujícím v jednotlivých akademických letech, materiály nebo přípravou studentů na hodiny předmětu Programování 1 statisticky významný rozdíl. V obou skupinách studentů je navíc předpoklad, že studenti vyplňovali dotazníkové šetření zodpovědně a ne náhodně.

### 7.3.2 Výsledky studentů, kteří neopakovali předmět Programování 1

Tabulka 102 – Nezamítnutí/odmítnutí nulové hypotézy v testech

Nulová hypotéza	Posttest	Retest 1	Retest 2
H1 <sub>0</sub>	1	1	x
H2 <sub>0</sub>	1	1	x
H3 <sub>0</sub>	1	1	x
H4 <sub>0</sub>	1	1	x
H5 <sub>0</sub>	1	1	x
H6 <sub>0</sub>	1	1	x
H7 <sub>0</sub>	1	1	x
H8 <sub>0</sub>	1	1	x
H9 <sub>0</sub>	1	1	x
H10 <sub>0</sub>	1	1	x
H11 <sub>0</sub>	1	1	x
H12 <sub>0</sub>	1	1	x
H13 <sub>0</sub>	1	1	x

Z tabulky 102 je patrné na základě dosažených výsledků studentů, že v posttestu i retestu 1 ve všech částech byla nulová hypotéza nezamítnuta.

Studenti experimentální skupiny dosáhli v posttestu v praktické části a v části objektové kromě dílčí části Metody a vlastnosti i části strukturované kromě dílčí části Návrh algoritmů metod vyšší hodnoty skóre výsledků než studenti kontrolní skupiny studentů. Studenti kontrolní skupiny dosáhli v celkových výsledcích a v teoretické části i ve všech jejích dílčích částech a v dílčích částech uvedených výše vyšší hodnoty skóre výsledků než studenti experimentální skupiny.

Studenti v kontrolní skupině bez studentů opakujících předmět Programování 1 z předchozího akademického roku dosáhli vyšší hodnoty skóre výsledků v posttestu ve všech hlavních částech kromě strukturované části než se všemi studenty. V dílčích částech dosáhli vyšší hodnoty skóre výsledků ve všech hlavních i dílčích částech posttestu kromě teoretické části Doplnění hodnot proměnných a strukturované části Naplnění a výpis posloupnosti.

Stejných výsledků v posttestu, jak bylo uvedeno na začátku této části, dosáhli studenti daných dvou skupin, kteří absolvovali i retest 1.

Studenti v kontrolní skupině, kteří absolvovali i retest 1 bez studentů opakujících předmět Programování 1 dosáhli ve všech hlavních i dílčích částech posttestu kromě dílčí teoretické části Doplnění hodnot proměnných vyšší hodnoty skóre výsledků než všichni studenti (i ti, kteří opakovali předmět Programování 1).

Studenti kontrolní skupiny dosáhli ve všech hlavních i dílčích částech retestu 1 vyšší hodnoty skóre výsledků než studenti experimentální skupiny.

Bez studentů, kteří opakovali předmět Programování 1, dosáhli studenti kontrolní skupiny ve všech částech retestu 1 kromě dílčí teoretické části Doplnění kódu vyšší hodnoty skóre, než se všemi studenty (těmi, co předmět Programování 1 opakovali). Výsledky studentů v retestu 1 mohou být ovlivněny lepšími výsledky z pretestu studentů kontrolní skupiny.

Studenti kontrolní skupiny dosáhli vyšší hodnoty skóre výsledků v retestu 1 než v posttestu kromě dílčích částí teoretické části Doplnění kódu a Interaktivní prvky.

Na základě dosažených výsledků studentů nezáleží na tom, jestli studenti dané skupiny dosáhli vyšší hodnoty skóre výsledků v tématech, která v předmětu Programování 1 probírali dříve než studenti druhé skupiny.

Studenti kontrolní skupiny dosáhli nejvyšší hodnoty skóre výsledků v testech v teoretické části a především v její dílčí části Interaktivní prvky. V retestu 1 v jeho dalších částech teoretické části dosáhli studenti kontrolní skupiny také vysokých hodnot skóre výsledků. Vysoké hodnoty skóre výsledků dosáhli studenti kontrolní skupiny i v objektové části i všech jejích dílčích částech a v dílčí části teoretické části Naplnění a výpis posloupnosti.

## **7.4 Závěrečné komplexní shrnutí výsledků vlastního výzkumu z jednotlivých akademických let**

V akademickém roce 2013/2014 uspěli výrazně lépe v posttestu kromě teoretické části i v retestu 1 kromě dílčí objektové části Definice konstrukturu a datových položek v předmětu Programování 1 studenti koncepce výuky programování objects-first (experimentální skupina). V posttestu byly odmítnuty nulové hypotézy  $H_{2_0}$ ,  $H_{4_0}$ ,  $H_{5_0}$ ,  $H_{9_0}$ ,  $H_{10_0}$  a  $H_{12_0}$  a byly přijaty příslušné alternativní hypotézy.

V akademickém roce 2014/2015 uspěli lépe v posttestu kromě objektové části i její dílčí části objektové části Instance a volání metod i v retestu 1 kromě objektové části i její dílčí části Instance a volání metod a dílčí části teoretické části Doplnění hodnot proměnných v předmětu Programování 1 studenti koncepce výuky programování objects-later (kontrolní skupina). V retestu 1 byla odmítnuta nulová hypotéza  $H_{6_0}$  a byla přijata alternativní hypotéza  $H_{6_A}$ . Také v retestu 2 uspěli lépe kromě dílčí teoretické části Interaktivní prvky (OOP) studenti kontrolní skupiny.

Studenti, kteří v akademickém roce 2014/2015 opakovali předmět Programování 1 z minulého akademického roku, měli na výsledky všech studentů vliv. V posttestu byla odmítnuta nulová hypotéza  $H_{7_0}$  a byla přijata alternativní hypotéza  $H_{7_A}$ .

Bez studentů opakujících předmět Programování 1 z minulého akademického roku dosáhli studenti experimentální skupiny v posttestu v jeho jednotlivých částech kromě dílčí teoretické části Doplnění hodnot proměnných nižší hodnoty skóre výsledků než se všemi studenty, tj. i těmi, kteří předmět Programování 1 opakovali z minulého akademického roku. Naopak, studenti kontrolní skupiny bez studentů opakujících předmět Programování 1 z minulého akademického roku dosáhli v jednotlivých částech posttestu kromě dílčí teoretické části Doplnění hodnot proměnných a strukturované části i její dílčí části Naplnění a výpis posloupnosti vyšší hodnotu skóre než se všemi studenty tj. i těmi, kteří předmět Programování 1 opakovali z minulého akademického roku.

Bez studentů opakujících předmět Programování 1 z minulého akademického roku dosáhli studenti koncepce výuky programování objects-first (experimentální skupina) v retestu 1 v jeho jednotlivých částech kromě teoretické části i jejích dílčích

částí nižší hodnoty skóre výsledků než se všemi studenty, tj. i těmi, kteří předmět Programování 1 opakovali z minulého akademického roku. Naopak, studenti výuky koncepce programování objects-later (kontrolní skupina) bez studentů opakujících předmět Programování 1 z minulého akademického roku dosáhli v jednotlivých částech retestu 1 v jeho jednotlivých částech kromě dílčí teoretické části Doplnění hodnot proměnných vyšší hodnoty skóre výsledků než se všemi studenty, tj. i těmi, kteří předmět Programování 1 opakovali z minulého akademického roku.

Studenti, kteří absolvovali pouze posttest, jak v akademickém roce 2013/2014 tak i v akademickém roce 2014/2015 ovlivnili celkové výsledky posttestu v obou skupinách. Studenti, kteří absolvovali i retest 1, dosáhli v posttestu obvykle vyšší hodnoty skóre výsledků než všichni studenti, kteří absolvovali pouze posttest. Výjimku tvořili pouze studenti v akademickém roce 2014/2015, kteří v posttestu uspěli a odmítli absolvovat retest 1. Jednalo se o studenty, kteří se vyučovali pomocí koncepce výuky programování objects-first.

Dotazníkové šetření neprokázalo mezi vyučujícími vliv na výsledky studentů v akademickém roce 2013/2014. V akademickém roce 2014/2015 podle výsledků dotazníkového šetření mohl mít na výsledky studentů (především horších výsledků experimentální skupiny) vliv vyučující.

Srovnáním výsledků studentů z akademických let 2013/2014 a 2014/2015, které vyučoval stejný vyučující, uspěli v posttestu v praktické části (jak v části strukturované i její dílčí části Naplnění a výpis posloupnosti, tak i objektové) lépe studenti koncepce výuky programování objects-first (experimentální skupina). V teoretické části posttestu uspěli lépe studenti koncepce výuky programování objects-later (kontrolní skupina). V retestu 1 a ve všech jeho částech dosáhli lepších výsledků studenti kontrolní skupiny. Výsledky ale mohly být ovlivněny rozdělením studentů do skupin, kde v kontrolní skupině dosáhli studenti z pretestu lepších výsledků. Studenti, kteří neuspěli v předmětu Programování 1 v akademickém roce 2013/2014 ovlivnili výsledky studentů kontrolní skupiny v akademickém roce 2014/2015 v posttestu i v retestu 1, kde studenti kontrolní skupiny bez těchto studentů dosáhli obvykle vyšší hodnoty skóre výsledků v jednotlivých částech testů.

Podle výsledků z akademických let 2013/2014 a 2014/2015 se nedá stanovit, jestli studenti v kontrolní nebo experimentální skupině mají v testech vyšší hodnoty



skóre v tématech, kterými se zabývali nejdříve v předmětu Programování 1 a mohli je tak více zažít než studenti druhé skupiny.

Nejvyšší hodnoty skóre výsledků dosahovali studenti v teoretických částech testů, především v dílčí části Interaktivní prvky. Menší hodnoty skóre dosahovali studenti v praktických částech testů, jak objektové části, tak i strukturované části. Ve strukturované části dosahovali studenti vyšší hodnoty skóre výsledků v části Naplnění a výpis posloupnosti. U objektové části dosahovali studenti vyšší hodnoty skóre výsledků v částech Definice konstrukturu a datových položek a Instance a volání metod.

Po výpočtu Spearmanova koeficientu pořadové korelace je mezi srovnávanými jevy (známka z posttestu a výsledcích studentů z posttestu a retestu 1) v akademickém roce 2013/2014 spíše značná až vysoká negativní závislost. To znamená, že lepší (nižší hodnota známky) známka z předmětu ALGDS odpovídá u studentů vyšší hodnotě skóre v jednotlivých částech testů. V akademickém roce 2014/2015 je mezi srovnávanými jevy v pretestu a posttestu značná až vysoká negativní závislost. To znamená, že lepší (nižší hodnota známky) známka z předmětu ALGDS odpovídá u studentů vyšší hodnotě skóre v jednotlivých částech testů. Mezi srovnávanými jevy v pretestu a retestu 1 i retestu 2 studentů v akademickém roce 2014/2015 je střední až nízká negativní závislost. To znamená, že lepší známka studentů v předmětu ALGDS nemusí odpovídat vyšší hodnotě skóre v jednotlivých částech testů.

**Z výše uvedených výsledků je pro studenty vhodnější, aby se v předmětu Programování 1 vyučovali podle koncepce výuky programování objects-first.**

## ZÁVĚR

Výuka programování bývá náročná nejen pro studenty, ale i pro vyučující. Ti musejí zvolit vhodný styl programování především v úvodu předmětu programování. Zvolený styl programování určuje, jakým způsobem budou studenti přistupovat k řešení problémů a jak budou vytvářet programy, tzn., jak budou programovat. V příslušném stylu programování bývá i více možností, jak vyřešit daný problém. Jedná se především o různé pořadí probíraných témat výuky programování, respektive o koncepci výuky programování.

Stylů programování i koncepcí výuky programování existuje velké množství, a jsou značně obsáhlé. V disertační práci jsou uvedeny především dva styly programování: objektově orientované paradigma programování a imperativní paradigma programování (strukturované programování) a dvě koncepce výuky programování objects-first a objects-later, ve kterých se dané dva styly programování využívají.

Úvodní část disertační práce byla věnována problematice výuky programování, na jejímž základě byl stanoven hlavní cíl disertační práce. Hlavním cílem disertační práce bylo zjistit, jak různé pořadí probíraných témat (koncepce výuky programování) ovlivňuje výsledky studentů 1. ročníku oboru učitelství informatiky na PřF UHK v předmětu Programování 1 v programovacím jazyku C#. Ke splnění hlavního cíle disertační práce bylo nutné popsat teoretická východiska, mezi která patří např. paradigmatu programování nebo koncepce výuky programování. U různých koncepcí výuky programování je předpoklad, že se v nich studenti seznámí i s objektově orientovaným paradigmatem programování, které je klíčovou kompetencí programátorů v návrzích programů.

S ohledem na množství koncepcí výuky programování, které uvádějí různí autoři v odborných publikacích, byly koncepce výuky programování klasifikovány podle toho, zda na začátku výuky programování preferují objektově orientované paradigma programování nebo preferují nejdříve jiný styl programování.

K dosažení hlavního cíle disertační práce vedly dílčí cíle disertační práce, které byly uvedeny v kapitole 2. Dílčí cíle disertační práce byly zformulovány pomocí výzkumných otázek uvedených v úvodní části disertační práce. Na základě

stanovených dílčích cílů disertační práce byl zformulován soubor 13-ti konkrétních nulových a alternativních hypotéz.

Vlastní výzkum ve formě pedagogického experimentu vycházel ze tří výzkumných studií (viz kap. 3.4), především pak ze studie autorů Ehlertha a Schulteho (2009), ovšem s určitými odlišnostmi. Na rozdíl od této studie nebyly do výzkumu v rámci disertační práce zahrnuty dědičnost, abstraktní metody a asociace. Základní konstrukce byly realizovány s jednorozměrným a vícerozměrným polem.

I rozdělení studentů do skupin bylo od uvedených studií odlišné. Bylo zvoleno i jiné vývojové prostředí a programovací jazyk. Proti výše uvedeným autorům, kteří znalosti a dovednosti studentů ověřovali různými typy zkoušek i úloh, v pedagogickém experimentu popsáném v disertační práci byly studentské znalosti a dovednosti ověřovány pomocí různých typů testů i úloh.

I když se vlastní výzkum odlišoval od výzkumů ve zmíněných studiích, výsledky studentů v posttestu v akademickém roce 2013/2014 podpořily, především v praktické části, výsledky výzkumů autorů Johnsona a Mosese (2008) a Uysala (2012), kde výrazněji uspěli v závěrečné zkoušce a v programovací úloze, jak bylo u výše uvedených autorů uvedeno v kapitolách 3.4.4 a 3.4.5, studenti vyučovaní podle koncepce výuky programování objects-first.

V rámci disertační práce byly vytvořeny obsahové náplně koncepcí výuky programování objects-first i objects-later. U obou koncepcí výuky programování byly zastoupeny stejné tematické okruhy. Lišil se především jejich způsob zápisu v programu a u úvodní demonstrace pojmů z objektově orientovaného paradigmatu programování se částečně lišily příklady, na kterých byly pojmy demonstrovány. Dále byly vytvořeny testy, kde byly hodnoceny základní konstrukce typické v koncepcích výuky programování objects-first a objects-later. V rámci vlastního výzkumu byly v testech porovnány základní programové konstrukce koncepcí výuky programování objects-first a objects-later. Hlavní i dílčí cíle disertační práce byly splněny.

V rámci vlastního výzkumu, s ohledem na dva různé vyučující, kteří v akademických letech 2013/2014 a 2014/2015 vyučovali studenty v experimentální a kontrolní skupině, studenti na konci semestru vyplnili anonymní dotazníkové šetření. Dotazníkové šetření neprokázalo mezi vyučujícími vliv na výsledky studentů

v akademickém roce 2013/2014. V akademickém roce 2014/2015 podle výsledků dotazníkového šetření mohl mít na výsledky studentů (především horších výsledků experimentální skupiny) vliv vyučující. Podle statistického zpracování odpovědí v experimentální a kontrolní skupině studentů v dotazníkovém šetření byl v některých otázkách zaměřených na vliv vyučujícího zjištěn významný rozdíl. Vyučující (autor disertační práce) v kontrolní skupině studentů byl studenty hodnocen spíše pozitivní variantou odpovědí. Druhý vyučující v experimentální skupině byl studenty hodnocen spíše negativní variantou odpovědí nebo neutrální variantou odpovědí. Oba vyučující měli pravidelné konzultace např. ohledně zvolených metod výuky a obsahové náplně předmětu Programování 1. Rozdíl mezi vyučujícími mohl být ve způsobu demonstrace příkladů a způsobem vysvětlování základních programátorských konstrukcí, v důrazu na odlišné programátorské konstrukce nebo v rychlosti výkladu. Dále byly srovnány, i s vědomím nedostatků, které byly uvedeny v kapitole 6.4, pro dokreslení výsledků z jednotlivých akademických let, výsledky studentů z akademických let 2013/2014 a 2014/2015, které vyučoval stejný vyučující.

V akademickém roce 2013/2014 jeden z vyučujících vyučoval experimentální skupinu studentů podle koncepce výuky programování objects-first a druhý vyučující vyučoval kontrolní skupinu studentů podle koncepce výuky programování objects-later. V následujícím akademickém roce 2014/2015 si vyučující dané skupiny prohodili. V rámci vlastního výzkumu tak oba dva vyučující vyučovali jak experimentální skupinu studentů, tak i kontrolní skupinu studentů. Porovnatelnost daných dvou skupin studentů vyučovaných stejným vyučujícím byla dána rozdělením studentů do dvou skupin podle jejich výsledků z pretestu. S ohledem na porovnatelnost skupin byly srovnány výsledky pouze experimentální a kontrolní skupiny studentů, kterou vyučoval autor disertační práce. Na základě získaných výsledků uspěli v předmětu Programování 1 lépe studenti experimentální skupiny, kteří byli vyučováni podle koncepce výuky programování objects-first.

Dále bylo zkoumáno, v jaké části a v jakém typu testu dosáhli studenti v experimentální a v kontrolní skupině nejvyšší hodnoty skóre výsledků a jestli studenti daných dvou koncepcí výuky programování dosahují vysoké hodnoty skóre

výsledků v tématech, která probírali nejdříve v pedagogickém experimentu v předmětu Programování 1.

Podle výsledků z akademických let 2013/2014 a 2014/2015 se nedá stanovit, zda studenti v kontrolní nebo experimentální skupině mají v testech vyšší hodnoty skóre výsledků v tématech, kterými se zabývali nejdříve v předmětu Programování 1 a mohli se s nimi více seznámit, než studenti druhé skupiny.

Nejvyšší hodnoty skóre výsledků dosahovali studenti v teoretických částech testů, především v dílčí části Interaktivní prvky. Nižší hodnoty skóre dosahovali studenti v praktických částech testů, jak objektové části, tak i strukturované části. Ve strukturované části dosahovali studenti vyšší hodnoty skóre výsledků v části Naplnění a výpis posloupnosti. U objektové části dosahovali studenti vyšší hodnoty skóre výsledků v částech Definice konstruktoru a datových položek a Instance a volání metod.

Po výpočtu Spearmanova koeficientu pořadové korelace byla mezi srovnávanými jevy (známka z pretestu a výsledcích studentů z posttestu a retestu 1) v akademickém roce 2013/2014 spíše značná až vysoká negativní závislost. To znamená, že lepší (nižší hodnota známky) známka z předmětu ALGDS, který byl pretestem pedagogického experimentu, odpovídá u studentů vyšší hodnotě skóre v jednotlivých částech testů. V akademickém roce 2014/2015 byla mezi srovnávanými jevy v pretestu a posttestu značná až vysoká negativní závislost. To znamená, že lepší (nižší hodnota známky) známka z předmětu ALGDS odpovídá u studentů vyšší hodnotě skóre v jednotlivých částech testů.

**Jedním z přínosů disertační práce je zjištění, že různé pořadí probíraných témat (koncepte výuky programování) ovlivňuje výsledky studentů 1. ročníku oboru učitelství informatiky na PŘF UHK v předmětu Programování 1 v programovacím jazyku C#. Ze získaných výsledků plyne, že pro studenty je spíše vhodnější, aby se v předmětu Programování 1 vyučovali podle koncepte výuky programování objects-first.**

**Za další přínos disertační práce považuji vypracování kategorizace konceptů výuky programování na základě v teoretické části zpracované rešerše.**

## SEZNAM POUŽITÉ LITERATURY

ASAGBA, Prince Oghenekaro a Edward E. OGHENEVO, 2008. A Comparative Analysis of Structured and Object-Oriented Programming Methods. *Journal of Applied Sciences and Environmental Management* [online]. **12**(4), 41-46 [cit. 2016-04-20]. ISSN 1119-8362. Dostupné z: <http://www.bioline.org.br/pdf?ja08064>

ASSOCIATION FOR COMPUTING MACHINERY a IEEE COMPUTER SOCIETY, 2001. *Computing Curricula 2001: Computer Science* [online]. [cit. 2016-07-10]. Dostupné z: [http://www.acm.org/education/curric\\_vols/cc2001.pdf](http://www.acm.org/education/curric_vols/cc2001.pdf)

ASSOCIATION FOR COMPUTING MACHINERY a IEEE COMPUTER SOCIETY, 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science* [online]. United States of America [cit. 2016-07-10]. ISBN 978-1-4503-2309-3. Dostupné z: <https://www.acm.org/education/CS2013-final-report.pdf>

BIELIKOVÁ, Mária a Pavol NÁVRAT, 2009. *Funkcionálne a logické programovanie* [online]. Slovenská technická univerzita v Bratislave: STU, s. 1-7 [cit. 2016-12-01]. ISBN 978-80-227-3225-3. Dostupné z: [https://www.researchgate.net/publication/268432411\\_Funkcionalne\\_a\\_logicke\\_programovanie](https://www.researchgate.net/publication/268432411_Funkcionalne_a_logicke_programovanie)

BOLSHAKOVA, Elena, 2005. Programming Paradigms in Computer Science Education. *Information Theories & Applications* [online]. **12**(3), 285-290 [cit. 2016-05-08]. ISSN 1313-0463. Dostupné z: <http://www.foibg.com/ijita/vol12/ijita12-3-p13.pdf>

BURTON, Philip J. a Russel E. BRUHN, 2003. Teaching programming in the OOP era. *ACM SIGCSE Bulletin*. New York: ACM New York, **35**(2), 111-114. DOI: 10.1145/782941.782993 cit. IN: LISTER, Raymond et al., 2006. Research perspectives on the objects-early debate. In: *ITiCSE-WGR '06 Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education*. New York, 146 - 165. DOI: 10.1145/1189215.1189183. ISBN 1-59593-603-3.

Dostupné také z:

<https://www.cs.auckland.ac.nz/courses/compsci747s2c/lectures/p146-lister.pdf>

ČADA, Ondřej, 2009. *Objektové programování: naučte se pravidla objektového myšlení*. 1. vydání. Praha: Grada. ISBN 978-80-247-2745-5.

DEHNADI, Saeed a Richard BORNAT, 2006. *The camel has two humps (working title)* [online]. 1-17 [cit. 2016-12-11]. Dostupné z:

<https://pdfs.semanticscholar.org/02cf/8dd74bf6f25c777cd53dfb57efc52367b07f.pdf>

DOHNAL, Pavel, 2009. *Programování na základních školách* [online]. Brno [cit. 2016-07-10]. Dostupné z:

[http://is.muni.cz/th/72886/pdf\\_m/Diplomova\\_prace\\_Pavel\\_Dohnal.pdf](http://is.muni.cz/th/72886/pdf_m/Diplomova_prace_Pavel_Dohnal.pdf). Diplomová práce. Vedoucí práce Ing. Martin Dosedla.

DONCHEV, Ivaylo a Emilia TODOROVA, 2008. Object-Oriented Programming in Bulgarian Universities' Informatics and Computer Science Curricula. *Informatics in Education* [online]. 7(2), 159-172 [cit. 2016-05-10]. ISSN 1648-5831 (printed), 2335-8971 (online). Dostupné z:

[http://www.mii.lt/informatics\\_in\\_education/pdf/INFE133.pdf](http://www.mii.lt/informatics_in_education/pdf/INFE133.pdf)

DRBAL, Pavel, 1992. *Co to je strukturované programování*. Praha: Nová vlna. Knihovnička učitele informatiky. ISBN 80-900325-6-7.

DRBAL, Pavel, 1994. Vývoj programování a jeho další cesta aneb Od selského rozumu k objektům. In: *Programování '94*. Ostrava: Dům techniky Ostrava, s. 1-7.

Dostupné také z: <http://prog-story.technicalmuseum.cz/images/dokumenty/Programovani-TSW-1975-2014/1994/1994-03.pdf>

DRBAL, Pavel, 1996. Proudý v objektově orientovaných metodikách. In: *Tvorba softwaru 96: Celostátní seminář*. Ostrava: Tanger, s. 1-10. ISBN 80-85988-09-7.

Dostupné také z: <http://prog-story.technicalmuseum.cz/images/dokumenty/Programovani-TSW-1975-2014/1996/1996-11.pdf>

EHLERT, Albrecht a Carsten SCHULTE, 2007. Learners Views on Objects-First and Objects-Later - Results of an Exploratory Study. CEBULLA, Michael. *Object-Oriented Technology: ECOOP 2007 Workshop Reader*. Berlin, Germany: Springer, p. 184. ISBN 3-540-78194-3. Dostupné také z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.865&rep=rep1&type=pdf>

EHLERT, Albrecht a Carsten SCHULTE, 2009. Empirical Comparison of Objects-First and Objects-Later. *ICER'09: Proceedings of the fifth International Computing Education Research Workshop* [online]. 15-26 [cit. 2015-06-29]. DOI: 10.1145/1584322.1584326. Dostupné z: <https://home.cc.gatech.edu/csed/uploads/2/ehlert2009.pdf>

FOJTÍK, Rostislav, 2013. Moderní přístupy k výuce programování. *JTIE: Journal of Technology and Information Education* [online]. 5(1), 58-62 [cit. 2015-11-26]. ISSN 1803-6805. Dostupné z: <http://jtie.upol.cz/pdfs/jti/2013/01/08.pdf>

GEWALI, Laxmi P a John T MINOR, 2006. Multi-Paradigm Approach for Teaching Programming. In: *Proceedings of the 2006 International Conference on Frontiers in Education: Computer Science & Computer Engineering: FECS 2006*. Las Vegas. ISBN 1-60132-008-6. Dostupné také z: [https://www.researchgate.net/publication/220844071\\_Multi-Paradigm\\_Approach\\_for\\_Teaching\\_Programming](https://www.researchgate.net/publication/220844071_Multi-Paradigm_Approach_for_Teaching_Programming)

HONZÍK, Max Jan, 1990. Principy objektově orientovaného programování. In: *Programování '90*. Ostrava: Dům techniky ČSVTS Ostrava, s. 1-20. Dostupné také z: <http://prog-story.technicalmuseum.cz/images/dokumenty/Programovani-TSW-1975-2014/1990/1990-06.pdf>

HOŘEJŠ, Jiří, 1984. 30 let strukturovaného programování. In: *Programování '84*. Ostrava: Dům techniky ČSVTS Ostrava, s. 1-7. Dostupné také z: <http://prog-story.technicalmuseum.cz/images/dokumenty/Programovani-TSW-1975-2014/1984/1984-01.pdf>



HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK, 2015a. Algorithmic Thinking in Paradigms of Programming. In: *Recent Advances in Educational Technologies, The 2015 International Conference on Education and Modern Educational Technologies (EMET 2015)*, Series: Educational Technologies Series, Zakynthos Island, Greece, July 16-20, pp. 48-53. ISBN 978-1-61804-322-1, ISSN 2227-4618.

HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK, 2015b. Evaluation of Algorithmic Thinking of Students Using Control Testing Environment. *International journal of education and information technologies*. North atlantic university union. 4s. ISSN 2074-1316.

HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK, 2015c. Object Thinking in Paradigms of Programming. In: *Recent Advances in Educational Technologies, The 2015 International Conference on Education and Modern Educational Technologies (EMET 2015)*, Series: Educational Technologies Series, Zakynthos Island, Greece, July 16-20 , pp. 54-58. ISBN 978-1-61804-322-1, ISSN 2227-4618.

CHEN, Kuan C, 2004. COMPARISON OF OBJECT-ORIENTED AND PROCEDURE-BASED COMPUTER LANGUAGES: CASE STUDY OF C++ PROGRAMMING. *Issues in Information Systems* [online]. 5(1), 70-76 [cit. 2016-05-05]. ISSN 1529-7314. Dostupné z: <http://iacis.org/iis/2004/Chen.pdf>

CHRÁSKA, Miroslav, 2003. *Úvod do výzkumu v pedagogice: Základy kvantitativně orientovaného výzkumu*. Olomouc: Univerzita Palackého v Olomouci. ISBN 80-244-0765-5.

*Jak se pozná normalita pomocí grafů?*, 2013 [online]. Praha: Statsoft CR [cit. 2016-08-12]. Dostupné z: [http://www.statsoft.cz/file1/PDF/newsletter/2013\\_10\\_09\\_StatSoft\\_Jak\\_se\\_pozna\\_normalita\\_pomoci\\_grafu.pdf](http://www.statsoft.cz/file1/PDF/newsletter/2013_10_09_StatSoft_Jak_se_pozna_normalita_pomoci_grafu.pdf)

JENSEN, Randall W, 1981. Structured Programming. *Computer*. 14(3), 31 - 48. DOI: 10.1109/C-M.1981.220374. ISSN 0018-9162. Dostupné také z: <https://www.computer.org/csdl/mags/co/1981/03/01667281.pdf>

JOHNSON, Richard A. a Duane R. MOSES, 2008. Objects-First vs. Structures-First Approaches to OO Programming education: An Empirical Study. In: *Proceedings of the International Academy for Case Studies*. **15**(2). Reno: International Academy for Case Studies, p. 244-248. Dostupné také z:

[https://www.researchgate.net/publication/242549890\\_OBJECTS-FIRST\\_VS\\_STRUCTURES-FIRST\\_APPROACHES\\_TO\\_OO\\_PROGRAMMING\\_EDUCATION\\_AN\\_EMPIRICAL\\_STUDY](https://www.researchgate.net/publication/242549890_OBJECTS-FIRST_VS_STRUCTURES-FIRST_APPROACHES_TO_OO_PROGRAMMING_EDUCATION_AN_EMPIRICAL_STUDY)

KERLINGER, Fred N, 1972. *Základy výzkumu chování: pedagogický a psychologický výzkum*. Praha: Academia cit. IN: CHRÁSKA, Miroslav, 2003. *Úvod do výzkumu v pedagogice: Základy kvantitativně orientovaného výzkumu*. Olomouc: Univerzita Palackého v Olomouci. ISBN 80-244-0765-5.

KICZALES, Gregor et al., 1997. Aspect-oriented programming. In: *ECOOP'97 — Object-Oriented Programming* [online]. Finland, s. 220-242 [cit. 2016-10-22]. DOI: 10.1007/BFb0053371. ISBN 978-3-540-69127-3. Dostupné z: <http://www2.parc.com/csl/groups/sda/publications/papers/Kiczales-ECOOP97/for-web.pdf>

KLIMEŠ, Cyril et al., 2008. *Informatika: pro maturanty a zájemce o studium na vysokých školách*. České vydání - aktualizováno a upraveno. Nitra: Enigma. ISBN 978-80-89132-71-3.

KÖLLING, Michael, 1999. The problem of teaching object-oriented programming: Part I: Languages. *Journal of Object-Oriented Programming*. **11**(8), 8-15. Dostupné také z: <https://www.bluej.org/papers/1999-08-JOOP1-languages.pdf>

KONEČNÝ, Jan a Vilém VYCHODIL, 2008. *Paradigmata programování IA* [online]. Olomouc [cit. 2015-11-15]. Dostupné z: <https://phoenix.inf.upol.cz/esf/ucebni/pp1a.pdf>

KRAVAL, Ilja, 1998. *Základy objektově orientovaného programování: za pomoci jazyka Microsoft Visual Basic 5.0*. Praha: Computer Press. ISBN 80-7226-047-2

- KRČEK, Břetislav a Pavel KREML, 1993. *Algoritmizace a programování v jazyku Pascal [Krček, 1993, záznam a]*. 1. vyd. Ostrava: Vysoká škola báňská. ISBN 80-7078-215-3 cit. IN: DOHNAL, Pavel, 2009. *Programování na základních školách*. Brno. Dostupné také z:  
[http://is.muni.cz/th/72886/pedf\\_m/Diplomova\\_prace\\_Pavel\\_Dohnal.pdf](http://is.muni.cz/th/72886/pedf_m/Diplomova_prace_Pavel_Dohnal.pdf). Diplomová práce. Vedoucí práce Ing. Martin Dosedla.
- LISTER, Raymond et al., 2006. Research perspectives on the objects-early debate. In: *ITiCSE-WGR '06 Working Group Reports on ITICSE on Innovation and Technology in Computer Science Education*. New York, 146 - 165. DOI: 10.1145/1189215.1189183. ISBN 1-59593-603-3. Dostupné také z:  
<https://www.cs.auckland.ac.nz/courses/compsci747s2c/lectures/p146-lister.pdf>
- PECINOVSKÝ, Rudolf, 2006. Současné trendy v metodice výuky programování. In: *Počítač ve škole 2006* [online]. Nové Město na Moravě: Gymnázium Vincence Makovského [cit. 2016-07-20]. Dostupné z:  
[http://www.pocitacveskole.cz/system/files/uzivatel/9/clanky/pecinovsky\\_pdf\\_13597.pdf](http://www.pocitacveskole.cz/system/files/uzivatel/9/clanky/pecinovsky_pdf_13597.pdf)
- PECINOVSKÝ, Rudolf, 2007a. *Metodika design patterns first a vyhodnocování studentských úloh*. [online]. [cit. 2015-10-01]. Dostupné z:  
[http://vyuka.pecinovsky.cz/prispevky/2007-SW\\_Metodika\\_DPF\\_a\\_vyhodnocovani\\_uloh.pdf](http://vyuka.pecinovsky.cz/prispevky/2007-SW_Metodika_DPF_a_vyhodnocovani_uloh.pdf)
- PECINOVSKÝ, Rudolf, 2007b. Metodika výuky programování na rozcestí. In: *Poškole 2007*. Praha: MOV Poškole, s. 48-57. ISBN 978-80-239-9126-0. Dostupné také z:  
[http://vyuka.pecinovsky.cz/prispevky/2007\\_Po\\_Metodika\\_vyuky\\_na\\_rozcesti.pdf](http://vyuka.pecinovsky.cz/prispevky/2007_Po_Metodika_vyuky_na_rozcesti.pdf)
- PECINOVSKÝ, Rudolf, 2008. Mýty ve výuce programování a metodika Design Patterns First. In: *Objekty 2008*. Žilina: Žilinská univerzita FRI, s. 17-27. ISBN 978-80-8070-927-3. Dostupné také z: [http://vyuka.pecinovsky.cz/prispevky/2008-OB\\_MytyVeVyuceOOPaDPF.pdf](http://vyuka.pecinovsky.cz/prispevky/2008-OB_MytyVeVyuceOOPaDPF.pdf)

PECINOVSKÝ, Rudolf, 2009. *Myslíme objektivě v jazyku Java: kompletní učebnice pro začátečníky*. 2., aktualiz. a rozš. vyd. Praha: Grada. ISBN 978-80-247-2653-3.

PECINOVSKÝ, Rudolf, 2010a. Quo Vadis programování? In: *Tvorba softwaru 2010: Celostátní konference s mezinárodní účastí*. Ostrava: VŠB-Technická univerzita Ostrava, s. 158-161. ISBN 978-80-248-2225-9. Dostupné také z: <http://prog-story.technicalmuseum.cz/images/dokumenty/Programovani-TSW-1975-2014/2010/2010-22.pdf>

PECINOVSKÝ, Rudolf, 2010b. Metodika Design Patterns First. In: *Počítač ve škole 2010* [online]. Nové město na Moravě: Gymnázium Vincence Makovského [cit. 2015-06-26]. Dostupné z: [http://www.pocitacveskole.cz/system/files/uzivatel/9/clanky/pecinovsky\\_pdf\\_61860.pdf](http://www.pocitacveskole.cz/system/files/uzivatel/9/clanky/pecinovsky_pdf_61860.pdf)

PECINOVSKÝ, Rudolf, 2010c. Metodika Design Patterns First v roce 2010. In: *Objekty 2010*. Ostrava: Ostravská univerzita, s. 207-217. ISBN 978-80-7368-899-8. Dostupné také z: [http://vyuka.pecinovsky.cz/prispevky/2010\\_OB\\_Metodika%20DPF%20v%20roce%202010.pdf](http://vyuka.pecinovsky.cz/prispevky/2010_OB_Metodika%20DPF%20v%20roce%202010.pdf)

PECINOVSKÝ, Rudolf, 2013. Methodology Architecture First. In: *JTIE: Journal of Technology and Information Education* [online]. Svazek 5. Olomouc, s. 107-114 [cit. 2016-07-08]. eISSN 1803-6805. Dostupné z: <http://jtie.upol.cz/pdfs/jti/2013/01/16.pdf>

PECINOVSKÝ, Rudolf, 2015. Metodika Architecture First. In: *DidInfo 2015 : 21. ročník národnej konferencie* [online]. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied v Banskej Bystrici, s. 118-122 [cit. 2016-07-08]. ISBN 978-80-557-0852-2. Dostupné z: <https://zssnpbb.edupage.org/files/Didinfo2015i.pdf>

PELIKÁN, Jiří, 1998. *Základy empirického výzkumu pedagogických jevů*. Praha: Karolinum. ISBN 80-7184-569-8.

PRŮCHA, Jan, Eliška WALTEROVÁ a Jiří MAREŠ, 2003. *Pedagogický slovník*. 4., aktualizované vydání. Praha: Portál. ISBN 80-7178-772-8.

REYNOLDS-HAERTLE, Robin A, 2002. *OOP: objektově orientované programování Visual Basic .NET, Visual C# .NET krok za krokem*. Praha: Mobil Media. ISBN 80-86593-25-8.

RÝDLO, Lukáš, 2012. *Programovací jazyky pro výuku programování na SŠ* [online]. Brno [cit. 2016-05-15]. Dostupné z: [http://is.muni.cz/th/139514/fi\\_m/dp.pdf](http://is.muni.cz/th/139514/fi_m/dp.pdf). Diplomová práce. Vedoucí práce Doc. RNDr. Tomáš Pitner, Ph.D.

SAJANIEMI, Jorma a Chenglie HU, 2006. Teaching Programming: Going beyond “Objects First”. In: *PPIG 2006: 18th Annual Workshop* [online]. University of Sussex, s. 255-265 [cit. 2016-07-14]. Dostupné z: <http://www.ppig.org/sites/default/files/2006-PPIG-18th-sajaniemi.pdf>

SCHILDT, Herbert, 2012. *Java 7: Výukový kurz*. Brno: Computer Press. ISBN 978-80-251-3748-2.

UYSAL, Murat Pasa, 2012. The Effects of Objects-First and Objects-Late Methods on Achievements of OOP Learners. *Journal of Software Engineering and Applications* [online]. 5(10), 816-822 [cit. 2015-10-05]. DOI: 10.4236/jsea.2012.510094. Dostupné z: [http://file.scirp.org/pdf/JSEA20121000008\\_34600702.pdf](http://file.scirp.org/pdf/JSEA20121000008_34600702.pdf)

VOBORNÍK, Petr, 2012a. *Univerzální testovací prostředí* [online]. Hradec Králové [cit. 2016-04-20]. Dostupné z: <http://download.petrvobornik.cz/docs/disertace.pdf>. Disertační práce. Vedoucí práce Prof. RNDr. Eva Milková, Ph.D.

VOBORNÍK, Petr, 2012b. *Univerzální testovací prostředí* [online]. [cit. 2015-10-18]. Dostupné z: <http://www.alltest.eu/>

VUJOŠEVIĆ-JANIČIĆ, Milena a Dušan TOŠIĆ, 2008. The role of programming paradigms in the first programming courses. *The Teaching of Mathematics* [online].

11(2), 63-83 [cit. 2016-04-30]. ISSN 2406-1077. Dostupné z:  
<http://elib.mi.sanu.ac.rs/files/journals/tm/21/tm1122.pdf>

VYSTAVĚL, Radek, 2008. Moderní výuka programování. In: *Tvorba softwaru 2008: Celostátní konference s mezinárodní účastí*. Ostrava: VŠB-TU Ostrava, s. 191-196. ISBN 80-248-1765-1. Dostupné také z: <http://prog-story.technicalmuseum.cz/images/dokumenty/Programovani-TSW-1975-2014/2008/2008-28.pdf>

ZIKMUND, Miloš, 2010. *Aspektově orientované programování a jeho podpora* [online]. Brno [cit. 2016-07-29]. Dostupné z:  
[http://is.muni.cz/th/173275/fi\\_m/dp.pdf](http://is.muni.cz/th/173275/fi_m/dp.pdf). Diplomová práce.

## PUBLIKAČNÍ ČINNOST AUTORA

HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK, 2015a. Algorithmic Thinking in Paradigms of Programming. In: *Recent Advances in Educational Technologies, The 2015 International Conference on Education and Modern Educational Technologies (EMET 2015)*, Series: Educational Technologies Series, Zakynthos Island, Greece, July 16-20, pp. 48-53. ISBN 978-1-61804-322-1, ISSN 2227-4618.

HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK, 2015b. Evaluation of Algorithmic Thinking of Students Using Control Testing Environment. *International journal of education and information technologies*. North atlantic university union, 4s. ISSN 2074-1316.

HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK, 2015c. Object Thinking in Paradigms of Programming. In: *Recent Advances in Educational Technologies, The 2015 International Conference on Education and Modern Educational Technologies (EMET 2015)*, Series: Educational Technologies Series, Zakynthos Island, Greece, July 16-20, 2015, pp. 54-58. ISBN 978-1-61804-322-1, ISSN 2227-4618.

HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK. Algorithmic Thinking in Methods of Learning of Programming Objects-First & Objects-Later. *International journal of education and information technologies*. North atlantic university union, ISSN 2074-1316. zasláno k recenzi.

HUBÁLOVSKÝ, Štěpán a Ondřej KOŘÍNEK. Development of Object Thinking in Paradigms of Programming. *International journal of education and information technologies*. North atlantic university union, ISSN 2074-1316. zasláno k recenzi.

KOŘÍNEK, Ondřej, 2013a. Výzkum přístupů ve výuce programování. In: *PRIT: Pedagogical Research on Information Technology 2013*. České Budějovice: Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta. ISBN 978-80-7394-433-9.

KOŘÍNEK, Ondřej, 2013b. Výzkum přístupů ve výuce programování. In: *DITECH: Digital Technology*. Hradec Králové: Univerzita Hradec Králové. ISBN 978-80-7435-377-2.

KOŘÍNEK, Ondřej, 2014. Srovnání vývojových diagramů a pseudokódu ve výuce algoritmicizace. *Matematika Fyzika Informatika* [online]. Praha: Prometheus, spol. s r. o., **23**(3), 219–225 [cit. 2014-06-09]. ISSN 1210-1761. Dostupné z: [http://mfi.upol.cz/files/23/2303/mfi\\_2303\\_219\\_225.pdf](http://mfi.upol.cz/files/23/2303/mfi_2303_219_225.pdf)

KOŘÍNEK, Ondřej, 2015a. Modelování a simulace v objektově orientovaném programování. *Media4u Magazine* [online]. J. Chromý. 32–36 [cit. 2016-03-09]. ISSN: 1214-9187.

KOŘÍNEK, Ondřej, 2015b. Výzkum přístupů ve výuce programování. In: *ICTE 2015: Information and Communication Technologies in Education 2015: PhD section*. Rožnov pod Radhoštěm.

KOŘÍNEK, Ondřej a Štěpán HUBÁLOVSKÝ, 2016. *Researches of Methods of Learning of Programming Objects First and Object Later*, Springer Verlag for the conference Proceedings, APSAC'16, Dobrovnik.

MILKOVÁ, Eva a Ondřej KOŘÍNEK, 2013. Students' Programming capabilities evaluation. In: *ERIE 2013: Efficiency and Responsibility in Education 2013*, Prague, pp. 434-440. ISSN 1803-1617.

MILKOVÁ, Eva a Ondřej KOŘÍNEK, 2014. Future ICT Teachers - Programming Aptitude, In: *ERIE 2014: Proceedings of the 11th International Conference Efficiency and Responsibility in Education 2014*, Prague, pp. 456-462.



## SEZNAM OBRÁZKŮ

Obrázek 1 – Ukázka testování studentů v akademickém roce 2013/2014 s časovým vymezením jednotlivých částí .....	86
Obrázek 2 – Úloha na doplňování kódu.....	88
Obrázek 3 – Úloha na využití interaktivních prvků.....	89
Obrázek 4 – Doplnění hodnot do proměnných.....	90
Obrázek 5 – Graf normality objektové části (vytvoření instance a volání metod) studentů z posttestu z akademického roku 2013/2014, obrázek byl vytvořen v programu Statistica 12 .....	98

## SEZNAM TABULEK

Tabulka 1 – „ <i>Sekvence témat dosud vyučovaných ve skupině koncepcí výuky programování objects-first a objects-later</i> “ (Ehlert a Schulte, 2009, s. 18), tabulka převzata z Ehlert a Schulte (2009, s. 18) .....	53
Tabulka 2 – „ <i>Sekvence a struktura probíraných témat kurzu pro koncepcí výuky programování objects-first a objects-later</i> “ (Uysal, 2012, s. 818), tabulka převzata z Uysala (2012, s. 818) .....	54
Tabulka 3 – „ <i>Výsledky studijní úspěšnosti podle testu Mann-Whitney</i> “ (Uysal, 2012), tabulka převzata z Uysala (2012, s. 821).....	58
Tabulka 4 – „ <i>Přibližná interpretace hodnot korelačního koeficientu</i> (Chráska, 2003, s. 111)“ Tabulka převzata z (Chráska, 2003, s. 111).....	73
Tabulka 5 – Přehled načasování etap pedagogického experimentu .....	75
Tabulka 6 – Počty studentů ve skupinách v různých akademických letech...	77
Tabulka 7 – Obsahová náplň kontrolní skupiny studentů (objects-later) a experimentální skupiny studentů (objects-first).....	79
Tabulka 8 – Rozdělení studentů do dvou skupin podle výsledku z pretestu..	95
Tabulka 9 – Výsledky studentů z hlavních částí posttestu.....	96
Tabulka 10 – Výsledky studentů z dílčích částí teoretické části posttestu ....	96
Tabulka 11 – Výsledky studentů z dílčích částí objektové části posttestu ....	97
Tabulka 12 – Výsledky studentů z dílčích částí strukturované části posttestu .....	98
Tabulka 13 – Rozdělení studentů, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu .....	100
Tabulka 14 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z hlavních částí .....	101
Tabulka 15 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z dílčích částí teoretické části.....	102
Tabulka 16 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z dílčích částí objektové části.....	102
Tabulka 17 – Výsledky studentů z posttestu (kteří absolvovali i retest 1) z dílčích částí strukturované části.....	103

Tabulka 18 – Rozdělení studentů do dvou skupin podle výsledku z pretestu. .....	105
Tabulka 19 – Výsledky studentů z retestu 1 z hlavních částí.....	105
Tabulka 20 – Výsledky studentů z retestu 1 z dílčích částí teoretické části	106
Tabulka 21 – Výsledky studentů z retestu 1 z dílčích částí objektové části	107
Tabulka 22 – Výsledky studentů z retestu 1 z dílčích částí strukturované části .....	107
Tabulka 23 – Rozdělení studentů do dvou skupin podle výsledku z pretestu .....	109
Tabulka 24 – Výsledky studentů z retestu 2 z hlavních částí.....	109
Tabulka 25 – Výsledky studentů z retestu 2 z dílčích částí teoretické části	110
Tabulka 26 – Výsledky studentů z retestu 2 z dílčích částí objektové části	110
Tabulka 27 – Rozdělení studentů do dvou skupin podle výsledku z pretestu. .....	111
Tabulka 28 – Výsledky studentů experimentální skupiny z hlavních částí posttestu a retestu 1 .....	112
Tabulka 29 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z teoretické části .....	112
Tabulka 30 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z objektové části .....	113
Tabulka 31 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 ze strukturované části .....	113
Tabulka 32 – Rozdělení studentů do dvou skupin podle výsledku z pretestu. .....	114
Tabulka 33 – Výsledky studentů kontrolní skupiny z hlavních částí posttestu a retestu 1.....	115
Tabulka 34 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z teoretické části.....	116
Tabulka 35 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z objektové části.....	116
Tabulka 36 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 ze strukturované části .....	117

Tabulka 37 – Rozdělení studentů do dvou skupin podle výsledku z pretestu v akademickém roce 2014/2015 .....	119
Tabulka 38 – Výsledky studentů z posttestu z hlavních částí v akademickém roce 2014/2015 .....	119
Tabulka 39 – Výsledky studentů z posttestu z dílčích částí teoretické části v akademickém roce 2014/2015 .....	120
Tabulka 40 – Výsledky studentů z posttestu z dílčích částí objektové části v akademickém roce 2014/2015 .....	121
Tabulka 41 – Výsledky studentů z posttestu z dílčích částí strukturované části v akademickém roce 2014/2015 .....	121
Tabulka 42 – Rozdělení studentů do dvou skupin podle výsledku z pretestu, kteří absolvovali i retest 1 .....	123
Tabulka 43 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z hlavních částí .....	124
Tabulka 44 – Výsledky studentů posttestu, kteří absolvovali i retest 1, z dílčích částí teoretické části.....	124
Tabulka 45 – Výsledky studentů posttestu, kteří absolvovali i retest 1, z dílčích částí objektové části.....	125
Tabulka 46 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí strukturované části.....	126
Tabulka 47 – Rozdělení studentů do dvou skupin podle výsledku z pretestu, kteří absolvovali retest 1 .....	127
Tabulka 48 – Výsledky studentů z retestu 1 z hlavních částí.....	128
Tabulka 49 – Výsledky studentů z retestu 1 z dílčích částí teoretické části	129
Tabulka 50 – Výsledky studentů z retestu 1 z dílčích částí objektové části	129
Tabulka 51 – Výsledky studentů z retestu 1 z dílčích částí strukturované části .....	130
Tabulka 52 – Rozdělení studentů do dvou skupin podle výsledku z pretestu, kteří psali retest 2 .....	132
Tabulka 53 – Výsledky studentů z retestu 2 z hlavních částí.....	133
Tabulka 54 – Výsledky studentů z retestu 2 z dílčích částí teoretické části	133
Tabulka 55 – Výsledky studentů z retestu 2 z dílčích částí objektové části	134

Tabulka 56 – Rozdělení studentů, kteří absolvovali posttest i retest 1, do dvou skupin podle výsledku z pretestu, .....	136
Tabulka 57 – Výsledky studentů experimentální skupiny z hlavních částí posttestu a retestu 1 .....	136
Tabulka 58 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z teoretické části .....	137
Tabulka 59 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 z objektové části .....	138
Tabulka 60 – Výsledky studentů experimentální skupiny z dílčích částí posttestu a retestu 1 ze strukturované části .....	138
Tabulka 61 – Rozdělení studentů, kteří absolvovali posttest i retest 1, do dvou skupin podle výsledku z pretestu .....	139
Tabulka 62 – Výsledky studentů kontrolní skupiny z hlavních částí posttestu a retestu 1 .....	140
Tabulka 63 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z teoretické části .....	140
Tabulka 64 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 z objektové části .....	141
Tabulka 65 – Výsledky studentů kontrolní skupiny z dílčích částí posttestu a retestu 1 ze strukturované části .....	141
Tabulka 66 – Rozdělení studentů, bez studentů opakujících předmět Programování 1, do dvou skupin podle výsledku z pretestu .....	144
Tabulka 67 – Výsledky studentů z posttestu z hlavních částí v akademickém roce 2014/2015 .....	145
Tabulka 68 – Výsledky studentů z posttestu z dílčích částí teoretické části v akademickém roce 2014/2015 .....	145
Tabulka 69 – Výsledky studentů z posttestu z dílčích částí objektové části v akademickém roce 2014/2015 .....	146
Tabulka 70 – Výsledky studentů z posttestu z dílčích částí strukturované části v akademickém roce 2014/2015 .....	147
Tabulka 71 – Rozdělení studentů, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu .....	149

Tabulka 72 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z hlavních částí .....	149
Tabulka 73 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí teoretické části.....	150
Tabulka 74 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí objektové části.....	151
Tabulka 75 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí strukturované části.....	151
Tabulka 76 – Rozdělení studentů, kteří absolvovali retest 1, do dvou skupin podle výsledku z pretestu .....	154
Tabulka 77 – Výsledky studentů z retestu 1 z hlavních částí.....	154
Tabulka 78 – Výsledky studentů z retestu 1 z dílčích částí teoretické části	155
Tabulka 79 – Výsledky studentů z retestu 1 z dílčích částí objektové části	156
Tabulka 80 – Výsledky studentů z retestu 1 z dílčích částí strukturované části .....	156
Tabulka 81 – Rozdělení studentů v experimentální skupině.....	158
Tabulka 82 – Výsledky studentů experimentální skupiny z hlavních částí posttestu a z retestu 1 .....	159
Tabulka 83 – Výsledky studentů experimentální skupiny z dílčích částí teoretické části z posttestu a z retestu 1 .....	160
Tabulka 84 – Výsledky studentů experimentální skupiny z dílčích částí objektové části z posttestu a z retestu 1 .....	160
Tabulka 85 – Výsledky studentů experimentální skupiny z dílčích částí strukturované části z posttestu a z retestu .....	161
Tabulka 86 – Rozdělení studentů v kontrolní skupině.....	162
Tabulka 87 – Výsledky studentů kontrolní skupiny z hlavních částí z posttestu a z retestu 1 .....	162
Tabulka 88 – Výsledky studentů kontrolní skupiny z dílčích částí teoretické části z posttestu a z retestu 1.....	163
Tabulka 89 – Výsledky studentů kontrolní skupiny z dílčích částí objektové části z posttestu a z retestu 1.....	164

Tabulka 90 – Výsledky studentů kontrolní skupiny z dílčích částí strukturované části z posttestu a z retestu 1 .....	164
Tabulka 91 – Rozdělení studentů do dvou skupin podle výsledku z pretestu u vyučujícího Voborníka.....	166
Tabulka 92 – Rozdělení studentů do dvou skupin podle výsledku z pretestu v akademickém roce 2014/2015 .....	167
Tabulka 93 – Rozdělení studentů, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu .....	168
Tabulka 94 – Rozdělení studentů, kteří absolvovali retest 1, do dvou skupin podle výsledku z pretestu .....	170
Tabulka 95 – Rozdělení studentů do dvou skupin podle výsledku z pretestu v akademickém roce 2014/2015 .....	171
Tabulka 96 – Rozdělení studentů z posttestu, kteří absolvovali i retest 1, do dvou skupin podle výsledku z pretestu .....	173
Tabulka 97 – Rozdělení studentů, kteří absolvovali retest 1, do dvou skupin podle výsledku z pretestu .....	174
Tabulka 98 – Nezamítnutí/odmítnutí nulové hypotézy v testech .....	177
Tabulka 99 – Nezamítnutí/odmítnutí nulové hypotézy v testech .....	179
Tabulka 100 – Nezamítnutí/odmítnutí nulové hypotézy v testech .....	183
Tabulka 101 – Nezamítnutí/odmítnutí nulové hypotézy v testech .....	186
Tabulka 102 – Nezamítnutí/odmítnutí nulové hypotézy v testech .....	189

## **PŘÍLOHY**

Příloha A: Dotazníkové šetření.....	I
Příloha B: Ukázka možného zadání posttestu .....	III
Příloha C: Ukázka možného zadání retestu 1 .....	IV
Příloha D: Ukázka možného zadání retestu 2.....	V
Příloha E: Spearmanův koeficient pořadové korelace .....	VI
Příloha F: Vyhodnocení dotazníkového šetření.....	XXIV
Příloha G: Vyhodnocení výsledků studentů jedním vyučujícím .....	XXVIII



## Příloha A: Dotazníkové šetření

Ke každé otázce zatrhněte pouze jednu možnost podle následující stupnice

- 1 - určitě ano     2 - spíše ano     3 - nevím     4 - spíše ne  
 5 - určitě ne

1. Vycházel Vám vyučující vstříc?  
 1     2     3     4     5
2. Snažil se Vám vyučující pomoci, když jste něco nevěděli?  
 1     2     3     4     5
3. Vysvětlil Vám vyučující látku v případě, kdy jste měli pochybnosti?  
 1     2     3     4     5
4. Snažil se Vás vyučující pozitivně motivovat při hodinách?  
 1     2     3     4     5
5. Snažil se Vám vyučující látku vysvětlit i vícekrát?  
 1     2     3     4     5
6. Zaujal Vás výklad látky při hodinách?  
 1     2     3     4     5
7. Pochopili jste danou látku od vyučujícího?  
 1     2     3     4     5
8. Snažil se vyučující hodiny odlehčit?  
 1     2     3     4     5
9. Komunikoval s Vámi vyučující?  
 1     2     3     4     5
10. Přistupoval k Vám vyučující individuálně?  
 1     2     3     4     5
11. Snažil se Vás vyučující podporovat?  
 1     2     3     4     5
12. Snažil se Vám vyučující dávat rady ke způsobu programování?  
 1     2     3     4     5
13. Nechával Vám vyučující na naprogramování úloh dostatek času?  
 1     2     3     4     5

14. Opakoval vyučující na začátku hodiny látku z minulého týdne?  
 1  2  3  4  5
15. Dával Vám vyučující rady, jak by se mělo nejlépe učit programování?  
 1  2  3  4  5
16. Hodnotíte materiály na kurzech jako dostačující?  
 1  2  3  4  5
17. Pomáhaly Vám materiály při učení se?  
 1  2  3  4  5
18. Stačily Vám příklady na procvičování?  
 1  2  3  4  5
19. Když jste něco nepochopili ve výuce, našli jste vysvětlení v materiálech?  
 1  2  3  4  5
20. Byly pro Vás probírané příklady těžké?  
 1  2  3  4  5
21. Hledali jste i jiné materiály na internetu?  
 1  2  3  4  5
22. Programovalo se Vám ve Visual Studiu příjemně?  
 1  2  3  4  5
23. Zdálo se Vám obtížné programování pomocí OOP?  
 1  2  3  4  5
24. Zdálo se Vám obtížné programování algoritmů bez OOP?  
 1  2  3  4  5

## Příloha B: Ukázka možného zadání posttestu

Ve Visual Studiu založte nový projekt konzolové aplikace C# a v něm zpracujte následující zadání. Vytvořte třídu **Posloupnost** pro obsluhu posloupností (jednorozměrného pole) s následujícími složkami:

### Základní algoritmy

- **Konstruktor** - vytvoří soukromou datovou položku typu jednorozměrné pole celých čísel dané velikosti, se kterou se bude nadále pracovat
- **N** - vlastnost pouze pro čtení určující délku posloupnosti
- **NaplnNahodne** - metoda, která naplní posloupnost náhodnými čísly v rozsahu zadaném vstupními parametry, jež bude možné zadat následujícími dvěma způsoby (metoda bude přetížena):
  - *minimum* a *maximum* - náhodné hodnoty budou generovány mezi minimem a maximem
  - pouze *maximum* - náhodné hodnoty budou generovány mezi nulou a maximem
- **VypisRadek** - vypíše posloupnost do řádku konzole (hodnoty členů odděleny čárkou)
- **Clen** - vrátí hodnotu člena na pozici zadané jako vstupní parametr

### Výpočty a hledání

- **Minimum** - najde a vrátí minimální hodnotu v posloupnosti

### Posuny hodnot

- **CyklickyPosunVzad** - cyklicky posune vzad (doprava) členy posloupnosti

### Vkládání/odebírání hodnot

- **Odeber** - odebere z posloupnosti člen na pozici určené vstupním parametrem

### Práce s více poli

- **SkalarniSoucin** - vypočte a vrátí hodnotu skalárního součinu této posloupnosti s druhou posloupností stejné délky (nutno ověřit) zadanou jako vstupní parametr

V hlavní části programu (metoda Main třídy Program) vytvořte instanci třídy Posloupnost a vhodně použijte všechny její implementované metody. **Složku celého projektu zabalte do archivu a nahrajte (odevzdejte) sem.**

## Příloha C: Ukázka možného zadání retestu 1

Ve Visual Studiu založte nový projekt konzolové aplikace C# a v něm zpracujte následující zadání. Vytvořte třídu **Matic** pro obsluhu matice (dvourozměrného pole) s následujícími složkami (metodami). Zadání pro danou metodu zkopírujte do komentáře nad její záhlaví. **Základní algoritmy:**

- **Konstruktor** - vytvoří soukromou datovou položku typu dvourozměrné pole celých čísel daných rozměrů (rozměry matice budou určeny vstupními parametry konstruktora), se kterou se bude nadále pracovat
- **N** a **M** - vlastnosti pouze pro čtení, určující rozměry matice (počet řádků a počet sloupců)
- **Vypis** - vypíše matici ve vhodném formátu do okna konzole
- **Clen** - vrátí hodnotu členu na souřadnicích určených vstupními parametry
- **NastavClen** - nastaví hodnotu členu na souřadnicích určených vstupními parametry tímž způsobem na zadanou hodnotu

### Naplnění matice

- **NaplňRadouNasobku** - naplní matici po řádcích jdoucími hodnotami řady definované takto:  $a_0 = a_1 = 1$ ,  $a_i = 4 \cdot a_{i-1} - 2 \cdot a_{i-2}$

### Výpočty a hledání

- **Maximum2** - najde a vrátí druhou největší hodnotu v matici

### Posuny hodnot

- **CyklickyPosunVzad** - ve všech řádcích matice cyklicky posune vzad (doprava) její členy

### Čtvercové matice

- **ProhodTrojuhleniky** - ověří, je-li matice čtvercová a pokud ano, tak hodnoty členů v horním trojúhelníku prohodí s hodnotami v dolním trojúhelníku (trojúhelníky ohraničují hlavní a vedlejší diagonála, ovšem hodnoty na nich ležící do trojúhelníku nespádají)

### Práce s více maticemi

- **SoucetClenuMatic** - sečte členy této matice se členy druhé matice stejných rozměrů (nutno ověřit) zadanou jako vstupní parametr typu *Matic*, do nové třetí matice, kterou vrátí jako výstupní hodnotu opět typu *Matic*

V hlavní části programu (metoda **Main** třídy Program) vytvořte instanci třídy **Matic** a vhodně použijte všechny její implementované metody.

## Příloha D: Ukázka možného zadání retestu 2

Ve Visual Studiu založte nový projekt konzolové aplikace C# a v něm zpracujte následující zadání. Vytvořte třídu **Soubory** pro obsluhu souboru s následujícími složkami (metodami). Zadání pro danou metodu zkopírujte do komentáře nad její záhlaví.

### Slučovač souborů

- **Rozdělení** – metoda, která v zadané složce rozdělí všechny soubory z jednoho binárního souboru. První parametr bude "-r", dalšími dvěma parametry budou pouze jeden soubor a jedna složka. Z tohoto souboru program extrahuje data dříve sloučených souborů a ty znovu obnoví (vytvoří) do složky zadané jako druhý vstupní parametr aplikace. V průběhu se vypisuje text: soubor se zpracovává, soubor je zpracován.

### Parsování

- **Načtení II** - metoda, která načte data ze 4 daných CSV souborů. Soubory obsahují název automobilky, dále pořadové číslo auta, název auta, typ, SPZ a cenu. Spočítejte maximální cenu jednotlivých aut z automobilek a daná data uložte vzestupně podle daných maximálních cen ve formátu automobilka, číslo auta, cena a název.

### Řazení

- **Řazení II** metoda, která z textového souboru rozparsuje pouze průměr čísel na lichých pozicích v jednotlivých řádcích a tyto průměry запиše do binárního souboru.

### Vyhledávání

- **Hledání I** – metoda, která vyhledá soubory s příponou docx, jejichž název obsahuje slovo program. U daných vyhledaných souborů zjistěte datum vytvoření a velikost souboru. Dané soubory uložte podle velikosti sestupně do CSV souboru, kde bude jejich seznam ve formátu jméno souboru; datum vytvoření; velikost.

V hlavní části programu (metoda **Main** třídy Program) vytvořte instanci třídy **Soubory** a vhodně použijte všechny její implementované metody

**Složku celého projektu zabalte do archivu a v čas nahrajte (odevzdejte) sem.**

## Příloha E: Spearmanův koeficient pořadové korelace

Tabulka 103 – Spearmanův koeficient pořadové korelace výsledků studentů z pretestu a z jednotlivých částí posttestu z akademického roku 2013/2014

Část testu	Koeficient korelace
Celkové výsledky	-0,676
Praktická část	-0,549
Teoretická část	-0,707
Objektová část	-0,547
Strukturovaná část	-0,491
Doplňování kódu	-0,469
Interaktivní prvky	-0,552
Doplnění hodnot proměnných	-0,632
Definice konstruktoru a datových položek	-0,519
Metody a vlastnosti	-0,552
Instance a volání metod	-0,547
Naplnění a výpis posloupnosti	-0,469
Návrh algoritmů metod	-0,543

Tabulka 104 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny z pretestu a z jednotlivých částí posttestu z akademického roku 2013/2014

Část testu	Koeficient korelace
Celkové výsledky	-0,784
Praktická část	-0,787
Teoretická část	-0,728
Objektová část	-0,718
Strukturovaná část	-0,802
Doplňování kódu	-0,312
Interaktivní prvky	-0,511
Doplnění hodnot proměnných	-0,481
Definice konstruktoru a datových položek	-0,486
Metody a vlastnosti	-0,748
Instance a volání metod	-0,571
Naplnění a výpis posloupnosti	-0,724
Návrh algoritmů metod	-0,821

**Tabulka 105 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny z pretestu a z jednotlivých částí posttestu z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,606
Praktická část	-0,360
Teoretická část	-0,686
Objektová část	-0,425
Strukturovaná část	-0,176
Doplňování kódu	-0,534
Interaktivní prvky	-0,613
Doplnění hodnot proměnných	-0,754
Definice konstruktoru a datových položek	-0,581
Metody a vlastnosti	-0,394
Instance a volání metod	-0,458
Naplnění a výpis posloupnosti	-0,176
Návrh algoritmů metod	-0,172

**Tabulka 106 – Spearmanův koeficient pořadové korelace výsledků studentů (kteří absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,595
Praktická část	-0,508
Teoretická část	-0,660
Objektová část	-0,551
Strukturovaná část	-0,440
Doplňování kódu	-0,424
Interaktivní prvky	-0,510
Doplnění hodnot proměnných	-0,546
Definice konstruktoru a datových položek	-0,513
Metody a vlastnosti	-0,519
Instance a volání metod	-0,537
Naplnění a výpis posloupnosti	-0,468
Návrh algoritmů metod	-0,519

**Tabulka 107 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny (kteří absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,662
Praktická část	-0,664
Teoretická část	-0,742
Objektová část	-0,700
Strukturovaná část	-0,778
Doplňování kódu	-0,143
Interaktivní prvky	-0,376
Doplnění hodnot proměnných	-0,358
Definice konstruktoru a datových položek	-0,337
Metody a vlastnosti	-0,700
Instance a volání metod	-0,532
Naplnění a výpis posloupnosti	-0,788
Návrh algoritmů metod	-0,783

**Tabulka 108 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny (kteří absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,537
Praktická část	-0,380
Teoretická část	-0,589
Objektová část	-0,534
Strukturovaná část	-0,118
Doplňování kódu	-0,441
Interaktivní prvky	-0,615
Doplnění hodnot proměnných	-0,639
Definice konstruktoru a datových položek	-0,721
Metody a vlastnosti	-0,4983
Instance a volání metod	-0,504
Naplnění a výpis posloupnosti	-0,118
Návrh algoritmů metod	-0,131



**Tabulka 109 – Spearmanův koeficient pořadové korelace výsledků studentů z pretestu a z jednotlivých částí retestu 1 z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,613
Praktická část	-0,474
Teoretická část	-0,670
Objektová část	-0,463
Strukturovaná část	-0,446
Doplňování kódu	-0,548
Interaktivní prvky	-0,358
Doplnění hodnot proměnných	-0,612
Definice konstruktoru a datových položek	-0,256
Metody a vlastnosti	-0,429
Instance a volání metod	-0,433
Naplnění a výpis posloupnosti	-0,334
Návrh algoritmů metod	-0,491

**Tabulka 110 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny z pretestu a z jednotlivých částí retestu 1 z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,493
Praktická část	-0,355
Teoretická část	-0,713
Objektová část	-0,355
Strukturovaná část	-0,355
Doplňování kódu	-0,740
Interaktivní prvky	-0,078
Doplnění hodnot proměnných	-0,553
Definice konstruktoru a datových položek	-0,194
Metody a vlastnosti	-0,368
Instance a volání metod	-0,267
Naplnění a výpis posloupnosti	-0,266
Návrh algoritmů metod	-0,314

**Tabulka 111 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny z pretestu a z jednotlivých částí retestu 1 z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,702
Praktická část	-0,624
Teoretická část	-0,624
Objektová část	-0,624
Strukturovaná část	-0,667
Doplňování kódu	-0,355
Interaktivní prvky	-0,563
Doplnění hodnot proměnných	-0,724
Definice konstruktoru a datových položek	-0,332
Metody a vlastnosti	-0,497
Instance a volání metod	-0,507
Naplnění a výpis posloupnosti	-0,386
Návrh algoritmů metod	-0,731

**Tabulka 112 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny z pretestu a z jednotlivých částí posttestu a retestu 1 z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,630
Praktická část	-0,524
Teoretická část	-0,721
Objektová část	-0,498
Strukturovaná část	-0,580
Doplňování kódu	-0,485
Interaktivní prvky	-0,236
Doplnění hodnot proměnných	-0,473
Definice konstruktoru a datových položek	-0,278
Metody a vlastnosti	-0,486
Instance a volání metod	-0,393
Naplnění a výpis posloupnosti	-0,477
Návrh algoritmů metod	-0,601

**Tabulka 113 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny z pretestu a z jednotlivých částí posttestu a retestu 1 z akademického roku 2013/2014**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,573
Praktická část	-0,507
Teoretická část	-0,314
Objektová část	-0,470
Strukturovaná část	-0,595
Doplňování kódu	-0,411
Interaktivní prvky	-0,528
Doplnění hodnot proměnných	-0,641
Definice konstruktoru a datových položek	-0,466
Metody a vlastnosti	-0,412
Instance a volání metod	-0,471
Naplnění a výpis posloupnosti	-0,281
Návrh algoritmů metod	-0,402

**Tabulka 114 – Spearmanův koeficient pořadové korelace výsledků studentů z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,668
Praktická část	-0,666
Teoretická část	-0,559
Objektová část	-0,637
Strukturovaná část	-0,605
Doplňování kódu	-0,291
Interaktivní prvky	-0,539
Doplnění hodnot proměnných	-0,638
Definice konstruktoru a datových položek	-0,583
Metody a vlastnosti	-0,716
Instance a volání metod	-0,600
Naplnění a výpis posloupnosti	-0,565
Návrh algoritmů metod	-0,572

**Tabulka 115 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,640
Praktická část	-0,571
Teoretická část	-0,697
Objektová část	-0,597
Strukturovaná část	-0,623
Doplňování kódu	-0,114
Interaktivní prvky	-0,515
Doplnění hodnot proměnných	-0,654
Definice konstruktoru a datových položek	-0,564
Metody a vlastnosti	-0,697
Instance a volání metod	-0,497
Naplnění a výpis posloupnosti	-0,699
Návrh algoritmů metod	-0,697

**Tabulka 116 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,811
Praktická část	-0,806
Teoretická část	-0,572
Objektová část	-0,761
Strukturovaná část	-0,753
Doplňování kódu	-0,495
Interaktivní prvky	-0,706
Doplnění hodnot proměnných	-0,657
Definice konstruktoru a datových položek	-0,648
Metody a vlastnosti	-0,791
Instance a volání metod	-0,681
Naplnění a výpis posloupnosti	-0,563
Návrh algoritmů metod	-0,565

**Tabulka 117 – Spearmanův koeficient pořadové korelace výsledků studentů (kteří absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,545
Praktická část	-0,617
Teoretická část	-0,450
Objektová část	-0,558
Strukturovaná část	-0,480
Doplňování kódu	-0,146
Interaktivní prvky	-0,430
Doplnění hodnot proměnných	-0,548
Definice konstruktoru a datových položek	-0,484
Metody a vlastnosti	-0,646
Instance a volání metod	-0,554
Naplnění a výpis posloupnosti	-0,441
Návrh algoritmů metod	-0,475

**Tabulka 118 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny (kteří absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,481
Praktická část	-0,481
Teoretická část	-0,638
Objektová část	-0,541
Strukturovaná část	-0,371
Doplňování kódu	0,181
Interaktivní prvky	-0,261
Doplnění hodnot proměnných	-0,488
Definice konstruktoru a datových položek	-0,409
Metody a vlastnosti	-0,638
Instance a volání metod	-0,301
Naplnění a výpis posloupnosti	-0,638
Návrh algoritmů metod	-0,640

**Tabulka 119 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny (kteří absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,776
Praktická část	-0,778
Teoretická část	-0,505
Objektová část	-0,714
Strukturovaná část	-0,706
Doplňování kódu	-0,398
Interaktivní prvky	-0,662
Doplnění hodnot proměnných	-0,604
Definice konstruktoru a datových položek	-0,620
Metody a vlastnosti	-0,769
Instance a volání metod	-0,681
Naplnění a výpis posloupnosti	-0,511
Návrh algoritmů metod	-0,517

**Tabulka 120 – Spearmanův koeficient pořadové korelace výsledků studentů z pretestu a z jednotlivých částí retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,522
Praktická část	-0,415
Teoretická část	-0,524
Objektová část	-0,404
Strukturovaná část	-0,442
Doplňování kódu	-0,274
Interaktivní prvky	-0,055
Doplnění hodnot proměnných	-0,610
Definice konstruktoru a datových položek	-0,336
Metody a vlastnosti	-0,450
Instance a volání metod	-0,272
Naplnění a výpis posloupnosti	-0,328
Návrh algoritmů metod	-0,412

**Tabulka 121 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny z pretestu a z jednotlivých částí retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,351
Praktická část	0,075
Teoretická část	-0,777
Objektová část	0,075
Strukturovaná část	-0,083
Doplňování kódu	-0,687
Interaktivní prvky	0,215
Doplnění hodnot proměnných	-0,803
Definice konstruktoru a datových položek	0,413
Metody a vlastnosti	-0,111
Instance a volání metod	0,227
Naplnění a výpis posloupnosti	-0,223
Návrh algoritmů metod	0,086

**Tabulka 122 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny z pretestu a z jednotlivých částí retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,776
Praktická část	-0,714
Teoretická část	-0,706
Objektová část	-0,800
Strukturovaná část	-0,505
Doplňování kódu	-0,243
Interaktivní prvky	-0,336
Doplnění hodnot proměnných	-0,513
Definice konstruktoru a datových položek	-0,758
Metody a vlastnosti	-0,627
Instance a volání metod	-0,591
Naplnění a výpis posloupnosti	-0,407
Návrh algoritmů metod	-0,689

**Tabulka 123 – Spearmanův koeficient pořadové korelace výsledků všech studentů z pretestu a z jednotlivých částí retestu 2 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,323
Praktická část	-0,221
Teoretická část	-0,433
Objektová část	-0,266
Strukturovaná část	-0,178
Doplňování kódu	-0,344
Interaktivní prvky (OOP)	-0,137
Doplnění hodnot proměnných	-0,401
Metody a vlastnosti	-0,283
Instance a volání metod	-0,227

**Tabulka 124 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny z pretestu a z jednotlivých částí retestu 2 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,244
Praktická část	-0,079
Teoretická část	0,047
Objektová část	-0,070
Strukturovaná část	-0,302
Doplňování kódu	-0,188
Interaktivní prvky (OOP)	-0,226
Doplnění hodnot proměnných	-0,296
Metody a vlastnosti	-0,137
Instance a volání metod	-0,021

**Tabulka 125 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny z pretestu a z jednotlivých částí retestu 2 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,525
Praktická část	-0,397
Teoretická část	-0,679
Objektová část	-0,443



Strukturovaná část	-0,397
Doplňování kódu	-0,632
Interaktivní prvky (OOP)	-0,087
Doplnění hodnot proměnných	-0,733
Metody a vlastnosti	-0,503
Instance a volání metod	-0,330

**Tabulka 126 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny z pretestu a z jednotlivých částí posttestu a retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,355
Praktická část	-0,203
Teoretická část	-0,516
Objektová část	-0,189
Strukturovaná část	-0,306
Doplňování kódu	-0,264
Interaktivní prvky	-0,077
Doplnění hodnot proměnných	-0,600
Definice konstruktoru a datových položek	0,031
Metody a vlastnosti	-0,341
Instance a volání metod	-0,050
Naplnění a výpis posloupnosti	-0,349
Návrh algoritmů metod	-0,214

**Tabulka 127 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny z pretestu a z jednotlivých částí posttestu a retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,706
Praktická část	-0,686
Teoretická část	-0,606
Objektová část	-0,693
Strukturovaná část	-0,603
Doplňování kódu	-0,316
Interaktivní prvky	-0,489
Doplnění hodnot proměnných	-0,572

Definice konstruktoru a datových položek	-0,650
Metody a vlastnosti	-0,685
Instance a volání metod	-0,591
Naplnění a výpis posloupnosti	-0,484
Návrh algoritmů metod	-0,617

**Tabulka 128 – Spearmanův koeficient pořadové korelace výsledků všech studentů (kteří neopakovali předmět) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,665
Praktická část	-0,582
Teoretická část	-0,634
Objektová část	-0,591
Strukturovaná část	-0,547
Doplňování kódu	-0,306
Interaktivní prvky	-0,500
Doplnění hodnot proměnných	-0,730
Definice konstruktoru a datových položek	-0,482
Metody a vlastnosti	-0,630
Instance a volání metod	-0,602
Naplnění a výpis posloupnosti	-0,505
Návrh algoritmů metod	-0,575

**Tabulka 129 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny (kteří neopakovali předmět) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,686
Praktická část	-0,564
Teoretická část	-0,696
Objektová část	-0,564
Strukturovaná část	-0,644
Doplňování kódu	-0,455
Interaktivní prvky	-0,659
Doplnění hodnot proměnných	-0,647

Definice konstruktoru a datových položek	-0,499
Metody a vlastnosti	-0,643
Instance a volání metod	-0,568
Naplnění a výpis posloupnosti	-0,646
Návrh algoritmů metod	-0,644

**Tabulka 130 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny (kteří neopakovali předmět) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,780
Praktická část	-0,652
Teoretická část	-0,799
Objektová část	-0,690
Strukturovaná část	-0,573
Doplňování kódu	-0,356
Interaktivní prvky	-0,608
Doplnění hodnot proměnných	-0,844
Definice konstruktoru a datových položek	-0,556
Metody a vlastnosti	-0,741
Instance a volání metod	-0,648
Naplnění a výpis posloupnosti	-0,410
Návrh algoritmů metod	-0,638

**Tabulka 131 – Spearmanův koeficient pořadové korelace výsledků všech studentů (kteří neopakovali předmět a absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,534
Praktická část	-0,492
Teoretická část	-0,471
Objektová část	-0,524
Strukturovaná část	-0,416
Doplňování kódu	-0,034
Interaktivní prvky	-0,389
Doplnění hodnot proměnných	-0,663

Definice konstruktoru a datových položek	-0,349
Metody a vlastnosti	-0,511
Instance a volání metod	-0,619
Naplnění a výpis posloupnosti	-0,343
Návrh algoritmů metod	-0,472

**Tabulka 132 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny (kteří neopakovali předmět a absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,663
Praktická část	-0,586
Teoretická část	-0,574
Objektová část	-0,586
Strukturovaná část	-0,586
Doplňování kódu	-0,219
Interaktivní prvky	-0,633
Doplnění hodnot proměnných	-0,529
Definice konstruktoru a datových položek	-0,366
Metody a vlastnosti	-0,586
Instance a volání metod	-0,589
Naplnění a výpis posloupnosti	-0,586
Návrh algoritmů metod	-0,589

**Tabulka 133 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny (kteří neopakovali předmět a absolvovali i retest 1) z pretestu a z jednotlivých částí posttestu z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,709
Praktická část	-0,538
Teoretická část	-0,774
Objektová část	-0,597
Strukturovaná část	-0,473
Doplňování kódu	-0,141
Interaktivní prvky	-0,499
Doplnění hodnot proměnných	-0,801

Definice konstruktoru a datových položek	-0,479
Metody a vlastnosti	-0,664
Instance a volání metod	-0,642
Naplnění a výpis posloupnosti	-0,296
Návrh algoritmů metod	-0,585

**Tabulka 134 – Spearmanův koeficient pořadové korelace výsledků všech studentů (kteří neopakovali předmět) z pretestu a z jednotlivých částí retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,425
Praktická část	-0,342
Teoretická část	-0,401
Objektová část	-0,368
Strukturovaná část	-0,368
Doplňování kódu	-0,262
Interaktivní prvky	0,061
Doplnění hodnot proměnných	-0,543
Definice konstruktoru a datových položek	-0,252
Metody a vlastnosti	-0,452
Instance a volání metod	-0,249
Naplnění a výpis posloupnosti	-0,190
Návrh algoritmů metod	-0,355

**Tabulka 135 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny (kteří neopakovali předmět) z pretestu a z jednotlivých částí retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,644
Praktická část	-0,426
Teoretická část	-0,772
Objektová část	-0,426
Strukturovaná část	-0,626
Doplňování kódu	-0,772
Interaktivní prvky	0,139
Doplnění hodnot proměnných	-0,626

Definice konstruktoru a datových položek	0,096
Metody a vlastnosti	-0,626
Instance a volání metod	-0,184
Naplnění a výpis posloupnosti	-0,629
Návrh algoritmů metod	-0,342

**Tabulka 136 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny (kteří neopakovali předmět) z pretestu a z jednotlivých částí retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,508
Praktická část	-0,433
Teoretická část	-0,362
Objektová část	-0,433
Strukturovaná část	-0,407
Doplňování kódu	0,015
Interaktivní prvky	-0,186
Doplnění hodnot proměnných	-0,429
Definice konstruktoru a datových položek	-0,680
Metody a vlastnosti	-0,484
Instance a volání metod	-0,462
Naplnění a výpis posloupnosti	-0,065
Návrh algoritmů metod	-0,568

**Tabulka 137 – Spearmanův koeficient pořadové korelace výsledků studentů experimentální skupiny (kteří neopakovali předmět) z pretestu a z jednotlivých částí posttestu a retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,579
Praktická část	-0,434
Teoretická část	-0,589
Objektová část	-0,457
Strukturovaná část	-0,525
Doplňování kódu	-0,496
Interaktivní prvky	-0,222
Doplnění hodnot proměnných	-0,517

Definice konstruktoru a datových položek	-0,102
Metody a vlastnosti	-0,571
Instance a volání metod	-0,387
Naplnění a výpis posloupnosti	-0,538
Návrh algoritmů metod	-0,417

**Tabulka 138 – Spearmanův koeficient pořadové korelace výsledků studentů kontrolní skupiny (kteří neopakovali předmět) z pretestu a z jednotlivých částí posttestu a retestu 1 z akademického roku 2014/2015**

<b>Část testu</b>	<b>Koeficient korelace</b>
Celkové výsledky	-0,557
Praktická část	-0,477
Teoretická část	-0,520
Objektová část	-0,531
Strukturovaná část	-0,450
Doplňování kódu	-0,090
Interaktivní prvky	-0,333
Doplnění hodnot proměnných	-0,622
Definice konstruktoru a datových položek	-0,554
Metody a vlastnosti	-0,558
Instance a volání metod	-0,489
Naplnění a výpis posloupnosti	-0,203
Návrh algoritmů metod	-0,551

## Příloha F: Vyhodnocení dotazníkového šetření

Tabulka 139 – Vyhodnocení dotazníkového šetření studentů z akademického roku 2013/2014

Číslo otázky	Vyučující	Četnost odpovědí					Hodnota p
		1	2	3	4	5	
1	Kořínek	8	2	0	0	0	0,564
	Voborník	5	3	0	0	0	
2	Kořínek	5	4	1	0	0	0,424
	Voborník	2	5	1	0	0	
3	Kořínek	3	5	1	1	0	0,625
	Voborník	1	5	2	0	0	
4	Kořínek	3	4	1	2	0	0,564
	Voborník	1	4	1	2	0	
5	Kořínek	4	2	3	0	1	0,965
	Voborník	2	4	2	0	0	
6	Kořínek	2	2	3	2	1	0,564
	Voborník	1	1	2	4	0	
7	Kořínek	1	5	0	2	2	0,625
	Voborník	0	2	3	3	0	
8	Kořínek	5	3	2	0	0	0,248
	Voborník	2	3	1	1	1	
9	Kořínek	5	5	0	0	0	0,286
	Voborník	2	5	0	1	0	
10	Kořínek	2	4	3	1	0	0,213
	Voborník	1	2	1	4	0	
11	Kořínek	2	5	1	2	0	0,929
	Voborník	1	4	3	0	0	
12	Kořínek	2	4	1	2	1	0,534
	Voborník	2	4	1	1	0	
13	Kořínek	3	6	1	0	0	0,929
	Voborník	3	4	0	1	0	
14	Kořínek	1	3	4	1	1	0,657
	Voborník	1	3	3	1	0	



15	Kořínek	1	4	0	5	0	0,756
	Voborník	0	3	2	2	1	
16	Kořínek	2	2	2	3	1	0,424
	Voborník	1	4	2	1	0	
17	Kořínek	2	4	2	1	1	0,965
	Voborník	2	1	5	0	0	
18	Kořínek	4	4	0	1	1	0,505
	Voborník	2	2	4	0	0	
19	Kořínek	1	3	4	1	1	0,722
	Voborník	0	5	1	2	0	
20	Kořínek	4	3	1	2	0	0,859
	Voborník	2	4	1	1	0	
21	Kořínek	6	1	2	1	0	0,722
	Voborník	3	4	0	1	0	
22	Kořínek	2	4	2	2	0	0,505
	Voborník	2	4	2	0	0	
23	Kořínek	3	2	1	3	1	0,798
	Voborník	2	1	4	1	0	
24	Kořínek	1	2	2	1	4	0,168
	Voborník	1	3	3	1	0	

**Tabulka 140 – Vyhodnocení dotazníkového šetření studentů z akademického roku 2014/2015**

Číslo otázky	Vyučující	Četnost odpovědí					Hodnota p
		1	2	3	4	5	
1	Kořínek	10	5	1	0	0	0,003
	Voborník	1	8	2	3	0	
2	Kořínek	10	5	1	0	0	0,006
	Voborník	2	6	4	1	1	
3	Kořínek	4	9	0	2	1	0,279
	Voborník	1	8	3	1	1	
4	Kořínek	4	10	1	0	1	0,0002
	Voborník	0	1	5	5	3	
5	Kořínek	7	7	1	0	1	0,006

	Voborník	1	4	3	6	0	
6	Kořínek	1	7	4	1	3	0,253
	Voborník	0	4	2	7	1	
7	Kořínek	1	4	4	3	3	0,934
	Voborník	0	3	6	4	1	
8	Kořínek	9	6	1	0	0	0,0002
	Voborník	1	2	3	5	3	
9	Kořínek	13	3	0	0	0	0,0002
	Voborník	1	8	2	3	0	
10	Kořínek	8	4	1	1	1	0,001
	Voborník	0	1	6	6	1	
11	Kořínek	6	4	2	3	1	0,023
	Voborník	0	2	6	4	2	
12	Kořínek	3	5	3	2	2	0,810
	Voborník	2	7	1	4	0	
13	Kořínek	6	6	0	3	0	0,052
	Voborník	1	6	3	2	2	
14	Kořínek	2	11	1	1	0	0,006
	Voborník	1	2	7	4	0	
15	Kořínek	1	3	4	3	4	0,777
	Voborník	0	4	1	6	3	
16	Kořínek	2	8	2	2	1	0,015
	Voborník	8	5	1	0	0	
17	Kořínek	5	5	1	3	1	0,052
	Voborník	10	3	0	1	0	
18	Kořínek	5	4	2	3	1	0,585
	Voborník	4	6	4	0	0	
19	Kořínek	2	6	2	3	2	0,144
	Voborník	5	5	2	2	0	
20	Kořínek	4	4	2	4	1	0,647
	Voborník	2	6	6	0	0	
21	Kořínek	5	9	1	0	0	0,432
	Voborník	6	2	1	5	0	
22	Kořínek	3	6	2	3	1	0,432

	Voborník	4	6	3	0	1	
23	Kořínek	4	6	3	1	1	0,471
	Voborník	0	10	1	3	0	
24	Kořínek	0	3	5	6	1	0,585
	Voborník	1	2	3	5	3	

## Příloha G: Vyhodnocení výsledků studentů jedním vyučujícím

Tabulka 141 – Výsledky studentů z posttestu z hlavních částí v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	1,00	
	Průměr	46,31%	47,66%
	Úspěšní studenti	5	7
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,649	
	Průměr	40,56%	32,80%
	Úspěšní studenti	5	5
	Neúspěšní studenti	3	3
<b>Teoretická část</b>	Hodnota p	0,166	
	Průměr	52,06%	62,52%
	Úspěšní studenti	5	11
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,425	
	Průměr	44,18%	29,07%
	Úspěšní studenti	5	4
	Neúspěšní studenti	4	5
<b>Strukturovaná část</b>	Hodnota p	0,894	
	Průměr	38,15%	35,28%
	Úspěšní studenti	4	7
	Neúspěšní studenti	4	4

Tabulka 142 – Výsledky studentů z posttestu z dílčích částí teoretické části v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,247	
	Průměr	36,61%	48,26%
	Úspěšní studenti	2	6

	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,077	
	Průměr	68,96%	79,52%
	Úspěšní studenti	9	15
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,287	
	Průměr	36,90%	47,69%
	Úspěšní studenti	2	8
	Neúspěšní studenti	3	2

Tabulka 143 – Výsledky studentů z posttestu z dílčích částí objektové části v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,160	
	Průměr	55,36%	30,42%
	Úspěšní studenti	7	5
	Neúspěšní studenti	4	9
<b>Metody a vlastnosti</b>	Hodnota p	0,805	
	Průměr	35,97%	31,34%
	Úspěšní studenti	5	5
	Neúspěšní studenti	5	6
<b>Instance a volání metod</b>	Hodnota p	0,115	
	Průměr	50,89%	25,00%
	Úspěšní studenti	7	5
	Neúspěšní studenti	6	12

Tabulka 144 – Výsledky studentů z posttestu z dílčích částí strukturované části v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje		
<b>Naplnění a výpis posloupností</b>	Hodnota p	0,362	
	Průměr	55,36%	40,28%
	Úspěšní studenti	9	8
	Neúspěšní studenti	4	5
<b>Návrh algoritmů metod</b>	Hodnota p	0,805	
	Průměr	29,55%	32,78%
	Úspěšní studenti	3	4
	Neúspěšní studenti	6	7

Tabulka 145 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z hlavních částí

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	0,782	
	Průměr	50,02%	49,61%
	Úspěšní studenti	4	7
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,436	
	Průměr	45,88%	34,73%
	Úspěšní studenti	4	5
	Neúspěšní studenti	1	2
<b>Teoretická část</b>	Hodnota p	0,183	
	Průměr	54,16%	64,50%
	Úspěšní studenti	4	11
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,292	
	Průměr	49,33%	30,78%
	Úspěšní studenti	4	4
	Neúspěšní studenti	2	4
<b>Strukturovaná</b>	Hodnota p	0,564	

část			
	Průměr	43,58%	37,35%
	Úspěšní studenti	4	7
	Neúspěšní studenti	2	3

Tabulka 146 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí teoretické části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,248	
	Průměr	37,59%	50,82%
	Úspěšní studenti	2	6
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,152	
	Průměr	70,23%	80,76%
	Úspěšní studenti	6	15
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,436	
	Průměr	41,67%	50,49%
	Úspěšní studenti	1	8
	Neúspěšní studenti	1	1

Tabulka 147 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí objektové části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,126	
	Průměr	59,75%	32,21%
	Úspěšní studenti	5	5
	Neúspěšní studenti	2	8
<b>Metody a vlastnosti</b>	Hodnota p	0,706	

	Průměr	38,75%	33,19%
	Úspěšní studenti	4	5
	Neúspěšní studenti	3	5
<b>Instance a volání metod</b>	Hodnota p	0,075	
	Průměr	60,00%	26,47%
	Úspěšní studenti	6	5
	Neúspěšní studenti	4	11

Tabulka 148 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí strukturované části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,259	
	Průměr	61,25%	42,65%
	Úspěšní studenti	7	8
	Neúspěšní studenti	2	4
<b>Návrh algoritmů metod</b>	Hodnota p	1,000	
	Průměr	34,75%	34,71%
	Úspěšní studenti	3	4
	Neúspěšní studenti	6	4

Tabulka 149 – Výsledky studentů z retestu 1 z hlavních částí

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	0,407	
	Průměr	49,95%	55,92%
	Úspěšní studenti	4	7
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,688	
	Průměr	39,14%	45,11%
	Úspěšní studenti	3	5



	Neúspěšní studenti	3	3
<b>Teoretická část</b>	Hodnota p	0,259	
	Průměr	60,77%	66,74%
	Úspěšní studenti	5	12
	Neúspěšní studenti	0	0
<b>Objektivá část</b>	Hodnota p	0,651	
	Průměr	38,10%	46,09%
	Úspěšní studenti	4	6
	Neúspěšní studenti	3	3
<b>Strukturovaná část</b>	Hodnota p	0,688	
	Průměr	39,83%	44,46%
	Úspěšní studenti	3	5
	Neúspěšní studenti	3	3

Tabulka 150 – Výsledky studentů z retestu 1 z dílčích částí teoretické části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,960	
	Průměr	51,77%	55,55%
	Úspěšní studenti	4	6
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,152	
	Průměr	73,29%	80,37%
	Úspěšní studenti	8	14
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,482	
	Průměr	47,50%	54,41%
	Úspěšní studenti	4	9
	Neúspěšní studenti	1	1

Tabulka 151 – Výsledky studentů z retestu 1 z dílčích částí objektové části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,547	
	Průměr	42,50%	44,41%
	Úspěšní studenti	4	5
	Neúspěšní studenti	5	4
<b>Metody a vlastnosti</b>	Hodnota p	0,530	
	Průměr	33,26%	43,49%
	Úspěšní studenti	3	7
	Neúspěšní studenti	4	5
<b>Instance a volání metod</b>	Hodnota p	0,782	
	Průměr	45,00%	51,47%
	Úspěšní studenti	4	12
	Neúspěšní studenti	3	3

Tabulka 152 – Výsledky studentů z retestu 1 z dílčích částí strukturované části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,436	
	Průměr	46,75%	59,71%
	Úspěšní studenti	4	12
	Neúspěšní studenti	3	3
<b>Návrh algoritmů metod</b>	Hodnota p	1,00	
	Průměr	36,38%	36,84%
	Úspěšní studenti	2	4
	Neúspěšní studenti	4	5

Tabulka 153 – Výsledky studentů z posttestu z hlavních částí v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	0,777	
	Průměr	46,31%	50,28%
	Úspěšní studenti	5	4
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,877	
	Průměr	40,56%	36,03%
	Úspěšní studenti	5	4
	Neúspěšní studenti	3	1
<b>Teoretická část</b>	Hodnota p	0,095	
	Průměr	52,06%	64,53%
	Úspěšní studenti	5	8
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,918	
	Průměr	44,18%	37,36%
	Úspěšní studenti	5	4
	Neúspěšní studenti	4	2
<b>Strukturovaná část</b>	Hodnota p	0,959	
	Průměr	38,15%	35,14%
	Úspěšní studenti	4	4
	Neúspěšní studenti	4	2

Tabulka 154 – Výsledky studentů z posttestu z dílčích částí teoretické části v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,100	
	Průměr	36,61%	53,47%
	Úspěšní studenti	2	5
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,054	

	Průměr	68,96%	81,87%
	Úspěšní studenti	9	10
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,471	
	Průměr	36,90%	45,83%
	Úspěšní studenti	2	5
	Neúspěšní studenti	3	2

Tabulka 155 – Výsledky studentů z posttestu z dílčích částí objektové části v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,572	
	Průměr	55,36%	42,50%
	Úspěšní studenti	7	5
	Neúspěšní studenti	4	5
<b>Metody a vlastnosti</b>	Hodnota p	0,607	
	Průměr	35,97%	41,11%
	Úspěšní studenti	5	4
	Neúspěšní studenti	5	2
<b>Instance a volání metod</b>	Hodnota p	0,198	
	Průměr	50,89%	29,17%
	Úspěšní studenti	7	4
	Neúspěšní studenti	6	8

Tabulka 156 – Výsledky studentů z posttestu z dílčích částí strukturované části v akademických letech 2013/2014 a 2014/2015

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Naplnění a výpis</b>	Hodnota p	0,382	

<b>posloupnosti</b>			
	Průměr	55,36%	39,17%
	Úspěšní studenti	9	5
	Neúspěšní studenti	4	3
<b>Návrh algoritmů metod</b>	Hodnota p	0,719	
	Průměr	29,55%	33,13%
	Úspěšní studenti	3	2
	Neúspěšní studenti	6	4

Tabulka 157 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z hlavních částí

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Celkové výsledky</b>	Hodnota p	0,860	
	Průměr	50,02%	53,54%
	Úspěšní studenti	4	4
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,699	
	Průměr	45,88%	39,30%
	Úspěšní studenti	4	4
	Neúspěšní studenti	1	0
<b>Teoretická část</b>	Hodnota p	0,072	
	Průměr	54,16%	67,78%
	Úspěšní studenti	4	8
	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,778	
	Průměr	49,33%	40,76%
	Úspěšní studenti	4	4
	Neúspěšní studenti	2	1
<b>Strukturovaná část</b>	Hodnota p	0,725	
	Průměr	43,58%	38,33%
	Úspěšní studenti	4	4
	Neúspěšní studenti	2	1

Tabulka 158 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí teoretické části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	0,072	
	Průměr	37,59%	57,87%
	Úspěšní studenti	2	5
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,098	
	Průměr	70,23%	84,00%
	Úspěšní studenti	6	10
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,549	
	Průměr	41,67%	50,00%
	Úspěšní studenti	1	5
	Neúspěšní studenti	1	1

Tabulka 159 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí objektové části

Část posttestu	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,549	
	Průměr	59,75%	46,36%
	Úspěšní studenti	5	5
	Neúspěšní studenti	2	4
<b>Metody a vlastnosti</b>	Hodnota p	0,673	
	Průměr	38,75%	44,85%
	Úspěšní studenti	4	4
	Neúspěšní studenti	3	1
<b>Instance a volání metod</b>	Hodnota p	0,130	

	Průměr	60,00%	31,82%
	Úspěšní studenti	6	4
	Neúspěšní studenti	4	7

**Tabulka 160 – Výsledky studentů, kteří absolvovali i retest 1, z posttestu z dílčích částí strukturované části**

<b>Část posttestu</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,307	
	Průměr	61,25%	42,73%
	Úspěšní studenti	7	5
	Neúspěšní studenti	2	2
<b>Návrh algoritmů metod</b>	Hodnota p	0,832	
	Průměr	34,75%	36,14%
	Úspěšní studenti	3	2
	Neúspěšní studenti	4	3

**Tabulka 161 – Výsledky studentů z retestu 1 z hlavních částí**

<b>Část retestu 1</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Celkové výsledky</b>	Hodnota p	0,170	
	Průměr	49,95%	62,46%
	Úspěšní studenti	4	6
	Neúspěšní studenti	0	0
<b>Praktická část</b>	Hodnota p	0,275	
	Průměr	39,14%	55,95%
	Úspěšní studenti	3	4
	Neúspěšní studenti	3	3
<b>Teoretická část</b>	Hodnota p	0,218	
	Průměr	60,77%	68,97%
	Úspěšní studenti	5	8

	Neúspěšní studenti	0	0
<b>Objektová část</b>	Hodnota p	0,307	
	Průměr	38,10%	55,66%
	Úspěšní studenti	4	5
	Neúspěšní studenti	3	0
<b>Strukturovaná část</b>	Hodnota p	0,245	
	Průměr	39,83%	56,14%
	Úspěšní studenti	3	4
	Neúspěšní studenti	3	0

Tabulka 162 – Výsledky studentů z retestu 1 z dílčích částí teoretické části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Doplňování kódu</b>	Hodnota p	1,000	
	Průměr	51,77%	55,50%
	Úspěšní studenti	4	3
	Neúspěšní studenti	0	0
<b>Interaktivní prvky</b>	Hodnota p	0,113	
	Průměr	73,29%	82,15%
	Úspěšní studenti	8	9
	Neúspěšní studenti	0	0
<b>Doplnění hodnot proměnných</b>	Hodnota p	0,342	
	Průměr	47,50%	59,09%
	Úspěšní studenti	4	6
	Neúspěšní studenti	1	0

Tabulka 163 – Výsledky studentů z retestu 1 z dílčích částí objektové části

Část retestu 1	Sledované údaje	Experimentální skupina	Kontrolní skupina
<b>Definice konstruktoru a datových položek</b>	Hodnota p	0,291	



	Průměr	42,50%	55,00%
	Úspěšní studenti	4	4
	Neúspěšní studenti	5	1
<b>Metody a vlastnosti</b>	Hodnota p	0,245	
	Průměr	33,26%	53,25%
	Úspěšní studenti	3	6
	Neúspěšní studenti	4	2
<b>Instance a volání metod</b>	Hodnota p	0,549	
	Průměr	45,00%	60,23%
	Úspěšní studenti	5	6
	Neúspěšní studenti	5	2

Tabulka 164 – Výsledky studentů z retestu 1 z dílčích částí strukturované části

<b>Část retestu 1</b>	<b>Sledované údaje</b>	<b>Experimentální skupina</b>	<b>Kontrolní skupina</b>
<b>Naplnění a výpis posloupnosti</b>	Hodnota p	0,130	
	Průměr	46,75%	75,23%
	Úspěšní studenti	4	9
	Neúspěšní studenti	3	0
<b>Návrh algoritmů metod</b>	Hodnota p	0,503	
	Průměr	36,38%	46,59%
	Úspěšní studenti	2	3
	Neúspěšní studenti	4	1

**Tabulka 165 – Výsledky dotazníkového šetření stejného vyučujícího z akademických let 2013/2014 a 2014/2015**

Číslo otázky	Vyučující	Četnost odpovědí					Hodnota P
		1	2	3	4	5	
1	Kořínek	8	2	0	0	0	0,444
	Kořínek	10	5	1	0	0	
2	Kořínek	5	4	1	0	0	0,598
	Kořínek	10	5	1	0	0	
3	Kořínek	3	5	1	1	0	0,833
	Kořínek	4	9	0	2	1	
4	Kořínek	3	4	1	2	0	0,752
	Kořínek	4	10	1	0	1	
5	Kořínek	4	2	3	0	1	0,477
	Kořínek	7	7	1	0	1	
6	Kořínek	2	2	3	2	1	0,970
	Kořínek	1	7	4	1	3	
7	Kořínek	1	5	0	2	2	0,527
	Kořínek	1	4	4	3	3	
8	Kořínek	5	3	2	0	0	0,635
	Kořínek	9	6	1	0	0	
9	Kořínek	5	5	0	0	0	0,197
	Kořínek	13	3	0	0	0	
10	Kořínek	2	4	3	1	0	0,268
	Kořínek	8	4	1	1	1	
11	Kořínek	2	5	1	2	0	0,854
	Kořínek	6	4	2	3	1	
12	Kořínek	2	4	1	2	1	0,911
	Kořínek	3	5	3	2	2	
13	Kořínek	3	6	1	0	0	1,000
	Kořínek	6	6	0	3	0	
14	Kořínek	1	3	4	1	1	0,806
	Kořínek	2	11	1	1	0	
15	Kořínek	1	4	0	5	0	0,360

	Kořínek	1	3	4	3	4	
16	Kořínek	2	2	2	3	1	0,437
	Kořínek	2	8	2	2	1	
17	Kořínek	2	4	2	1	1	0,677
	Kořínek	5	5	1	3	1	
18	Kořínek	4	4	0	1	1	0,598
	Kořínek	5	4	2	3	1	
19	Kořínek	1	3	4	1	1	0,912
	Kořínek	2	6	2	3	2	
20	Kořínek	4	3	1	2	0	0,390
	Kořínek	4	4	2	4	1	
21	Kořínek	6	1	2	1	0	0,739
	Kořínek	5	9	1	0	0	
22	Kořínek	2	4	2	2	0	0,890
	Kořínek	3	6	2	3	1	
23	Kořínek	3	2	1	3	1	0,542
	Kořínek	4	6	3	1	1	
24	Kořínek	1	2	2	1	4	0,677
	Kořínek	0	3	5	6	1	