

Bc. Jana Bromová

Výuka algoritmizace na ZŠ – aktuální stav

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta Katedra informatiky

Jihočeská univerzita

Pedagogická fakulta

Katedra informatiky

Výuka algoritmizace na ZŠ – aktuální stav

Autor: Bc. Jana Bromová

Vedoucí práce: doc. PaedDr. Jiří Vaníček, Ph. D.

České Budějovice 2012

Název práce: Výuka algoritmizace na ZŠ – aktuální stav

Autor: Bc. Jana BROMOVÁ

Katedra: Katedra Informatiky

Vedoucí diplomové práce: doc. PaedDr. Jiří Vaniček, Ph. D.

E-mail vedoucího: vanicek@pf.jcu.cz

Abstrakt:

Cílem této diplomové práce je zmapovat současný stav výuky algoritmizace na základních školách.

V teoretické části jsou uvedeny základní informace přibližující danou problematiku - souvislosti výuky algoritmizace a RVP ZV a také základní předpoklady výuky algoritmizace, jako je historické hledisko, cíle, metody a možnosti výuky algoritmizace. Nedílnou součástí je i podpora výuky algoritmizace, do níž jsou zařazeny učebnice věnující se algoritmizaci a programování, programová podpora, projekty a soutěže.

Praktickou část tvoří pedagogický výzkum. Tento výzkum se skládá z pěti dílčích výzkumů:

Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB.

Přehled mimoškolní výuky algoritmizace v ČB.

Zmapování celorepublikové výuky algoritmizace na ZŠ.

Zkušenosti učitele s učebnicí.

Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“.

Klíčová slova:

algoritmizace, algoritmus, programování, výzkum, ZŠ, výuka, Bobřík informatiky, RVP

Title: Current situation in teaching of algorithmization in secondary school

Author: Bc. Jana BROMOVÁ

Department: Katedra Informatiky

Supervisit: doc. PaedDr. Jiří Vaníček, Ph. D.

Supervisor's e-mail address: vanicek@pf.jcu.cz

Abstract:

The aim of this thesis is mapping of current state of teaching of algorithmization in secondary school.

Basic information about the subject is given in theoretical part - a relation between teaching of algorithmization and RVP ZV, and basic assumptions of teaching of algorithmization: a historical aspect, methods, aims and possibilities of teaching of algorithmization (and an approach on teaching). The essential part of teaching of algorithmization is an education support - a textbook about algorithmization and programming, programming support, projects and competitions for students.

Practical part is a pedagogical research. This research consists of five themes:

A quantitative research of teaching of algorithmization in secondary school in České Budějovice.

A review of extracurricular teaching of algorithmization in ČB.

A statewide mapping of teaching of algorithmization in Czech Republic.

A teacher's experience with a textbook.

A statewide research on a success rate of the competition „Bobřík informatiky“ (informatics beaver).

Keywords:

Algorithmization, algorithm, programming, research, secondary schools, teaching

Poděkování

Děkuji doc. PaedDr. Jiřímu Vaníčkovi, Ph. D. za informace, trpělivost, rady a materiály, které mi poskytl během vypracovávání diplomové práce.

Prohlašuji, že jsem předloženou diplomovou prací vypracovala samostatně a použila jen pramenů, které cituji a uvádím v seznamu použité literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Č. Budějovicích 26. 4. 2012

Podpis

Obsah

ÚVOD	1
CÍLE	3
TEORETICKÁ VÝCHODISKA	5
1.0 ALGORITMIZACE.....	5
1.1 <i>Co je to algoritmizace</i>	5
1.2 <i>Co je to algoritmus</i>	6
2.0 ALGORITMIZACE A RVP.....	8
2.1 <i>RVP ZV</i>	8
2.2 <i>RVP G a odborné vzdělání</i>	8
2.3 <i>Mezinárodní srovnání</i>	10
3.0 VÝUKA ALGORITMIZACE.....	13
3.1 <i>Historie výuky algoritmizace na ZŠ v ČR</i>	13
3.2 <i>Cíle a metodika výuky algoritmizace</i>	13
3.3 <i>Možnosti výuky algoritmizace</i>	16
4.0 PODPORA VÝUKY ALGORITMIZACE.....	17
4.1 <i>Vývojové nástroje</i>	17
4.2 <i>Učebnice zabývající se algoritmizací/programováním</i>	23
4.3 <i>Projekty zabývající se algoritmizací na ZŠ</i>	24
METODIKA VÝZKUMU	26
1.0 OBSAHOVÁ ANALÝZA.....	26
2.0 METODA ROZHOVORU.....	26
3.0 POZOROVÁNÍ.....	27
4.0 ANALÝZA ÚLOH.....	27
A. KVANTITATIVNÍ VÝZKUM VÝUKY ALGORITMIZACE NA ZŠ V ČB.	28
1.0 POPIS VÝZKUMNÉHO VZORKU.....	28
2.0 VÝSLEDKY.....	28
3.0 ROZBOR PŘÍSTUPU K VÝUCE.....	30

Obsah

3.1	<i>Škola A</i>	30
3.2	<i>Škola B</i>	32
B.	PŘEHLED MIMOŠKOLNÍ VÝUKY ALGORITMIZACE V ČB	37
C.	ZMAPOVÁNÍ CELOREPUBLIKOVÉ VÝUKY ALGORITMIZACE NA ZŠ.	39
D.	ZKUŠENOSTI UČITELE S UČEBNICÍ	41
E.	CELOREPUBLIKOVÝ VÝZKUM ÚSPĚŠNOSTI V SOUTĚŽI „BOBŘÍK INFORMATIKY“	42
1.0	POPIS VÝZKUMNÉHO VZORKU	42
2.0	VÝSLEDKY	44
3.0	ROZBOR ÚLOH	48
3.1	<i>Kategorizace</i>	87
3.2	<i>Analýza kategorií</i>	90
	DISKUSE	94
	SHRNUTÍ A ZÁVĚR	97

Úvod

V současné přetechnizované době, kdy se s počítači setkáváme doslova na každém rohu, rostou nároky na počítačovou gramotnost. Součástí počítačové gramotnosti je i algoritmická gramotnost. Rozvoj algoritmického myšlení u žáků napomáhá nejen vychovat lepší počítačové programátory, ale zdokonaluje i běžnější činnosti - například schopnost efektivněji vyhledávat na internetu. Z tohoto důvodu se stále častěji setkáváme s otázkou, zda algoritmizaci a programování učít již v počátku vzdělávání a zda mají žáci na základní škole již takovou mentální úroveň, aby byli schopni algoritmického myšlení a pochopení algoritmických struktur.

Někteří odborníci jsou toho mínění, že algoritmizaci je možné vyučovat již na prvním stupni ZŠ. K těmto odborníkům patří například profesor Melichar z UJEP. Profesor Melichar (Melichar, 2006) se věnuje didaktice matematiky a ve své práci **Algoritmy na 1. stupni ZŠ** uvádí, že již žáci na prvním stupni základní školy jsou schopni algoritmického myšlení.

S algoritmy se setkáváme v životě na každém kroku. K nejprimitivnějším algoritmům patří instrukce typu „Po použití znehodnoťte“ nebo „Při odchodu zhasněte světlo“. Typickým příkladem algoritmů jsou návody na obsluhu telefonního automatu, automatické pračky nebo jiných přístrojů, předpisy na přípravu jídel.

Stejně jako v každodenním životě, tak i ve vědě a technice je úloha algoritmů významná. V každé oblasti vědy a techniky existuje celá řada nejrůznějších příruček. Každá taková příručka je ve své podstatě sbírkou algoritmů nashromážděných za dlouhá léta příslušnou disciplínou. Existují příručky pro konstruktéry, techniky, mistry, lékaře, architekty apod. Velký význam mají samozřejmě algoritmy v matematice. Např. algoritmus dělení.

Formulace a použití algoritmů je svázáno s dovedností jasně a srozumitelně formulovat pravidla a přesně je dodržovat.

V libovolné oblasti činnosti často vzniká potřeba sestavit určité instrukce, pravidla, předpisy (např. pravidla silničního provozu atd.). Ne každý dovede tyto instrukce, předpisy, pravidla (tj. algoritmy) vytvořit, ale přesně je dodržovat musí umět každý člověk, protože vlastně na každém kroku plníme jistá pravidla vyjadřující organizaci našeho života.

Maďarský profesor Varga zdůrazňuje, že při výuce vývojových diagramů (a tedy i algoritmů) je třeba začít nějakou známou činností ze „světa dětí“. Např. ranní odchod

Úvod

do školy. Jednotlivé úkony celého procesu jsou zapsány v operačních blocích. Úkolem žáků je dát jednotlivé činnosti do správného pořadí, jak je ráno provádějí. Žáci plní úkol například popsat činnosti při úspěšné přípravě čaje a činnosti časově seřadit nebo činnosti pro napsání sms na mobilním telefonu, atp.

Tématem této diplomové práce je zmapovat současný stav výuky algoritmizace na druhém stupni ZŠ. Z historického hlediska se algoritmizace a programování na různých typech škol vyučuje již od osmdesátých let minulého století. Úroveň této výuky postupem času kolísá. Tomuto tématu se věnuje Dr. Pitner (Pitner, 2000) ve své práci nazvané **Výuka programování na základní a střední škole**.

Tato práce vychází z cílů výuky informačních a komunikačních technologií, které jsou uvedeny v Rámcovém vzdělávacím programu pro základní vzdělávání (RVP ZV). Zde je jedním z cílů: „*schopnosti formulovat svůj požadavek a využívat při interakci s počítačem algoritnické myšlení*“ (Balada, 2007a, s. 32). Tato práce se pokusí zmapovat, jak jednotlivé školy interpretují tento cíl výuky. Důraz bude kladen na instituce, které se výuce algoritmizace věnují.

Cíle

Cíle

Cílem této práce je zmapovat současný stav výuky algoritmizace na základních školách a dalších institucích zabývajících se vzděláváním dětí ve věku žáků druhého stupně ZŠ.

Jedná se o výzkum. Tento výzkum má dvě části. První je nalezení vzorku deseti vzdělávacích institucí, druhou částí je výzkum a rozbor přístupu těchto institucí k výuce.

Z důvodu malého množství nalezených škol zabývajících se výukou algoritmizace na území Jihočeského kraje, došlo k mírnému odklonu od původního zadání této práce:

Cílem teoretické části je seznámení s algoritmizací. Součástí je i obsahová analýza základních oficiálních pedagogických dokumentů, a to zejména RVP ZV a mezinárodní srovnání se Slovenskem a jejich ŠVP ISCED2 (Štátny vzdelavaci program pre 2. stupeň základných škôl). Dalším cílem teoretické části je uvedení do problematiky výuky algoritmizace. Jedná se zejména o vytýčení cílů a metod výuky algoritmizace. Teoretická východiska by měla také obsahovat zmapování podpory výuky, a to zejména komparace učebnic (včetně náhledu do cizích učebnic, např. slovenských) a vývojová prostředí.

Cílem praktické části je zmapovat současný stav výuky algoritmizace na ZŠ, a to prostřednictvím pěti samostatných oblastí.

Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB.

Budou dotázány všechny základní školy a nižší gymnázia působící na území města České Budějovice, zda a v jaké míře je výuka algoritmizace vyučována.

Přehled mimoškolní výuky algoritmizace v ČB.

Jedná se o zájmové kroužky algoritmizace a programování provozované institucemi, jako jsou například domy dětí a občanská sdružení, zabývajících se vzděláváním dětí ve věku žáků druhého stupně ZŠ.

Zmapování celorepublikové výuky algoritmizace na ZŠ.

Prostřednictvím internetových vyhledávačů nalezení škol, zabývajících se výukou algoritmizace a seznámení s jejich přístupem k výuce.

Zkušenosti učitele s učebnicí.

Jde o nalezení učitele, který při výuce využívá nějakou učebnici a následným rozбором obsahové části a úloh zhodnocení dané učebnice pro potřeby této výuky.

Cíle

Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“.

Zmapování úspěšnosti žáků při řešení algoritmických úloh v soutěži Bobřík informatiky ve všech již uskutečněných ročnících (čtyři ročníky) a to v kategoriích Benjamin a Kadet, které odpovídají věkové skupině žáků druhého stupně ZŠ.

Na základě obsahové analýzy byly stanoveny následující hypotézy:

- A. V současné době se výuce algoritmizace věnuje jen malé procento českobudějovických základních škol.
- B. Při současném rozmachu organizací organizujících zájmové kroužky dětí bude v Českých Budějovicích několik kroužků zabývajících se výukou algoritmizace a programování.
- C. V každém větším městě v ČR se najde alespoň jedna ZŠ vyučující algoritmizaci či programování.
- D. Na českém trhu se nabízí velké množství učebnic informatiky a výpočetní techniky, výuce programování se jistě bude věnovat více učebnic. Zaměření těchto učebnic bude určitě také rozmanité, proto nebude obtížné nalézt učitele, který vyučuje programování s jejich pomocí.
- E. V soutěži Bobřík informatiky je úspěšnost řešení algoritmických úloh stejná jako u ostatních úloh.

Teoretická východiska

1.0 Algoritmizace

1.1 Co je to algoritmizace

Algoritmizace je metodický přístup k vytváření programu. Zabývá se formulací postupů řešení daného problému. Výsledkem algoritmizace je algoritmus, což je posloupnost příkazů popisující řešení daného problému. (Dohnal, 2009)

Důležitou vlastností algoritmizace je relativní neměnnost v čase. To je zásadní rozdíl od programovacích jazyků. Programovací jazyky se časem stávají zastaralými a jsou nahrazovány novějšími a dokonalejšími jazyky. Algoritmizace je nezávislá na programovacím jazyku, zůstává stále stejná a je možné ji využívat v různých vývojových prostředích.

Algoritmizaci lze rozdělit do několika etap:

- ❖ Formulace problému
- ❖ Analýza úlohy
- ❖ Vytvoření algoritmu
- ❖ Sestavení programu
- ❖ Odladění programu

Formulace problému spočívá v přesném vymezení požadavků, určení výchozích hodnot a požadovaných výsledků. Výsledky je nutné dále specifikovat, jakou mají mít formu a s jakou přesností se bude pracovat.

Při **analýze úlohy** se ověřuje řešitelnost problému a provádí se hrubý nástin řešení.

Vytvoření algoritmu úlohy je realizování sestavením přesného sledu operací vedoucích k vyřešení dané úlohy. Tento algoritmus pouze udává postup řešení daného problému, ale samotný problém neřeší.

Následuje **sestavení programu** dle daného algoritmu. Tento program je již napsán v konkrétním programovacím jazyce.

Posední fází je **odladění programu**, během něhož se odstraní z programu chyby. Jedná se zejména o chyby syntaktické, které odhalí již překladač. Mnohem závažnější jsou chyby ve funkčnosti, kdy je nutné se vrátit zpět k analýze úlohy a sestavení algoritmu a nalezení chyby na této úrovni.

Teoretická východiska

1.2 Co je to algoritmus

Algoritmus je přesný postup, který popisuje řešení daného problému v konečném počtu kroků. (Pšenčíková, 2007).

Algoritmus je tvořen jednotlivými primitivními kroky, které na sebe přesně navazují a jejichž provádění vede k vyřešení daného programu. Do algoritmu jsou dodána vstupní data, která se jeho prováděním transformují na data výstupní.

Podle struktury algoritmu je možné rozlišit algoritmus nevětvený tzv. sekvenční a větvený tzv. paralelní.

Příkladem slovního algoritmu může být postup při provádění libovolné činnosti, jako například tankování vozidla u benzínové čerpací stanice.

*Zastavit vozidlo
Vystoupit z vozidla
Je řidič připoután bezpečnostním pásem?
 Jestliže ANO – rozepnout bezpečnostní pás
 Najít zámek bezpečnostního pásu
 Uvolnit pojistku bezpečnostního pásu
 Sejmout bezpečnostní pás
 Umístit bezpečnostní pás do pohotovostní polohy
 Jestliže NE – přejdi na 2.b
Otevřít dveře vozidla
Vystoupit
Zavřít dveře
Natankovat
Zaplatit za zboží
Nastoupit do vozidla
Rozjet se*

Obrázek 1: Příklad slovního algoritmu

Převzato a upraveno z Melichar2006

Každý algoritmus musí splňovat určitá pravidla. Patří mezi ně zejména determinovanost, obecnost, rezultativnost a konečnost. Informace uvedené v následujícím výčtu vycházejí z údajů v publikacích Pšenčíkové (Pšenčíková, 2007) a Krčka a Kremla (Krček, Kreml, 1993).

Determinovanost znamená, že každý krok musí být jednoznačně a přesně určen. Vždy musí být zcela jasné, co má následovat.

Teoretická východiska

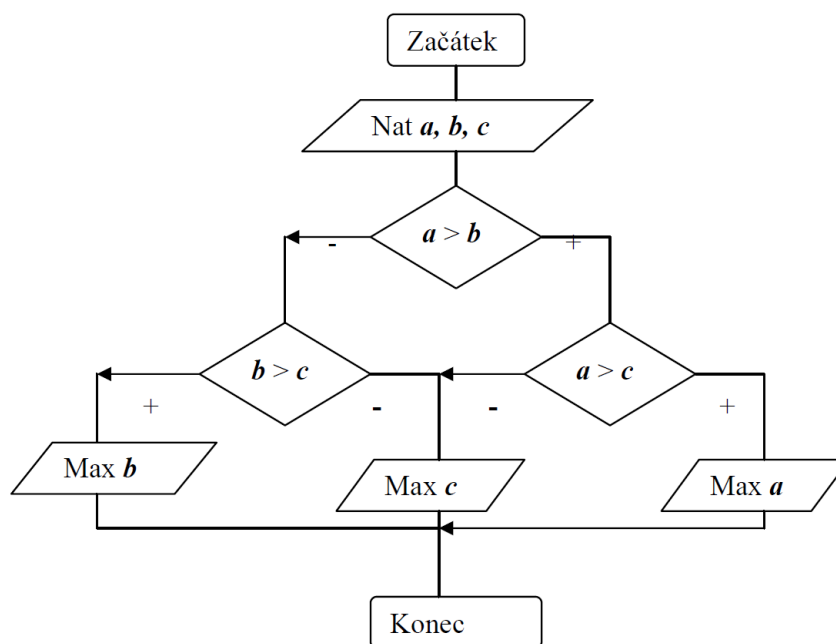
Obecnost algoritmu znamená, že je univerzální. Tj. není sestaven pouze pro jeden konkrétní problém, ale pro všechny problémy založené na stejném principu. Například algoritmus pro operaci sčítání čísel pět a šest by měl být použitelný pro jakékoli sčítání.

Rezultativnost je vlastnost algoritmu, která říká, že po provedení určitého počtu kroků algoritmus dosáhne správného výsledku.

Konečnost říká, že algoritmus po svém provedení skončí. Počet těchto kroků závisí na vstupních datech a strukturovanosti algoritmu. Pro určitý vstup může být výsledku dosaženo již po několika málo krocích, zatímco pro jiný vstup se provede mnohem větší počet kroků.

Pro daný problém lze vytvořit více algoritmů. V tomto případě je důležité použít co nejefektivnější algoritmus. Tj. algoritmus, který výsledku dosáhne v co nejkratším čase.

Algoritmy můžeme zapisovat různě. Jednou z možností je například slovní zápis. Dalším a častěji používaným způsobem je grafický zápis. Ten se realizuje pomocí tzv. vývojových diagramů. Mezi složitější způsoby zápisu patří zápis pseudokódem, nebo přímo programovacím jazykem. Této problematice se věnuje Jan Melichar ve své práci **Algoritmy na 1. stupni základní školy**.



Obrázek 2: Vývojový diagram – výběr největšího čísla ze tří přirozených čísel.

Převzato a upraveno z <http://pf.ujep.cz/files/KMA_poznamkydidamat04.pdf>.

2.0 Algoritmizace a RVP

2.1 RVP ZV

Rámcový vzdělávací program pro základní vzdělávání je základní dokument charakterizující obsah, cíle a očekávané výstupy základního vzdělání. RVP je členěn do vzdělávacích oblastí. Mezi tyto oblasti patří například Matematika a její aplikace, Informační a komunikační technologie, Člověk a společnost, Člověk a příroda aj.

Algoritmizace je v rámci RVP začleněna do oblasti nazvané Informační a komunikační technologie. Tato oblast udává základní úroveň informační gramotnosti – *„získat elementární dovednosti v ovládnutí výpočetní techniky a moderních informačních technologií, orientovat se ve světě informací, tvořivě pracovat s informacemi a využívat je při dalším vzdělávání i v praktickém životě. Vzhledem k narůstající potřebě osvojení si základních dovedností práce s výpočetní technikou byla vzdělávací oblast Informační a komunikační technologie zařazena jako povinná součást základního vzdělávání na 1. a 2. stupni. Získané dovednosti jsou v informační společnosti nezbytným předpokladem uplatnění na trhu práce i podmínkou k efektivnímu rozvíjení profesní i zájmové činnosti.“* (Balada, 2007a, s. 32).

Jediná zmínka o algoritmizaci, resp. o algoritmičtém myšlení je v RVP v části o cílovém zaměření vzdělávací oblasti. Zde jedním z cílů je: *„schopnosti formulovat svůj požadavek a využívat při interakci s počítačem algoritmičtého myšlení“* (Balada, 2007a, s. 32).

Z tohoto důvodu je zařazení algoritmizace do výuky zcela v režii základních škol. Záleží pouze na učitelích výpočetní techniky, zda se bude algoritmizace vyučovat a v jakém rozsahu.

2.2 RVP G a odborné vzdělání

Absolvent základní školy může pokračovat na gymnázium nebo střední školu, či nějaký učňovský obor. Jak je to s výukou algoritmizace na tomto stupni vzdělání? Co říkají RVP pro tyto organizace?

2.2.1 Gymnázia

Rámcové vzdělávací programy pro gymnázia jsou celkem tři, a to pro dvojjazyčná gymnázia (RVP DG), pro gymnázia se sportovní přípravou (RVP GSP) a pro ostatní gymnázia (RVP G). Po obsahové stránce je vzdělávací oblast informatika a ICT ve všech těchto dokumentech totožná.

Teoretická východiska

V RVP pro gymnázia se o algoritmizaci píše častěji. Již v charakteristice vzdělávací oblasti se s algoritmizací setkáme: „*Žák se seznámí se základními principy fungování prostředků ICT a soustředí se na pochopení podstaty a průběhu informačních procesů, algoritmického přístupu k řešení úloh a významu informačních systémů ve společnosti.*“ (Balada, 2007b, s. 63).

Algoritmizace je již také zařazena mezi cíle této vzdělávací oblasti: „*Vzdělávání v dané vzdělávací oblasti směřuje k utváření a rozvíjení klíčových kompetencí tím, že vede žáka k uplatňování algoritmického způsobu myšlení při řešení problémových úloh.*“ (Balada, 2007b, s. 63).

Co se týče vzdělávacího obsahu, zde je algoritmizace zařazena do bloku s názvem **Zpracování a prezentace informací**. Zde (Balada, 2007b, s. 65) je již konkrétně řečeno, jaké jsou očekávané výstupy a učivo, které má žák z této oblasti zvládnout:

očekávané výstupy

- ❖ *žák zpracovává a prezentuje výsledky své práce s využitím pokročilých funkcí aplikačního softwaru, multimediálních technologií a internetu*
- ❖ *žák aplikuje algoritmický přístup k řešení problémů*

učivo

- ❖ *publikování – formy dokumentů a jejich struktura, zásady grafické a typografické úpravy dokumentu, estetické zásady publikování*
- ❖ *aplikační software pro práci s informacemi – textové editory, tabulkové kalkulátory, grafické editory, databáze, prezentační software, multimedia, modelování a simulace, export a import dat*
- ❖ *algoritmizace úloh – algoritmus, zápis algoritmu, úvod do programování*

2.2.2 Odborné vzdělání

Pro odborné vzdělání vzniklo mnoho speciálních RVP. Každý obor má své RVP. Pro lepší orientaci jsou všechna tato RVP pro odborné vzdělání rozdělena do kategorií. Tyto dokumenty se obsahově velice liší. RVP pro vzdělávací obory s výučním listem se vůbec o algoritmizaci nezmiňují. Některá RVP pro obory zakončené maturitou s výukou algoritmizace či programování již počítají. Mezi takové obory patří například **Mechanik seřizovač** (RVP 2345L01).

RVP pro odborné vzdělání je členěno poněkud jinak. Zde se zajímáme o **Vzdělávání v informačních a komunikačních technologiích**. Mezi učivo je zde zařazena také algoritmizace a to bez další bližší specifikace. Pouze ve výsledcích vzdělání je řečeno: „*Žák ovládá principy algoritmizace úloh a sestavuje algoritmy řešení konkrétních úloh (dekompozice*

Teoretická východiska

úlohy na jednotlivé elementárnější činnosti za použití přiměřené míry abstrakce.“(RVP 2345L01, s. 44).

2.2.3 Shrnutí

RVP pro střední vzdělání se svým obsahem a zájmem o algoritmizaci velice různí. U učňovských oborů se s algoritmizací vůbec nepracuje. U maturitních oborů je to různé, ale úroveň, kterou by měli absolventi těchto škol po algoritmické stránce mít, není příliš vysoká. Jedinou výjimkou jsou gymnázia, kde by se žáci měli naučit i základy programování.

2.3 Mezinárodní srovnání

Po historické stránce máme společné základy výuky se Slovenskem, které bylo ještě do roku 1993 součástí tehdejšího Československa. Z tohoto důvodu zde také uvádím srovnání, jakým směrem se ubírá výuka na Slovensku v současné době.

Stejně jako v ČR tak i na Slovensku se v poslední době pokouší o modernizaci výuky. V obou zemích vychází tato novela ze stejného základu a tím jsou rámcové vzdělávací programy (na Slovensku Štátny vzdelávací program^a).

Jak české RVP přistupuje k výuce algoritmizace, již víme. Nyní se podíváme, jak se tohoto úkolu zhostili na Slovensku a jaké mají požadavky na žáky na základních školách v oblasti algoritmizace a algoritmického myšlení.

2.3.1 Slovensko

ŠVP SR je členěn do vzdělávacích oblastí. Algoritmizace spadá do vzdělávací oblasti **Matematika a práce s informacemi** a předmětu informatika.

Slovenské ŠVP (Blichová, Šestáková, 2008, s. 4) je mnohem konkrétnější, k algoritmizaci se vztahuje více cílů:

Cílem vyučování informatiky na 2. stupni ZŠ je zpřístupnit základní pojmy a techniky používané při práci s údaji a při tvorbě algoritmů a výpočetních procesů.

...

Výchovně-vzdělávací proces na 2. stupni základní školy směřuje k tomu, aby žáci

- ❖ rozuměli pojmu algoritmus a program (formální zápis automatizovaného zpracování údajů);*

^a ŠVP SR

Teoretická východiska

- ❖ *rozvítjeli si schopnost algoritmizovat zadaný problém, rozvítjeli si programátorské zručnosti, naučili se pracovat v prostředí běžných aplikačních programů, naučili se efektivně vyhledávat informace uložené na CD nebo na síti a naučili se komunikovat přes síť;*
- ❖ *získali schopnosti potřebné pro výzkumnou práci (tj. schopnost realizovat jednoduchý výzkumný projekt, zformulovat problém), rozvítjeli si formální a logické myšlení, naučili se různé metody řešení problémů,*

...

Další část tvoří obsahový a výkonový standart, který je rozdělen do pěti tematických okruhů. Algoritmizaci se věnuje okruh nazvaný **Postupy, řešení problémů, algoritnické myšlení** (Blichová, Šestáková, 2008, s. 4 - 9):

*V tematickém okruhu **Postupy, řešení problémů, algoritnické myšlení** se žáci seznámí se specifickými postupy řešení problémů prostřednictvím IKT. Seznámí se s pojmy jako algoritmus, program, programování. Největším přínosem tohoto okruhu bude to, že žáci získají základy algoritnického myšlení a schopnost uvažovat nad řešením problémů pomocí IKT. Naučí se uvažovat nad různými parametry efektivnosti různých řešení problémů, naučí se různé postupy a mechanismy při řešení úloh z různých oblastí.*

...

Postupy, řešení problémů, algoritnické myšlení

Pojmy:

- ❖ *postup řešení, etapy řešení problémů*
- ❖ *programovací jazyk, elementární příkaz, funkce, procedura, cyklus*
- ❖ *složitost řešení problému*

Vlastnosti a vztahy, postupy a metody:

- ❖ *v dětském programovacím prostředí řešení úloh s opakováním nějakých činností, seskupování částí řešení do procedur*
- ❖ *porovnání času trvání různých řešení problému*

...

Obsahový standard

- ❖ *Postup řešení, formální zápis řešení, etapy řešení problémů.*
- ❖ *Programovací jazyk, elementární příkaz, funkce, cyklus, procedura, parametry, proměnná, hodnota, přiřazení.*

Teoretická východiska

- ❖ *Složitost řešení problému.*

Výkonový standard

Žák

- ❖ *Dokáže zapisovat a interpretovat postupy do formálního zápisu (například zápis matematických algoritmů, algoritmus na jednoduché zašifrování textu).*
- ❖ *Demonstruje v dětském programovacím prostředí řešení úloh s opakováním nějakých činností, zapamatování výpočtů do proměnných, seskupování částí řešení do procedur.*
- ❖ *Dokáže porovnat čas trvání různých řešení problému.*

2.3.2 Shrnutí

Jak je již patrné z výše uvedených citací, všichni absolventi slovenských základních škol by měli mít mnohem lepší úroveň algoritmického myšlení, než jejich čeští vrstevníci.

Na Slovensku je úroveň všech žáků téměř stejná, zatímco v České Republice se tato úroveň různí žák od žáka, škola od školy. Výstupní úroveň je zcela v kompetenci vedení školy a jejich ŠVP a přístupu jednotlivých učitelů.

3.0 Výuka algoritmizace

3.1 Historie výuky algoritmizace na ZŠ v ČR

Historii výuky algoritmizace na ZŠ zmapoval RNDr. Tomáš Pitner, Dr. Jedná se o podpůrný text k přednášce z cyklu seminářů pro učitele informatiky, která proběhla 13. března 2000 na Fakultě informatiky Masarykovy univerzity v Brně.

Dr. Pitner se před více než deseti lety ve své přednášce snažil poukázat na klesající úroveň výuky algoritmizace, i proto začal svou přednášku otázkou: „*Historie: Cesta ze středu pozornosti na periferii zájmu?*“ (Pitner, 2000). Všechny své poznatky o výuce shrnuje v závěru: „*Výuka algoritmizace programování dosáhla “zlatého věku” zhruba před patnácti až deseti lety. Nyní se týká opět jen zájemců na určitých typech škol (gymnázia, průmyslové školy, učiliště s příslušným zaměřením, vyšší a vysoké školy). Na základních školách se “programuje” spíše ojediněle. Pro praktické programování se používá buďto specializovaných nástrojů určených k výuce (Logo, Baltazar/Baltik, Petr, Karel) nebo klasických profesionálních nástrojů (PASCAL, Visual BASIC, Delphi, JavaScript).*“

3.2 Cíle a metodika výuky algoritmizace

3.2.1 Co je cílem?

Cíle výuky algoritmizace na ZŠ formuloval Dr. Tomáš Pitner (Pitner, 2000):

- ❖ *Primárním cílem je naučit algoritmicky myslet, zformulovat zadání problému.*
- ❖ *Problém analyzovat nejdříve dekompozicí – rozložením na podproblémy.*
- ❖ *Nezbytné je též umět myšlenku dovést k formalizovanému návrh algoritmu, nejlépe v grafické podobě – vývojové diagramy, struktogramy..., evt. ve formě přesného slovního popisu za použití předem daných „obratů“, tj. vlastně povolených programových struktur.*
- ❖ *Přepsání formalizovaného návrhu do podoby programu je technická záležitost, nikoli hlavní cíl výuky.*
- ❖ *K samotným algoritmům nedílně patří, ale spíše až „ve druhém pořadí“, datové struktury (objekty).*
- ❖ *Vedlejším cílem (na nižších stupních dokonce hlavním cílem) je celkový rozvoj tvořivosti.*

Teoretická východiska

3.2.2 Jakou zvolit metodiku?

Dr. Pitner ve své práci navrhl dva možné přístupy k výuce algoritmizace. Prvním z nich je strukturovaný přístup, druhým objektově orientovaný přístup.

Strukturovaný přístup spočívá ve formulaci posloupnosti akcí, které povedou k vyřešení daného programu. Tato posloupnost vede ke vzniku algoritmu a následnému přepisu v programovacím jazyku. Primárním cílem tohoto přístupu je zvládnutí základních algoritmických struktur. Tj. sestavení algoritmu vedoucímu k vyřešení problému. Ten je tvořen posloupností příkazů, větvením, cykly a dalšími strukturami.

Z již dříve provedených výzkumů vyplývá, jak jsou žáci základních škol schopni chápat algoritmické postupy (Kopecká, 1998):

4. a 5. ročník	6. a 7. Ročník	8. a 9. ročník
Hotové programy	Větvení programu (1 podmínka)	Větší programy
Jednoduché příkazy	Jeden cyklus	Datové struktury
Sekvence příkazů	Proměnná	

Tabulka 1: úroveň chápání algoritmických postupů žáky ZŠ

Základem **objektově orientovaného přístupu** je uvědomění si, že všechno jsou objekty. Tyto objekty se chovají podle určitých vlastností. Řešení problému spočívá v nalezení objektů z reálného světa, definice jejich vlastností a vzájemných vztahů.

Své zkušenosti s výukou algoritmizace a výhody a nevýhody jednotlivých přístupů Dr. Pitner shrnuje takto:

Praxe ale obvykle ukáže, že tento přístup není nezbytný a začneme-li učit (kvalitně) strukturovaně, nic se nezmešká. Naopak, začít objektově předpokládá vyšší úroveň abstraktního myšlení. Za své hovoří také fakt, že většina výukových nástrojů pro algoritmizaci a programování je spíše strukturovaného zaměření, což je současně i škoda.

Obecně lze doporučit bez ohledu na objektovost/strukturovanost hned od počátku problémově orientovanou výuku s vysokým podílem samostatné práce.

Žáci se s pomocí známých či hotových prostředků snaží problém vyřešit, přicházejí postupně na to, co neumí, učitel jim to ve vhodnou chvíli odtajní a objasní a oni to vzápětí použijí v praxi. Výhodné je zpočátku vždy detailně rozebrat určité řešení včetně rozboru a napsání kódu (algoritmu) přesně na tabuli.

Teoretická východiska

A na základě svých zkušeností doporučuje následující postup:

Problém → jeho rozbor → samostatný pokus o řešení → špatný výsledek pokusu → seznámení s novým obratem (příkazem) → samostatný pokus o řešení s novým obratem → konfrontace jednotlivých řešení (společně na tabuli) → závěr, doporučení → definice vlastního problému a jeho samostatné či týmové řešení.

3.2.3 Desatero zásad výuky algoritmizace

Dr. Pitner specifikoval ve své práci zásady, kterými by se měl řídit vyučující algoritmizace:

***Zásada č. 1:** zvaž své vlastní síly – musíš nejen teoreticky zvládat algoritmizaci, ale i praktické programování a programovací prostředky*

***Zásada č. 2:** zvaž síly a motivaci žáků – algoritmizaci a programování uč jen zájemce, systematicky myslet uč všechny*

***Zásada č. 3:** ujasni si předem cíle, až podle nich zvol metodiku a nástroje*

***Zásada č. 4:** nespěchej, podobně jako v matematice je pro algoritmizaci třeba jisté úrovně abstraktního myšlení (8. – 9. třída)*

***Zásada č. 5:** kde nemůžeš pomoci, neuškod' – raději, než naučit špatně, neučit vůbec*

***Zásada č. 6:** vol adekvátní prostředky, které jdou bez zbytečností přímo k cíli*

***Zásada č. 7:** zdůrazňuj podstatné, odstiňuj od nepodstatného*

***Zásada č. 8:** nenechávej pochopení důležitých pojmů „na samostudium“*

***Zásada č. 9:** musíš-li v zájmu pochopení věci „klesnout pod odbornou úroveň“, udělej to*

***Zásada č. 10:** vlastní programátorské zkušenosti jsou nezbytným předpokladem kvalifikované výuky*

3.2.4 Chyby při výuce

Mezi základní chyby, kterých se může učitel algoritmizace dopustit a měl by se jich snažit vyvarovat, patří (Pitner, 2000):

- ❖ *Špatně nastavené cíle výuky (především přecenění sebe nebo žáků)*
- ❖ *Neodhadnutí úrovně abstraktního myšlení žáků (např. koncept proměnné v programovacím jazyce nepochopí každý žák 6. třídy)*

Teoretická východiska

- ❖ *Nevhodně zvolený nástroj (správný nástroj vede žáka tím, že mu dovolí právě to, co potřebuje – nic méně, nic více)*
- ❖ *Nevhodná metodika*
- ❖ *Naučení špatných, těžko odstranitelných návyků (přílišná orientace na maličkosti a technické detaily systémů, programování „try-and-see“ bez předchozí důkladné analýzy, nestrukturované programování, orientace na vizuální podobu a efektní uživatelské rozhraní výsledných programů)*

3.3 Možnosti výuky algoritmizace

Algoritmizace, pokud se vyučuje, je součástí předmětu informatika resp. výpočetní technika (jméno předmětu není jednotné, každá škola může mít jiný název).

Algoritmizaci může být věnována část hodin informatiky. Zde je jen a pouze na vyučujícím a na škole, zda budou považovat algoritmizaci za podstatný cíl jejich vzdělávacího působení na žáky a své cíle v této oblasti zahrnou i do ŠVP. V tomto případě je jednoznačně řečeno, co se mají žáci naučit, jaká je časová dotace aj.

Další možností je výuka v rámci hodin informatiky. V tomto případě může učitel považovat výuku algoritmizace za rozšiřující učivo, které se bude probírat pouze, pokud na to bude čas a vhodné podmínky.

To by byla výuka v rámci předmětu informatika. Dále se nabízí možnost výuky algoritmizace pouze pro zájemce formou zájmového kroužku na ZŠ nebo při jiné organizaci jako jsou například domy dětí a mládeže nebo různá občanská sdružení provozující zájmové kroužky.

Výuka formou zájmového kroužku má mnoho výhod, mezi největší patří zájem žáků o danou problematiku, menší počet dětí aj.

Poslední možností výuky algoritmizace je nevědomá výuka. V takovém případě se žáci učí algoritmicky myslet, aniž by si uvědomovali, že se to učí. Jedná se například o řešení fyzikálních úloh, kdy je nutné postupovat po jednotlivých dílčích krocích, aby se dospělo k výsledku. Dalším příkladem, kde se děti s algoritmy setkávají, aniž by si to uvědomovaly, je například kuchařka (zde se sice nejedná o přesný algoritmus, ale pouze o pracovní postup). Typickým příkladem, kdy děti musí vytvořit algoritmus, je pokud mají zpětně popsat, jak postupovali při nějaké činnosti (např. rýsování trojúhelníka při zadané straně a dvou úhlech).

4.0 Podpora výuky algoritmizace

V současné době je snaha o rozšíření výuky algoritmizace a programování na základních školách. Proto se také začíná programování a algoritmizace objevovat v učebnicích výpočetní techniky a rodí se projekty podporující rozvoj algoritmické gramotnosti nejen mezi žáky, ale i mezi učiteli.

4.1 Vývojové nástroje

Co se týká programového vybavení, existuje mnoho výukových i profesionálních nástrojů, jejichž výběr je zcela v režii učitele. Většina výukových programů je již léty prověřena.

Otázce volby vývojového nástroje se věnuje dr. Pitner (Pitner, 2000) a zamýšlí se nad otázkou, zda na ZŠ používat profesionální či výukové nástroje: „*Spor: výukové vs. profesionální nástroje. V profesionálních vytvoříme úplnou a funkční aplikaci, obvykle lze později znovupoužít zdrojový kód. Žáci později v praxi snadno přejdou na jiný profesionální nástroj. Je to ovšem problém základní školy? Výukové nástroje zbavují nutnosti znát většinu technických detailů daného prostředí, tím pádem ale nabízejí méně – nedovolí např. programem ovládnout všechny možnosti systému. Nelze v nich tedy vytvořit vše, co praxe žádá. Vadí to však na základní škole?*“

4.1.1 Profesionální nástroje

Profesionální nástroje jsou primárně určeny pro již zkušené programátory a jejich využití při výuce na ZŠ skýtá mnohé obtíže.

Mezi tyto nástroje patří zejména:

- MS Visual Basic
- Java
- C#

Výhody a nevýhody profesionálních nástrojů shrnul Dr. Pitner (Pitner, 2000):

- ❖ *Poměrně složité ovládání, nutnost zvládnutí základních programátorských principů předem.*
- ❖ *Aktuální verze obvykle náročné na hardware.*
- ❖ *Obvykle disponují vizuálním prostředím pro návrh uživatelského rozhraní, resp. objektové hierarchie aplikace.*
- ❖ *Výjimečně podporují i vizuální tvorbu strukturovaných algoritmů.*

Teoretická východiska

- ❖ *Většinou zahraniční provenience. Nabídky a nápovědy programovacího prostředí a syntaxe jazyka obvykle anglicky.*
- ❖ *Výsledkem je použitelný a dále šiřitelný program s profesionálním vzhledem.*
- ❖ *Často (většinou) lze i zdrojový kód použít mimo původní prostředí.*

4.1.2 Speciální prostředky pro výuku

Zkušení programátoři si postupem času uvědomili, že je třeba vyvinout nějaký výukový nástroj, který pomůže vychovat novou generaci programátorů. Z tohoto důvodu vznikly speciální výukové programy.

Mezi tyto nástroje patří:

- Karel
- Imagine Logo
- Baltík/Baltazar
- Petr
- Scratch

Výhody a nevýhody výukových nástrojů shrnul Dr. Pitner:

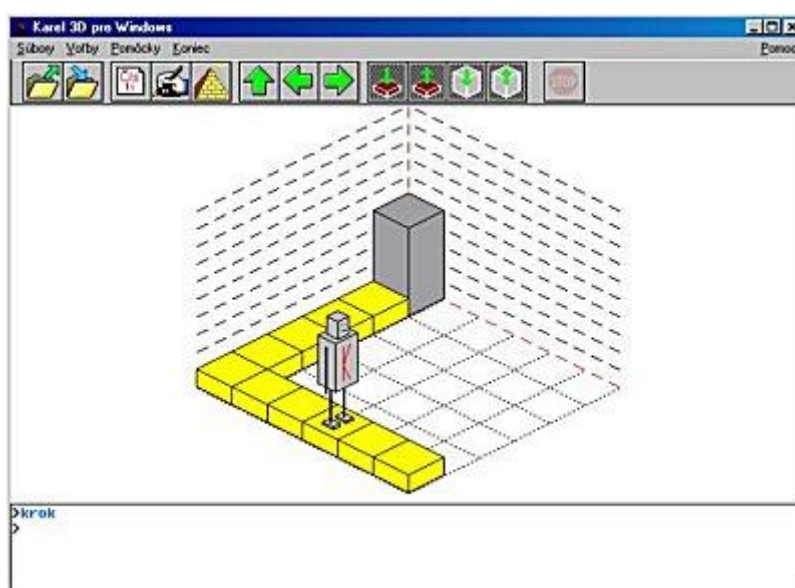
- ❖ *Omezené možnosti přispívají k jednoduššímu zvládnutí.*
- ❖ *Méně náročné na hardware.*
- ❖ *Dovolují rychlejší postup „k věci“ bez nutnosti vstřebávání „balastu“.*
- ❖ *Zaměřují se spíše na vizuální podporu strukturovaného programování (algoritmů) než na vizuální tvorbu objektových struktur nebo uživatelského rozhraní.*
- ❖ *Obvykle nižší až nulová cena (např. díky nemožnosti profesionálního použití).*
- ❖ *Často tuzemské provenience – letitá tradice české kvality, originálně v češtině a pro české prostředí, vytvářeno a testováno v českých verzích OS, lepší cena, licenční politika „laděná“ na české školy.*
- ❖ *K dispozici zdarma bývá (omezená) demoverze – např. Petr, Baltazar/Baltík*
- ❖ *Ostré verze mají někdy omezení komerčního využívání programů v systému vytvořených (Petr).*
- ❖ *Trpí uzavřeností do sebe. Téměř vždy jsou nekompatibilní s profesionálními nástroji i jinými výukovými prostředky, což znamená „doživotní“ závislost na výrobcích.*

Teoretická východiska

4.1.2.1 Karel

Karel je programovací jazyk určen pro výuku programování úplných začátečníků a svým prostředím a jednoduchostí ovládání je vhodný i pro děti. Současné verze umožňují i objektově orientované programování. V současné době existuje i 3D verze.

Oficiální představení Karla zní (Klíma): „*Karel představuje robůtka, který se pohybuje po městě a vykonává vaše příkazy. Dále je tady něco, co je v Karlovi označováno jako "město". Je to místnost obehnaná zdí, přes kterou se nemá šanci dostat, takže to víc připomíná vězení než město. Tohle město je tvořeno několika poli. Počet polí závisí na rozměrech města. A v tomto městě může Karel dělat skoro všechno, co se mu líbí, stavět vysoké zdi z cihel, pokládat značky, nebo se jen tak procházet ode zdi ke zdi.*“



Obrázek 3: Karel 3D pro Windows.

Převzato a upraveno z [www: <http://www.flatulent.szm.com/tchv/obr/karel3dwin.jpg>](http://www.flatulent.szm.com/tchv/obr/karel3dwin.jpg)

4.1.2.2 Imagine Logo

Programovací jazyk Logo byl navržen pro výuku algoritmického myšlení. „*Imagine Logo je kompletně objektový jazyk, který je řízen událostmi. Podporuje paralelní programování a má též propracovanou ideu obrázkových tvarů želv. Má některé nové prvky, které jsou typické pro programy pod Windows, např. překrývající se grafické plochy (jako listy papíru), tlačítka i s obrázky, posuvné lišty, textová pole, lišty tlačítek apod. Nechybějí ani multimédia, Internet a též vzájemná spolupráce Imagine-programů v síti.*“ (Vaniček, 2011)

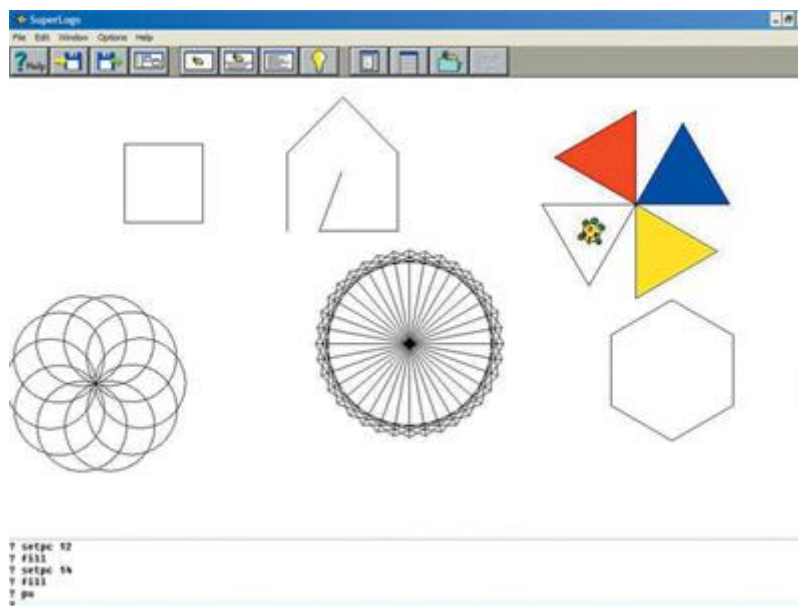
V současné době lze Logo použít nejen k výuce programování a algoritmizace: „*Imagine lze ale použít nejen jako prostředek k výuce programování, ale také na vytváření*

Teoretická východiska

multimediálních prezentací: můžeme vytvořit projekt složený z více stránek, na každé stránce se kromě textů, obrázků a zvuků mohou pohybovat i animované objekty.“ (Vaníček, 2011)

Základním prvkem je zde želva, která se pohybuje po písčité pláži. Želva svým ocáskem kreslí v písku stopu, pokud má ocásek zvednutý, pohybuje se bez tvorby stopy. Této grafice se začalo říkat želví grafika.

Základem jazyka je 200 slov, kterým se říká primitiva. Jedná se o příkazy, operátory a funkce.



Obrázek 4: Imagine Logo.

Převzato a upraveno z <http://www.rm.com/_RMVirtual/Media/Images/imaginelogoprimg3.jpg>.

4.1.2.3 Baltík / Baltazar

Baltík je výukový multimediální programovací a kreslicí nástroj pro děti a mládež. V současné době již existují čtyři verze tohoto populárního dětského programovacího nástroje. Prvním, v této řadě, byl SGP Baltazar. Následoval SGP Baltík 2, vhodný pro děti ve věku 4 – 8 let. Poté přišel SGP Baltík 3, který je určen dětem od 6. let. Zatím posledním je SGP Baltie 4 C#, který již podporuje objektově orientované programování, je trojrozměrný. Z tohoto důvodu je také vhodný až pro starší děti a to od 10. let věku.

*Program **SGP Baltík 3** je nadčasový výukový multimediální programovací a kreslicí nástroj pro děti a mládež. Baltík rozvíjí logické myšlení a tvořivost. Žáci se s pomocí Baltika naučí ovládat počítač, zvládnou základy práce s textovým editorem, s grafickým editorem, s multimédií, s elektronickou poštou, internetem a také základy algoritmizace a programování.*

Teoretická východiska

Beze zbytku pochopí základní pojmy jako počítač, program, data, soubory, složky a Internet. V Baltíkovi mohou žáci vytvářet multimediální prezentace, výukové programy i hry. (SGP Systems, a)

Baltík má tři úrovně. Úroveň **skládej scénu** funguje jako grafický editor. Další je **čaruj scénu**, kde se již děti učí používat příkazy a Baltíka ovládat. Poslední úrovní je **programuj**, kde z příkazových ikon sestavují posloupnost příkazů.



Obrázek 5: Baltík.

Převzato a upraveno z <http://www.zsalsova.cz/stare/i-centrum/baltik/baltik-1-04_soubory/image002.jpg>.

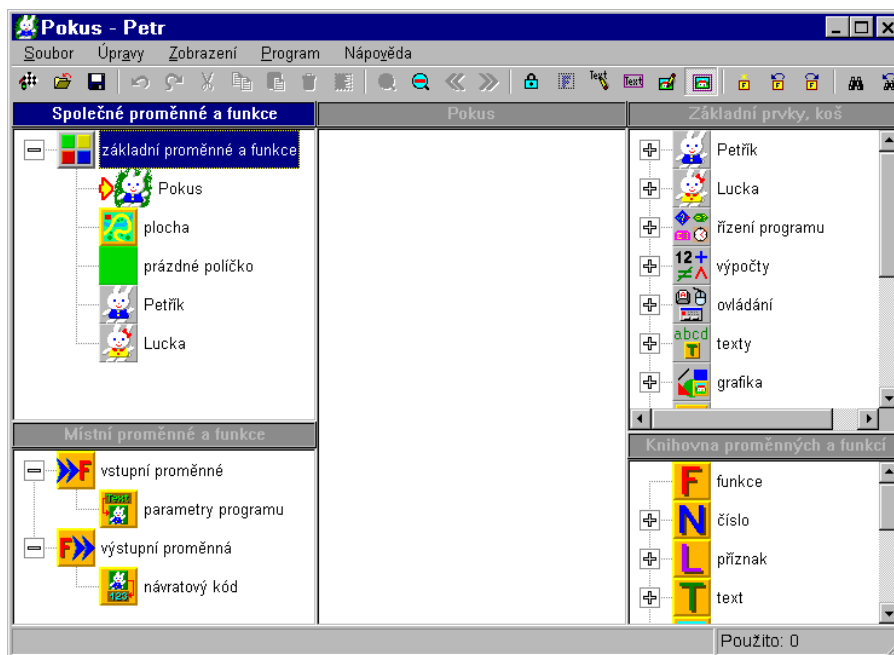
*SGP Baltie 4 C# je moderní tvůrčí objektově orientovaný výukový programovací nástroj založený na C#, DirectX a .NET, umožňující snadné programování 3D aplikací pro Windows (2000/XP/Vista/7). Je určen pro děti, mládež i dospělé. Podle věku a zkušeností si můžete vybrat vhodný režim pro svou práci - od úplného začátečníka (od 4 let) až po úplného profesionála (od 16 let). Program obsahuje také **Model Editor**, který umožňuje změnit hotové 3D modely (velikost, textury, barvy materiálů), vytvářet skupinové 3D modely, případně skládáním z malých kostiček vytvářet své vlastní jednoduché 3D modely.* (SGP Systems, b)

Nevýhodou Baltie 4 je nutnost překladu, který je poměrně zdlouhavý a děti během těch pár minut ubírají svou pozornost jiným směrem, což působí rušivě.

4.1.2.4 Petr

Petr je český programovací nástroj pro začínající programátory. Jeho ovládání je jednoduché a intuitivní, aby se v něm i začátečník snadno orientoval. „Petr nabízí příjemné grafické prostředí a přitom v něm lze dosáhnout velice dobrých výstupů. Naprogramujete tak jednoduché programy a nástroje, ale také propracované trojrozměrné hry.“ (Kraus, 2011)

Teoretická východiska



Obrázek 6: Petr.

Převzato a upraveno z <http://www.islandsoft.cz/img/petr_tutorial_pokus.gif>.

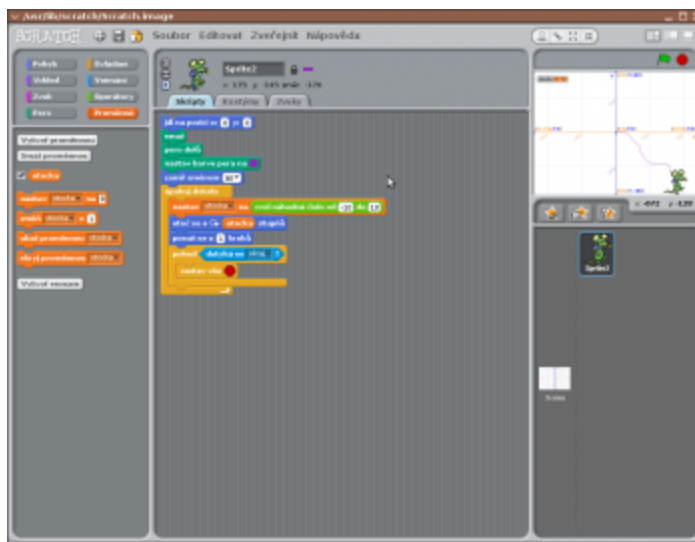
4.1.2.5 Scratch

„Scratch je jednoduchý výukový volně šiřitelný programovací jazyk vyvinutý na Massachusetts Institute of Technology umožňující programovat za pomoci k sobě přiřazených bloků příkazů. Je vhodný pro využití nejen doma, ale i ve škole či při zájmové činnosti dětí orientované na práci s počítačem. Dnes je rozšířen po celém světě. Určen je mládeži od šesti do šestnácti let bez předchozích programovacích zkušeností. Prostředí Scratch je intuitivní a snadné na ovládání. Lze ho zdarma stáhnout z portálu <http://scratch.mit.edu/>.“ (Čechová Humpolcová, 2012).

Scratch je postaven na principu vývojových diagramů, kde uživatel z nabídky vybírá stavební bloky, které tvoří program. Jednotlivé bloky se skládají pod sebe nebo do sebe a tudíž je zajištěna syntaktická správnost programu.

Stavební bloky jsou pro lepší orientaci rozdělené do osmi kategorií, například pohyb, vzhled, ovládání, proměnné aj.

Teoretická východiska



Obrázek 7: Scratch.

Převzato a upraveno z < <http://i.iinfo.cz/images/351/lang08-3-prev.png>>.

Jak je vidět na obrázku, vývojové prostředí je přehledné a jednoduše ovladatelné. V levé části se nacházejí bloky, uprostřed se nachází prostor pro tvorbu programu a pravou část tvoří scéna.

4.2 Učebnice zabývající se algoritmizací/programováním

V současné době je na českém trhu řada učebnic informatiky (výpočetní techniky) pro základní školy. Po obsahové stránce se tyto učebnice velmi různí. Jakési minimum je v každé učebnici, ale způsob jeho zpracování je různý a oblasti dalšího vzdělávání nad toto minimum se také velmi různí.

Z dostupných učebnic se pouze jediná věnuje výuce programování a to **Informatika pro základní školy a víceletá gymnázia, 3.díl** jejímiž autory jsou Vaníček a Mikeš.

Učebnice pro střední školu již nabízejí mnohem větší výběr. Zde se již mnohem častěji setkáváme s učebnicemi, které se výuce programování věnují. Dokonce existuje i kniha zabývající se přímo výukou algoritmizace (Pšenčíková Jana, Algoritmizace) a svou koncepcí je také vhodná pro středoškoláky.

Pokud se podíváme, jak je tomu například na Slovensku, zjistíme, že tam je situace lepší. Dokonce zde existuje i učebnice z řady Tvořivá informatika, přímo určená pro základní školy a věnující se výhradně programování (Zošit programovania od autorů Andrej Blaho, Ivan Kalaš).

Teoretická východiska

4.2.1 Informatika pro základní školy a víceletá gymnázia, 3. díl

Jedná se o učebnici z třídílné řady učebnic. Programování je v této učebnici popisováno prostřednictvím výukového programovacího prostředí Imagine Logo.

Programování se zde věnuje hned první kapitola. Obsahově se jedná o úvodní seznámení s želvičkou a prostředím. Dále se žáci učí jednoduché příkazy pro kreslení, které si ihned mohou vyzkoušet na celé řadě úloh. Poté se seznámí s cykly prostřednictvím příkazu „opakuj“, který jim může velmi zjednodušit například nakreslení čtverce, šestiúhelníku... Následují hrátky se vzhledem vykreslované čáry. Tím končí úvodní seznámení a přiblížení si práce se želvičkou. Dále již začíná trochu klasičtější programování, kdy žáci učí želvu nové příkazy, seznamují se s událostmi atd. Závěr kapitoly uvádí mnoho opakovacích úloh a přináší přehled probraných příkazů i s popisem jejich funkce.

K této učebnici existuje také metodická příručka (Metodická příručka – Informatika pro základní školy a víceletá gymnázia, 3. díl) od doc. Vaníčka, která poradí i nepříliš zběhlému učiteli v programování a jeho výuce, jak k výuce přistupovat, kde najde nějaké další zdroje informací a samozřejmě i správná řešení všech úloh.

Součástí je také CD-ROM s příklady z učebnice a dalšími materiály.

4.3 Projekty zabývající se algoritmizací na ZŠ

V současné době je snaha o rozšíření výuky algoritmizace a programování na základních a středních školách. Z tohoto důvodu také vznikají projekty, které mají podpořit a zvýšit prestiž výuky algoritmizace nejen mezi žáky, ale hlavně mezi učiteli výpočetní techniky. Jedná se například o projekt PROŠ, Tvořivý učitel tvořivé informatiky a Mladí programátoři.

4.3.1 Projekt PROŠ

Projekt Pedagogické fakulty Univerzity Palackého (PdF UP) v Olomouci přinese základy programování do škol. Informace o tomto projektu byly čerpány od Straky (Straka, 2007) a Hronové (Hronová, 2011).

„Výstupem projektu, který se jmenuje Programování do škol, a do něhož se přihlásila gymnázia ve Šternberku, Litovli, Uničově, Hranicích, Jeseníku a gymnázium P. I. Čajkovského v Olomouci, bude ucelený metodický materiál. Ten v budoucnu poskytneme všem školám – tedy nejen těm, které nyní na projektu participují. Základy programování tak bude moci do své výuky zahrnout každá škola,“ řekl Mgr. Milan Klement, proděkan pro informační a vzdělávací technologie PdF UP a zároveň hlavní manažer projektu.

Teoretická východiska

V první fázi projektu budou vyškoleni metodici výuky na školách zapojených v projektu. Závěrečnou fází bude vznik multimediální učebnice programování a metodická příručka. Od září 2012 by se měl celý projekt začít testovat v praxi. Po testování a následném vyhodnocení bude celý výukový celek nabídnut všem školám. Projekt by také měl pomoci pedagogům, pro které bude připraveno speciální školení.

V rámci projektu by měl vzniknout metodický portál, který bude obsahovat školící materiály ve formě e-learningu, další interaktivní výukové materiály, metodické postupy aj.

4.3.2 Tvořivý učitel tvořivé informatiky

„Cílem projektu Tvořivý učitel bylo a nadále je nabídnout učitelům základních a středních škol kvalitní a praxí na školách prověřené kurzy výuky programování, dát učitelům do rukou metodické materiály a pomáhat jim i nadále konzultacemi a různými podpůrnými aktivitami. Naším záměrem je dosáhnout v oblasti ICT a programování, aby i prostřednictvím těchto hodin výuky odcházeli z našich škol děti vzdělané komplexně, schopné řešit problémy, vidět souvislosti, získávat informace a přistupovat k úkolům tvořivým způsobem.“ uvedlo občanské sdružení TIB (TIB, 2008).

Občanské sdružení bude v rámci tohoto projektu zdarma poskytovat učitelům ze ZŠ a SŠ pět různých kurzů. Jedná se o kurzy tvořivé informatiky a algoritmizace a programování na různých úrovních.

4.3.3 YOUNG DEVELOPERS

Jedná se o projekt vytvářející model moderní výuky programování na ZŠ a SŠ. Cílem projektu je vzdělávací systém rozvíjející programátorské schopnosti a vědomosti žáků, hledání talentovaných jedinců a následná pomoc při jejich profesním růstu.

Tento projekt není určen pouze učitelům základních škol, ale i žákům a vyučujícím na VŠ, kteří připravují budoucí pedagogy.

Metodika výzkumu

Při výzkumné práci se zpravidla užívají vědecké metody, které pomáhají shromáždit a následně i zpracovat získaný materiál. Při řešení problému je nutné nejprve vybrat vhodnou vědeckou metodu. Typ problému určuje výběr metody. Většinou se vymezují dvě velké skupiny metod:

- ❖ metody teoretického výzkumu,
- ❖ metody experimentálního výzkumu.

Problém, kterým se zabývá tato práce, patří do skupiny metod teoretického výzkumu. V této práci je použito více výzkumných metod z dané skupiny. Jedná se zejména o následující metody:

- ❖ obsahová analýza
- ❖ metoda rozhovoru (individuální rozhovor)
- ❖ pozorování

1.0 Obsahová analýza

Jane Kronicková ve své práci (Kronick, 1997) definuje obsahovou analýzu následovně: *„Obsahová analýza, jako všechny výzkumné nástroje, předpokládá existenci logicky vytvořené teorie ve formě obecných tvrzení a odvozených hypotéz. Tyto hypotézy jsou výroky o vztahu mezi proměnnými. Technika obsahové analýzy umožňuje měření těchto proměnných v psaném textu. Psaný text je v obsahové analýze považován za totéž, co přináší výzkumné šetření, neboť text jsou vlastně data, jež umožňují měření proměnných, které jsou pro daný výzkum zásadní. Cílem obsahové analýzy je extrakce proměnných z textu v měřitelné podobě.“*

2.0 Metoda rozhovoru

Hošpesová (Hošpesová, 2011) popisuje metodu rozhovoru následovně:

Metoda rozhovoru (interview, dotazování), je založena na přímém kontaktu, důležitá schopnost navázat kontakt, komunikovat. Liší se od běžného rozhovoru svou cílovou orientací a pečlivou přípravou.

Teď hlavně v psychologii, její klinické praxi, v psychiatrii i pedagogice (kde je často ke škodě věci rozhovor nahrazován dotazníkem – protože dotazník má svou formou blízko ke standardizovanému rozhovoru.

V pedagogických výzkumech používán jako doplňková metoda při pozorování a experimentu.

Metodika výzkumu

Rozhovor má mnoho druhů. Jedná se například o individuální či hromadný rozhovor, standardizovaný nebo nestandardizovaný či polostandardizovaný rozhovor.

3.0 Pozorování

Pedagogické pozorování je nejrozšířenější metodou získávání dat. Je definováno jako „sledování smyslově vnímatelných jevů, zejména chování osob, průběhu dějů aj.“ (Průcha, Walterová, Mareš, 2003).

Pozorování lze klasifikovat podle nejrůznějších hledisek. Například podle času rozlišujeme krátkodobé a dlouhodobé pozorování. Podle toho, zda se pozorovatel přímo setkává s objekty pozorování, rozlišujeme pozorování vlastní a nevlastní. Krátkodobé pozorování je definováno jako „pozorování, která netrývají zpravidla déle než jednu vyučovací jednotku.“ (Chráska, 2007). Vlastní pozorování znamená, že se pozorovatel účastní pozorování přímo a setkává se s předmětem pozorování.

V tomto výzkumu bude využito pouze krátkodobé vlastní pozorování. V praxi bude toto pozorování realizováno jako následek při výuce programování a to v rozsahu jedné vyučovací hodiny.

4.0 Analýza úloh

Nejprve bude provedena analýza každé úlohy z Bobříka informatiky.

Poté budou jednotlivé úlohy vyhodnoceny dle četnosti jednotlivých odpovědí. Jedná se o **snadné** otázky, kde správné odpovědi dosahovaly alespoň padesáti procent. Další skupinou jsou úlohy, kde sice dominovala správná odpověď, ale její četnost byla méně, než poloviční. V tomto případě se jedná o **středně obtížné** úlohy. Poslední skupinou jsou **těžké** úlohy, které se dále dělí na dvě podskupiny A a B. **A** podskupinu tvoří matoucí otázky, kde dominovala jedna nesprávná odpověď, přičemž procentuální rozdíl nejčtenější odpovědi a další v pořadí je minimálně deset procent. Další podskupinu **B** tvoří úlohy, kde dominovaly dvě odpovědi, z toho jedna byla správná.

Následuje analýza jednotlivých skupin úloh a hledání jejich společných rysů.

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

1.0 Popis výzkumného vzorku

Výzkumný soubor je tvořen vzdělávacími institucemi působícími na území města České Budějovice a zabývajícími se vzděláváním dětí ve věku žáků druhého stupně základní školy.

Na území města České Budějovice je dle rejstříku škol a školských zařízení MŠMT v současné době pětadvacet institucí pro vzdělávání dětí ve věku žáků druhého stupně ZŠ (viz. Příloha 1: Seznam škol ČB). Z toho šest je nižších gymnázií a zbylých devatenáct je základních škol. Z výzkumu byly vyřazeny speciální základní školy (tři).

Tabulka 2 : Počet vzdělávacích institucí v ČB

Základní školy	16
Gymnázia	6
Celkem	22

Výzkumu se zúčastnilo osmnáct institucí, z toho dvanáct základních škol a všech šest gymnázií. Ostatní základní školy projevily neochotu spolupracovat.

2.0 Výsledky

Během výzkumu bylo zjištěno následující:

Tabulka 3: Výsledky výzkumu A

Instituce vyučující algoritmizaci	5
Instituce nevyučující algoritmizaci	13
Ostatní instituce	4
Celkem institucí	22

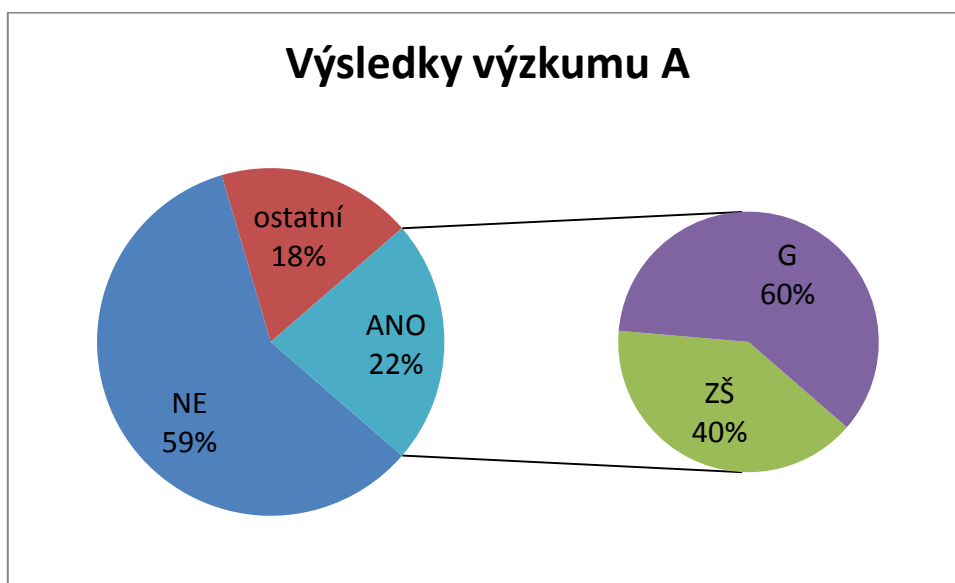
A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

Jak vyplývá z výše uvedené tabulky (tabulka 3), celkem bylo dotázáno dvaadvacet institucí. Z tohoto počtu odpovědělo osmnáct institucí. Bylo zjištěno, že se algoritmizace vyučuje pouze na pěti institucích.

Tabulka 4: Rozbor institucí vyučujících algoritmizaci

Základní školy	2
Gymnázia	3
Celkem institucí	5

Tabulka 4 ukazuje, že pouze dvě z těchto institucí jsou základní školy. V ostatních případech se jedná o gymnázium.



Obrázek 8: Graf - výsledky výzkumu A

3.0 Rozbor přístupu k výuce

Závěrem výzkumu je, že výuka algoritmizace a programování v současné době není na základních školách v Českých Budějovicích příliš rozšířená. Na gymnáziích je již situace lepší.

Výuka algoritmizace a programování na gymnáziích probíhá v různých jazycích. Jedná se zejména o **IMAGINE LOGO**, **BALTÍK**, či dokonce i výuka **algoritmizace** jako takové. Tato výuka většinou probíhá v kvartě (odpovídá deváté třídě ZŠ). Ve většině případů je to pouze volitelný seminář. Na jednom gymnáziu probíhá dokonce volitelný seminář, kde se učí programování v **PASCALu**.

Hlavní těžiště zájmu této práce je výuka algoritmizace na základních školách, proto tyto školy a jejich metody výuky budou rozebrány blíže.

3.1 Škola A

Výuka programování probíhá v **IMAGINE LOGO** a **SCRATCH**. Výuce programování se zde věnují již pět let. Výuka programování je zaměřena na objektově orientované programování. Do loňského roku výuka probíhala v Imagine Logu. Loni začali experimentovat se Scratchem, ke kterému v letošním roku přešli. Výuka ve Scratchi je jednodušší a pro žáky lépe pochopitelná.

3.1.1 ŠVP

Analýzou ŠVP bylo zjištěno následující:

Výuka programování zde probíhá v rámci povinného předmětu informatika, který se vyučuje v pátém a devátém ročníku s časovou dotací jedna hodina týdně. Během této výuky se věnují pouze základům práce s počítačem.

Dále na škole probíhá volitelný předmět informatika, který mohou děti navštěvovat od šestého až do devátého ročníku. V rámci tohoto předmětu se vyučuje i programování. Programování se věnují v druhém pololetí devátého ročníku. Dle ŠVP se zde učí základy objektově orientovaného programování. Výstupem je: „*napiše jednoduchý program v aplikaci pro objektově orientované programování, orientuje se v základních pravidlech programování.*“ (Fuková, 2007, s. 173). Během této výuky si děti rozvíjí i algoritmické myšlení:

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

„Prostřednictvím praktických cvičení si dále rozvíjejí algoritmické myšlení a jsou vedeni k systematickému přístupu při řešení problémů.“ (Fuková, 2007, s. 480).

3.1.2 Náslech

Náslech proběhl v deváté třídě při výuce volitelného předmětu. Pro žáky to byla teprve devátá hodina výuky programování. Skupina byla tvořena dvanácti dětmi.

Tato škola spolupracuje s pedagogickou fakultou JU a probíhá zde průběžná praxe studentů. Tato praxe probíhá také během tohoto předmětu. Hodinu tedy nevedl učitel, ale studentka pedagogické fakulty.

Během hodiny měli žáci pracovat na více úlohách. Na těchto úlohách si nejprve měli zopakovat, co již se naučili o struktuře programu a datových strukturách a následně i napsat svůj vlastní program podle předem daných kritérií.

První úlohou byl program Balónek. Úloha spočívala v nalezení chyby ve struktuře programu. Znění úlohy bylo: „Balónek letí pouští, doletí na kaktus, ale nepraskne, proč?“ Žáci měli zjistit, že podmínku, která testuje, zda je balónek u kaktusu byla vložena až za cyklus, který zajišťoval let balónku. Řešením tedy bylo vložení této podmínky do těla cyklu jako jeho poslední příkaz. Během této úlohy bylo vidět, jak se děti orientují ve struktuře programu, chápou posloupnost jednotlivých příkazů a datových struktur a rozumí danému zadání. Rychlost, s jakou žáci tuto chybu odhalili, byla vcelku rozmanitá. Ti nejrychlejší se hned po spuštění programu zorientovali v kódu a chybu odhalili a opravili doslova během několika málo vteřin. Naopak ti nejpomalejší pouze zkoušeli metodou pokus omyl, zda se jim daný problém podaří vyřešit, což se v několika případech ani nepodařilo a namísto opravy celý program ještě více poškodili.

Nalezením řešení problému ještě práce s touto úlohou neskončila. Cyklus, který zajišťoval let balónku, byl definován jako nekonečný. Nyní paní učitelka ukázala žákům druhou variantu tohoto cyklu, který již byl podmíněný. Při běhu obou variant programu se žáci zaměřili na objekt balónku a co se s ním děje, když v programu praskne. Ve variantě s nekonečným cyklem („opakuj dokola“) bylo vidět, že balónek se stále pohybuje, i když v programu už není. V případě podmíněného cyklu („opakuj dokola dokud...“) se po prasknutí balónku v programu zastaví i objekt balónek. Žáci sami usoudili, že by se měl využívat podmíněný cyklus, který dále nezatěžuje paměť.

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

Práce na této úloze žákům trvala čtvrt hodiny. Žáci si zopakovali strukturu programu, orientaci v kódu, větvení programu a naučili se rozdíl mezi podmíněným a nekonečným cyklem.

V druhé části měli žáci napsat program Kouzelník. Scéna byla tvořena kouzelníkem v lese. Žáci měli předem dané tři objekty – dva kouzelníky v různých pozicích a obláček dýmu. Na základě těchto podkladů byl nastíněn problém: *„Napište program, který bude tvořen čtyřmi objekty, na scéně bude současně vždy pouze jeden. Výstupem bude čarující kouzelník, který se přemění. V každé fázi své přeměny by měl něco říkat – zaklínadlo.“* Žáci na tuto práci měli deset minut.

Řešením byl jednoduchý nevětvený program, přibližně na deset řádek. V důsledku měli děti pouze vymyslet posloupnost příkazů se zadaným počátečním vzhledem, daným výstupem a přidat do programu jeden objekt z galerie objektů.

Postup řešení byl opět velmi individuální. Žáci s více rozvinutým algoritmickým myšlením a zájmem o danou problematiku pracovali systematicky a základní řešení měli již za pár minut. Poté si pouze hráli s detaily, jako například volba čtvrtého objektu, vymýšlení originálnějšího zaklínadla. Další skupinka řešitelů měla mírné problémy se správnou volbou příkazů a jejich posloupností, ale práci ve stanoveném čase zvládli. Poslední skupinka pouze zkoušela vkládat příkazy, a pokud daný příkaz nevykonal, co potřebovali, opět jej smazali. Těchto pár žáků daný problém nestihlo ve stanoveném čase dořešit. Poté úlohy uložili, promíchali se u počítačů a zkoušeli funkčnost. Na závěr ve dvojicích opravovali cizí nefunkční programy.

Na těchto dvou úlohách bylo již vidět zájem některých jedinců o programování, u některých algoritmické myšlení. Všichni žáci si rozvíjeli tvořivost, algoritmické a logické myšlení, naučili se vyhledávat a opravovat chyby atd.

3.2 Škola B

Výuka programování probíhá v **BALTÍKovi**. Tato škola se velmi intenzivně věnuje výuce programování již u dětí velmi útlého věku. V současné době probíhá výuka Baltika v rámci nepovinných kroužků již u žáků první třídy (na přání rodičů). Tato škola má s výukou programování již dlouholeté zkušenosti. Programátorské kroužky probíhají již osmým rokem a volitelné předměty již čtvrtým rokem. Tato škola nějakou dobu vyučovala programovací jazyk

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

Karel, chvíli experimentovali s Logem a nakonec si vybrali Baltíka, ve kterém výuka probíhá již několik let.

Škola je zapojena do projektů **Tvořivá informatika s Baltíkem** a **Mladí vývojáři**.

Její žáci se pravidelně účastní programátorských soutěží, ve kterých se velmi často umisťují i na postu nejvyšším. Škola se také podílí na organizování soutěží (celostátní kolo mezinárodní soutěže BALTIE 2011 aj.). Nedílnou součástí jsou i školní soutěže programování. Dle paní učitelky se základna škol účastnících se soutěží spíše snižuje. Pravidelně se účastní žáci z pěti škol.

Rozsah výuky informatiky je v rámci povinných hodin v třetím až osmém ročníku s dotací jedna hodina týdně a v devátém ročníku dvě hodiny týdně. Výuka informatiky také probíhá v rámci povinně volitelných a nepovinných předmětů v třetím až pátém ročníku s dotací jedna hodina týdně a v osmém i devátém ročníku s dotací dvě hodiny týdně. Tyto předměty se věnují pouze výuce programování. Na prvním stupni (výběr ze dvou předmětů) se v průměru hlásí na programování celá polovina žáků. Na druhém stupni (výběr z osmi předmětů) je to třetina všech žáků.

Programování v Baltíkovi (ZŠ L. Kuby, 2011):

*„Na naší škole v současné době pracujeme s **Baltíkem 3** v několika kroužcích pro 1. stupeň a v hodinách informatiky pro 2. stupeň a též v hodinách povinně volitelného předmětu Baltík pro žáky 3., 4. a 5. tříd.*

Baltíka znají již děti od druhé třídy a mohou se zapojit do kroužků, které probíhají několikrát týdně v odpoledních hodinách.

Činnosti všech kroužků směřují ke zvládnutí programovacího jazyka Baltík 3 a posléze Baltík 4, kde se již programuje v prostoru.

*Žáci vytvářejí **vlastní programy** (hry, příběhy...), připravují se na různé **programovací soutěže**, ve kterých mohou změřit své síly s dětmi z jiných základních škol i gymnázií v České i Slovenské republice.*

*V **Baltíkovi 4** programují zájemci v kroužku či v povinně volitelných hodinách programování.“*

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

Výuka probíhá na základě osnov^a přejatých ze serveru Tvořivá informatika a pomocí dalších výukových příkladů, speciálně vytvořených pro výuku na této škole. Pro výuku Baltíka 4 si na škole vytvořili vlastní výukové lekce. Výuka v Baltíkovi 4 má dle paní učitelky nevýhodu z důvodu dlouhého překladu programu (1 minuta), kdy žáci přerušují svou koncentraci a soustředěnost na programování.

V rámci spolupráce s rodiči škola také pořádá Vánoční posezení s Baltíkem a Závěrečné posezení s Baltíkem, kde malí programátoři předvádí své nejlepší programky.

3.2.1 ŠVP

Tato škola zahrnuje programování i do svého ŠVP (Tlapáková, 2010, s. 3):

Dlouhodobé projekty

Projekty dotvářejí obsah vzdělávacího procesu, zvyšují jeho efektivitu, motivují žáky a podporují spolupráci pedagogů a žáků. Škola je zapojena v celorepublikových projektech Tvořivá informatika s Baltíkem (TIB) a Mladí vývojáři. Cílem obou těchto projektů je s pomocí programování v programovacích nástrojích pro mládež Baltík 3 a Baltík 4.NET nastartovat a rozvíjet u dětí s hlubším zájmem o informatiku tyto procesy:

- ❖ *tvořivost*
- ❖ *cílevědomost*
- ❖ *sebevzdělávání (samostatné hlubší studium oblasti svého zájmu)*
- ❖ *samostatnou komunikaci*
- ❖ *výchovu pro týmovou spolupráci*
- ❖ *rozvíjení mimoškolní tvůrčí a komunikační aktivity*
- ❖ *sebepoznávání*
- ❖ *usnadnění hledání svých zájmů, potřeb a svého místa na trhu práce*

Aby se děti mohly porovnat s žáky ostatních škol, zúčastňujeme se pravidelně soutěží, organizovaných několikrát ročně v rámci celé republiky.

^a Osnovy pro začátečníky a pokročilé, dostupné na

http://www.baltik.wz.cz/vyuka_b3/vyuka_zacatecnici.pdf a

http://www.baltik.wz.cz/vyuka_b3/vyuka_mirne_pokrocili.pdf

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

Dokonce v rámci klíčových kompetencí pro jednotlivé oblasti je uvedeno (Tlapáková, 2010, s. 182): „na základě pochopení pak tyto informace propojuje a užívá při tvorbě obrázků, prezentací, www stránek a vlastních jednoduchých programů“.

3.2.2 Náslech

Náslech proběhl ve čtvrté třídě při výuce povinně volitelného předmětu programování. Skupina byla tvořena čtyřicetimi dětmi. Část dětí již chodí i do kroužku programování. Skupina byla tudíž velmi diferenciovaná. Děti by se daly rozdělit do tří skupin dle svých programátorských dovedností. Tato diferenciací klade vysoké nároky na práci pedagožky, jejíž přístup při hodině byl velmi individualizovaný. Tyto vysoké nároky jsou kladeny také na přípravu na každou vyučovací hodinu, orientaci v dané problematice atd.

Během náslechu děti dodělávaly úlohu Medvídci školního kola soutěže Mladý programátor 2012. Všichni žáci řešili stejnou úlohu. Tato úloha se dala řešit již se základními znalostmi práce v Baltíkovi. Touto formou to také někteří žáci řešili. Pokročilejší již využívali složitější programové konstrukce a využívali nástroje, které jim Baltík nabízí, jako například pomocníky (metody). V úloze bylo možné využít pro zjednodušení cykly, pomocníky (metody), souřadný systém aj.

Čas potřebný pro vyřešení daného problému byl velmi rozdílný. Někteří žáci byli hotovi méně než za jednu vyučovací hodinu, zatímco ti nejpomalejší to nestihli vyřešit ani za dvě hodiny. Nejslabší žáci dokonce pracovali ve dvojicích. Žáci, kteří byli již s prací na této úloze hotovi, dělali další soutěžní úlohy.

Zajímavostí je, jakým způsobem nechávají žáci kouzelníka Baltíka procházet scénou. Scéna je tvořena šachovnicí. Všichni se při výkladu učili procházet toto pole odspodu po řádcích zleva doprava. Při řešení této soutěžní úlohy někteří procházeli pole po sloupcích a jeden žák dokonce ve tvaru písmene L. To je také důkaz toho, jak programování může rozvíjet dětskou tvořivost.

3.2.3 Zájmový kroužek

Zájmový kroužek probíhá již od první třídy.

V první třídě se zaměřují pouze na seznámení s Baltíkem. Při práci využívají zejména režim skládat scénu. Všechny úlohy, na kterých pracují, jsou umístěny na školním moodlu. Tyto úlohy jsou volně přístupné a žáci na nich mohou pracovat také doma se svými rodiči. Když se

A. Kvantitativní výzkum výuky algoritmizace na ZŠ v ČB

naučí pracovat v režimu skládat scénu, přecházejí do režimu čarovat scénu a již se učí základy programování.

V dalších ročnících se již věnují prohloubení svých programátorských schopností a připravují se na soutěže.

V současné době není úplná návaznost kroužků z jednotlivých ročníků. Z tohoto důvodu se ve čtvrté třídě v povinně volitelném předmětu potkají žáci různé úrovně programátorských schopností a dovedností. Jsou tam žáci, kteří mají za sebou již dva roky kroužku, nebo také úplní začátečníci. V osmé a deváté třídě se setkává skupina žáků, která je velmi členitá – pouze z kroužků, z kroužků i povinně volitelného předmětu ze čtvrté a páté třídy s úplnými začátečníky.

B. Přehled mimoškolní výuky algoritmizace v ČB

B. Přehled mimoškolní výuky algoritmizace v ČB

V Českých Budějovicích působí mnoho organizací, které pořádají zájmové kroužky. Pro děti ve věku žáků druhého stupně je těchto kroužků celá řada. Většina kroužků je sportovních, jazykových, řemeslných aj. Kroužků zaměřených na výuku práce s počítačem se pořádá již mnohem méně.

Výuce programování se věnuje pouze jediný kroužek - **ROBOZOR**. Ani u tohoto kroužku se nejedná primárně o výuku programování a algoritmizace, ale o konstrukci robotů: „Učíme se zde konstruovat roboty jak po mechanické, elektronické i softwarové stránce.“ (DDM ČB, 2010). Po mechanickém sestavení robota se mu pokoušejí vdechnout život pomocí řídicí jednotky a algoritmů: „Výuka směřuje k praktickému zkoušení a simulování algoritmů používaných v jednoduchých robotech.“ (DDM ČB, 2010).

V kroužku využívají různé robotické stavebnice, jako například Merkur, Lego MindStorms, také elektronické stavebnice UST aj.



Obrázek 9: MERKUR: Robotický Slídlil ALFA.

Převzato a upraveno z <http://www.merkurtoys.cz/editor/image/produkty2/obrazek_931.jpg>.

Děti z druhého stupně ZŠ pracují zejména s robotickou stavebnicí Lego MindStorms. Programování těchto žáků probíhá v jazyce C (NQC).

B. Přehled mimoškolní výuky algoritmizace v ČB



Obrázek 10: LEGO MINDSTORMS NXT 2.0.

Převzato a upraveno z <<http://www.robotobchod.cz/670-1221-thickbox/lego-mindstorms-nxt-20.jpg>>.

Tento kroužek není určen všem dětem. Je sice vhodný pro děti od deseti let, ale podmínkou jsou znalosti práce s počítačem a základy programování. Není tudíž určen naprostým začátečníkům, kteří by touto formou chtěli s programováním teprve začít.

Do kroužku aktuálně chodí deset dětí, což představuje maximální možnou kapacitu.

C. Zmapování celorepublikové výuky algoritmizace na ZŠ

C. Zmapování celorepublikové výuky algoritmizace na ZŠ

Výuka algoritmizace není na základních školách příliš populární. Bylo nalezeno pouze padesát škol v celé České Republice, které se algoritmizací nebo programování věnují. Z toho osm škol působí na území města Prahy. Plošné rozložení těchto škol znázorňuje následující obrázek.



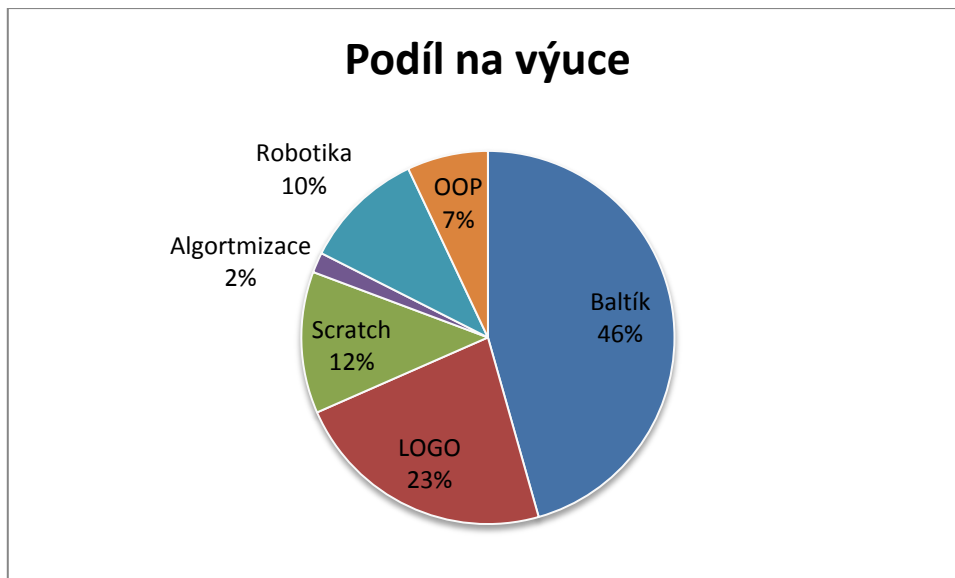
Obrázek 11: Mapa republiky s vyznačenými školami, kde se vyučuje algoritmizace

Výuka programování probíhá ve většině případů pouze v zájmových kroužcích, které jsou na školách organizovány, velký podíl také mají povinně volitelné předměty. V rámci povinných předmětů se algoritmizace a programování téměř vůbec nevyučuje. Výjimkou jsou pouze speciálně zaměřené základní školy a třídy. Těch je v republice pouze pár.

Dvacet procent škol se zabývá výukou ve více prostředích, například se jedná o kombinaci Baltík a profesionální jazyk, nebo Baltík a robotika s robotickými stavebnicemi Lego aj.

C. Zmapování celorepublikové výuky algoritmizace na ZŠ

Rozbor oblíbenosti jednotlivých nástrojů shrnuje následující graf.



Obrázek 12: Podíl jednotlivých vývojových nástrojů na výuce

Z grafu je jasně vidět, že nejpopulárnějším nástrojem pro výuku je Baltík, v Baltíkovi programuje téměř polovina všech škol. Dále je pro výuku také oblíbené LOGO, ať již Imagine LOGO, nebo Comenius LOGO. Stále vzrůstá oblíbenost Scratche. Některé školy využívají při výuce profesionální nástroje, jako například Java, C#, Visual Basic aj. Pouze na jediné škole se vyučuje nejen programování, ale i algoritmizace.

D. Zkušenosti učitele s učebnicí

D. Zkušenosti učitele s učebnicí

Na českém trhu byla nalezena jediná učebnice pro základní školy, která by se věnovala výuce programování. Jedná se o **Informatiku pro základní školy a víceletá gymnázia, 3. díl** autorů Vaníček, Mikeš. Algoritmizaci se nevěnuje žádná učebnice pro základní školy. Což také vypovídá současnému stavu výuky, kde se algoritmizace vyučuje pouze na jediné škole.

Bohužel se nepodařilo nalézt učitele, který by tuto knihu aktivně využíval při své výuce.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

1.0 Popis výzkumného vzorku

Výzkumný vzorek je tvořen algoritmičnými úlohami, které řešily děti v rámci soutěže Bobřík informatiky. Tyto otázky byly vybírány ze všech již proběhlých ročníků soutěže (první ročník proběhl v roce 2008, jedná se tudíž o čtyři ročníky) z kategorií pro žáky základních škol.

Oficiální představení soutěže je následující (Příbyl, 2012):

Bobřík informatiky je předmětová soutěž, registrovaná Ministerstvem školství. Soutěž má za sebou čtyři ročníky. Soutěž organizuje katedra informatiky Pedagogické fakulty Jihočeské univerzity v Českých Budějovicích pod patronací Jednoty školských informatiků. Soutěž probíhá na školách, u počítačů. Každý soutěžící dostane elektronický test s patnácti otázkami, odpovídá nejčastěji zaškrtnutím možností. Délka testu je čtyřicet minut. Test obsahuje otázky lehké, střední a těžké.

Soutěž zjišťuje informační gramotnost, logické a algoritmičké myšlení, jak soutěžící rozumí počítačům aj. Zaměření otázek reaguje na vývoj technologií, nástup sociálních sítí. Soutěžícími jsou žáci druhého stupně základních škol a středních škol.

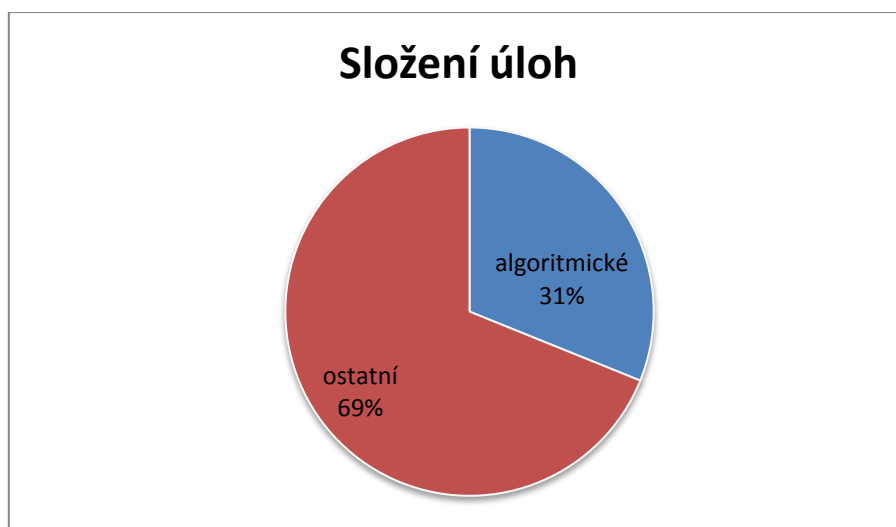
První ročník proběhl v roce 2008 a měl pouze tři kategorie (Benjamin, Junior, Senior), kdy žáci základních škol patřili do kategorie Benjamin. Prvního ročníku se zúčastnilo 4069 dětí. Druhý ročník měl opět tři kategorie. Ve třetím ročníku se kategorie pro žáky základních škol rozdělila na dvě. Pro žáky šestých a sedmých tříd byla kategorie Benjamin a žáci osmých a devátých tříd patřili do kategorie Kadet. Stejně rozdělení měl i čtvrtý ročník.

Tabulka 5: Bobřík inforamtiky - počet soutěžících

Rok	Benjamin	Kadet	Celkem
2008	1758		4069
2009	3254		8832
2010	4553	3888	13775
2011	6079	4810	17598

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Celkem za čtyři roky řešili žáci 90 úloh z obou kategorií. Z toho bylo vybráno celkem 28 algoritmičkových úloh. Tematické složení úloh shrnuje následující graf:



Obrázek 13: Tematické složení úloh

Výběr algoritmičkových úloh nebyl zcela jednoznačný a do vzorku jsou řazeny i úlohy, které nejsou primárně algoritmičkové, ale s algoritmičkováním, algoritmičkovým myšlením a programováním do značné míry souvisejí.

V každém ročníku a kategorii je rozdílný počet těchto úloh, jak shrnuje následující tabulka:

Tabulka 6: Bobřík informatiky - počet algoritmičkových úloh v ročnících

Rok	Benjamin	Kadet
2008	4	
2009	6	
2010	5	4
2011	6	3

Veškerá zde uvedená data vycházejí z oficiálních výsledků jednotlivých ročníků soutěže.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

2.0 Výsledky

Výsledky algoritmických úloh se nijak zvláště od ostatních neliší. Jedná se pouze o rozdíly v jednotkách procent. Jak shrnuje následující tabulka.

Tabulka 7: Průměrná úspěšnost v kategoriích

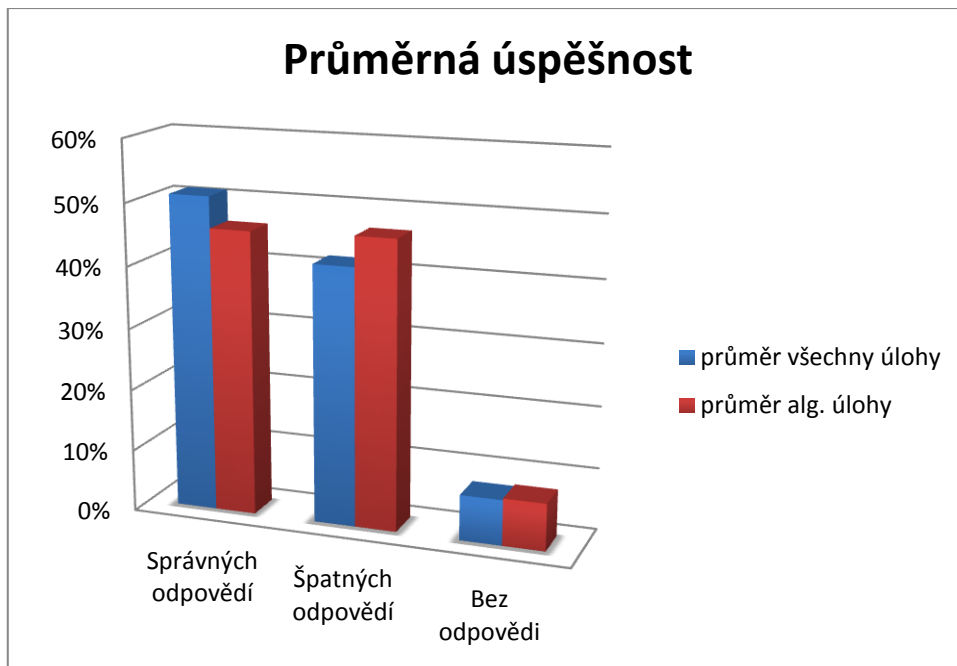
Kategorie	Popis	Správných odpovědí	Špatných odpovědí	Bez odpovědi
Benjamin	průměr všechny úlohy	50 %	42 %	8 %
	průměr alg. úlohy	45 %	46 %	9 %
Kadet	průměr všechny úlohy	52 %	41 %	7 %
	průměr alg. úlohy	46 %	47 %	7 %
Celkem	průměr všechny úlohy	51 %	42 %	7 %
	průměr alg. úlohy	46 %	46 %	8 %

V kategorii Benjamin se celkově jedná o pětiprocentní rozdíl v úspěšnosti řešení algoritmických úloh oproti úspěšnosti řešení všech úloh v dané kategorii za všechny již proběhlé ročníky. U špatných odpovědí je tento rozdíl čtyřprocentní. Rozdíl těchto dvou hodnot tvoří vyšší počet nezodpovězených algoritmických otázek.

V kategorii Kadet lze usuzovat, že zde bylo více obtížných úloh, neboť procentuální rozdíl správných odpovědí u všech úloh a algoritmických činí šest procent. Tomu také odpovídá větší podíl špatných odpovědí. V této kategorii se řešitelé méně vyhýbali nezodpovězení algoritmických úloh, kde jsou hodnoty totožné.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Celkový průměr za obě kategorie a všechny ročníky shrnuje následující graf:



Obrázek 14: Průměrná úspěšnost řešení úloh

Meziroční srovnání není zcela objektivní, neboť každý rok se řeší jiné úlohy. Některé úlohy jsou zaměřeny spíše na programování, jiné jsou více algoritmičké, logické, další jsou zaměřeny na interpretaci algoritmu nebo programu. Z tohoto důvodu není možné určit vývojovou tendenci úspěšnosti řešení algoritmičkových úloh. Dalším důvodem je malý počet doposud proběhlých ročníků soutěže.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

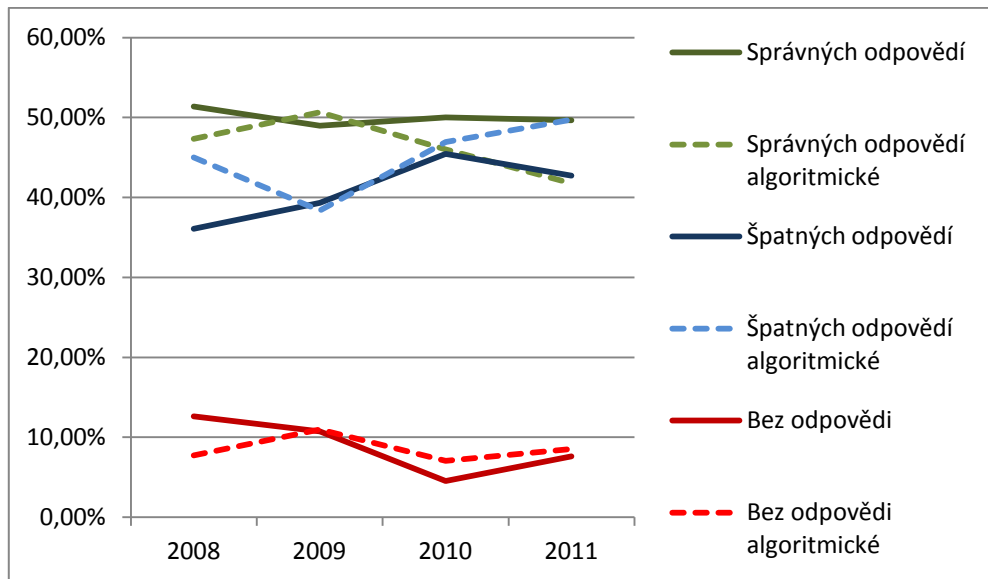
Porovnání úspěšnosti řešení všech úloh a algoritmických shrnuje následující tabulka:

Tabulka 8: Průměrná úspěšnost v jednotlivých ročnících

Kategorie	Ročník	Popis	Správných odpovědí	Špatných odpovědí	Bez odpovědi
Benjamin	2008	průměr všechny úlohy	51 %	36 %	13 %
		průměr alg. úlohy	47 %	45 %	8 %
	2009	průměr všechny úlohy	49 %	40 %	11 %
		průměr alg. úlohy	51 %	38 %	11 %
	2010	průměr všechny úlohy	50 %	45 %	5 %
		průměr alg. úlohy	46 %	47 %	7 %
	2011	průměr všechny úlohy	49 %	43 %	8 %
		průměr alg. úlohy	42 %	50 %	8 %
Kadet	2010	průměr všechny úlohy	48 %	46 %	6 %
		průměr alg. úlohy	49 %	44 %	7 %
	2011	průměr všechny úlohy	55 %	38 %	7 %
		průměr alg. úlohy	44 %	49 %	7 %

Pro lepší přehlednost a lepší vizuální porovnání jsou tyto data procentuálně zobrazena v následujících grafech pro jednotlivé kategorie. Průměrné výsledky jsou zobrazovány plnou čarou a k nim odpovídající výsledky algoritmických úloh jsou zobrazeny přerušovanou čarou a jsou o odstín světlejší.

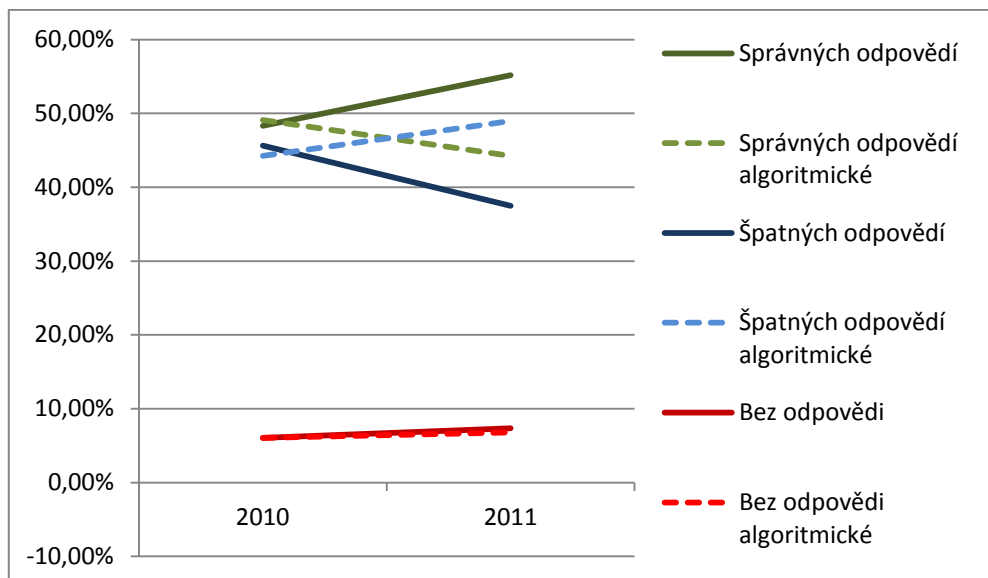
E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 15: Průměrná úspěšnost v kategorii Benjamin

Jak je patrné z grafu pro kategorii Benjamin (obrázek 20), úspěšnost řešení algoritmičkových úloh je ve většině ročníků o málo nižší, než je průměr všech úloh v daném ročníku a kategorii. Jedinou výjimkou je druhý ročník soutěže (rok 2009), kdy úspěšnost řešení algoritmičkových úloh byla nadprůměrná. Naopak zatím poslední ročník soutěže (rok 2011) se ukázal jako výrazněji podprůměrný a procentuální rozdíl správných odpovědí algoritmičkových a všech úloh byl osm procent, což je prozatím nejvíce. Procento žáků, kteří dané otázky nezodpověděli, je téměř stejné s průměrem všech úloh. Jediným ročníkem s výraznějším rozdílem byl první ročník (rok 2008), kdy se algoritmičkovým otázkám vyhnulo pět procent méně dětí, než byl průměr u všech úloh.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 16: Průměrná úspěšnost v kategorii Kadet

V kategorii Kadet proběhly prozatím pouze dva ročníky, jejichž procentuální výsledky se značně liší. V roce 2010 byla úspěšnost řešení algoritmických úloh dokonce mírně nadprůměrná, v roce 2011 tomu bylo naopak a průměrná úspěšnost řešení všech úloh byla o více než deset procent vyšší, než u algoritmických úloh. Procento žáků, kteří algoritmické úlohy nezodpověděli, je téměř shodné s průměrem všech úloh.

3.0 Rozbor úloh

Zde bude proveden rozbor jednotlivých úloh.

Rozbor každé úlohy má předem danou strukturu. Nejprve je proveden rozbor zadání s uvedenou obtížností, jak ji určili autoři, následuje zadání, pole odpovědí. Následují výsledky. Závěr tvoří komentář s nástinem možných příčin špatného řešení. V mnoha případech je příčinou nesprávného řešení nepozornost, nedbalost při čtení zadání, nebo stres doprovázející každou soutěž.

Úloha 1: AUTA V GARÁŽI

Rozbor zadání:

První algoritmická úloha v tomto ročníku byla Auta v garáži. Jednalo se o těžkou úlohu. Úloha spočívala v pochopení principu datové struktury zásobník. V zadání bylo uvedeno, že

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

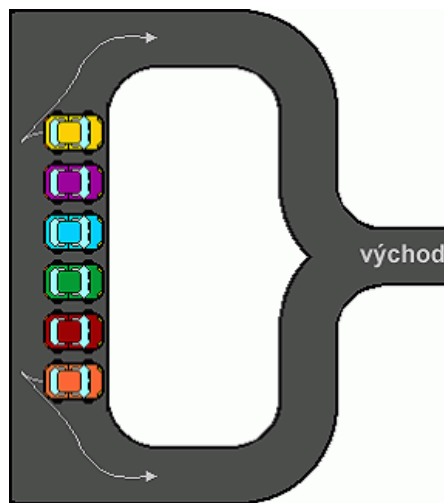
auta stojí na parkovišti, vyjíždět mohou pouze krajní. Řešitelé měli určit pořadí aut, jak vyjedou z garáže.

Zadání:

Auta stojí v úzké garáži takto:

Z garáže může vyjíždět vždy pouze jedno z krajních aut, jak naznačují šipky. Tedy jako první žluté nebo oranžové.

V jakém pořadí mohla auta vyjet z garáže?



Obrázek 17: Benjamin 2009, Auta v garáži - zadání

Pole odpovědí:

-
-
-
-
- Žádná odpověď

Obrázek 18: Benjamin 2009, Auta v garáži - odpovědi

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 9: Benjamin 2008, Auta v garáži

Odpověď	1.	2.	3.	4.	5.
Počty	168	1812	808	118	348
v %	5 %	56 %	25 %	3 %	11 %

Komentář:

Nesprávnost řešení této úlohy byla nejspíš způsobena neznalostí datové struktury zásobníku, špatnou aplikací daného algoritmu či nedostatečnou představivostí.

Úloha 2: AUTODRÁHA

Rozbor zadání:

Obtížnost řešení této otázky byla podle autorů těžká.

Jednalo se o sestavení autodráhy podle předem daného algoritmu a následně rozhodnout, který z nabízených algoritmů sestavení byl reálný a vznikl tím uzavřený okruh.

Zadání:

*Sofiina **autodráha** obsahuje čtyři druhy dílů.*

Zatáčky mohou být pravotočivé nebo levotočivé: levá/pravá 90°, levá/pravá 60°

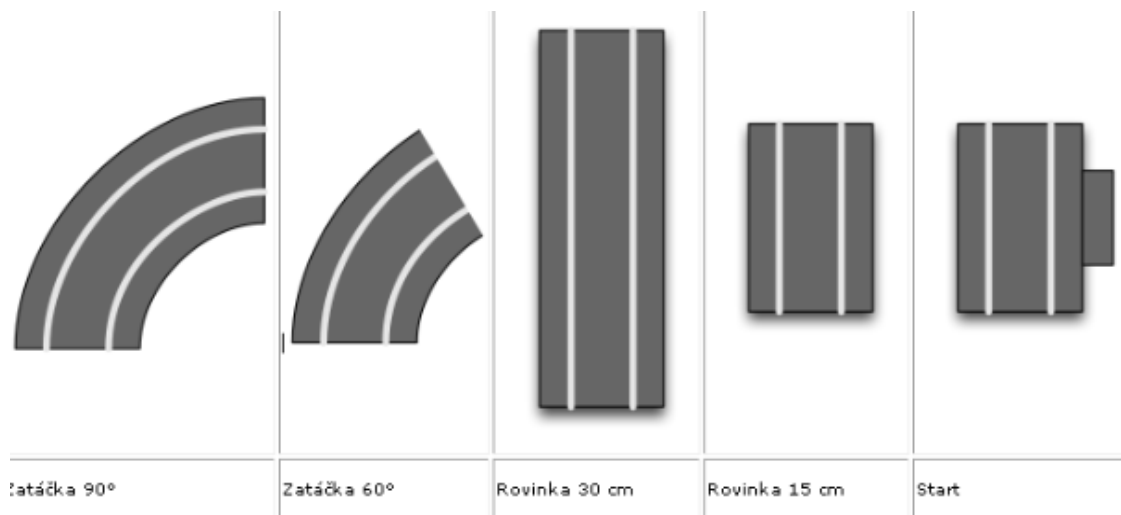
Jedna z 15 cm dlouhých rovinek přivádí elektrinu; nazývá se start.

Sestavená trať funguje správně, pouze pokud obsahuje start a je uzavřená.

Sofie si ráda hraje s autodráhou a pořád vymýšlí nějaké nové trati. Aby si je pamatovala, dělá si zápisky. Například její první trať vypadala takto: start, levá 90, levá 90, rovinka 30, levá 90, levá 90, rovinka 15. Byla to správná trať.

*Jednou si Sofie chtěla znovu postavit staré trati. Když si procházela zápisky, zjistila, že **jen jedna trať z následujících je správná. Poznáš, která?***

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 19: Kadet 2010, Autodráha

Pole odpovědí:

1. *pravá 60, levá 60, rovinka 15, pravá 90, pravá 90, pravá 60, levá 60, start, pravá 90, pravá 90*
2. *rovinka 30, levá 90, levá 90, rovinka 15, start, pravá 90, pravá 90*
3. *levá 90, levá 90, rovinka 30, levá 60, levá 60, rovinka 15, rovinka 15*
4. *rovinka 15, pravá 60, levá 60, levá 60, levá 60, rovinka 30, levá 90, levá 90, start*
5. *Žádná odpověď*

Výsledky:

Tabulka 10: Kadet 2010, Autodráha

Odpověď	1.	2.	3.	4.	5.
Počet	835	1422	457	775	399
(v %)	21 %	37 %	12 %	20 %	10 %

Komentář:

Nesprávnost řešení této úlohy nejspíš spočívala ve špatné představivosti, nebo nenačrtnutí si jednotlivých okruhů. Případně neuvědoměním si, že byly k dispozici dva druhy zatáček a dvě délky rovných částí.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Úloha 3: BROUK A JEHO POUŤ

Rozbor zadání:

Tato úloha byla zařazena jak do kategorie Benjamin, tak i do kategorie Kadet. V kategorii Benjamin se jednalo o středně těžkou úlohu, v kategorii Kadet byla již zařazena mezi lehké.

V tomto případě se jednalo o ryze algoritmickou úlohu, která říkala, že ve svém důsledku pravidla každé hry jsou algoritmem, jak danou hru hrát a vyhrát a na hráčích je pouze, jak danému algoritmu porozumějí a dokáží jej aplikovat na konkrétní případy.

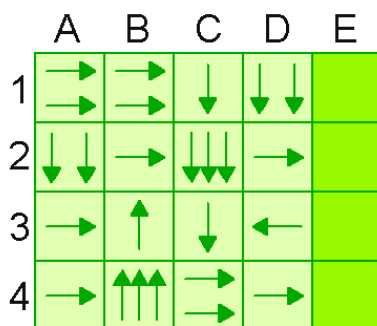
Princip této hry spočíval v nalezení počátku cesty, jejímž průchodem se dostane brouk do cíle.

Zadání:

Brouk se pohybuje po hracím plánu (obrázek) podle těchto pravidel:

- *Začíná svoji cestu na libovolném políčku.*
- *Při jednom tahu popojede o tolik políček, kolik šipek je na políčku, na kterém stojí, ve směru, který šipky ukazují.*
- *Během tohoto tahu se brouk nezajímá o šipky na políčkách, přes která se pohybuje.*
- *Brouk opakuje své tahy tak dlouho, dokud nevyleze pryč z hracího plánu nebo nedorazí do některého políčka bez šipek (sloupec E).*

Ve kterém z políček ve sloupci A musí brouk svoji pouť začít, aby skončil v políčku ve sloupci E?



Obrázek 20: Kadet 2010, Brouk a jeho pouť

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pole odpovědí:

1. A1, A2
2. A2, A4
3. A1, A4
4. A2, A3, A4
5. Žádná odpověď

Výsledky:

Tabulka 11: Benjamin 2010, Brouk a jeho pouť

	Odpověď	1.	2.	3.	4.	5.
Benjamin	Počty	545	2034	895	765	314
	v %	12 %	45 %	19 %	17 %	7 %
Kadet	Počty	366	2308	559	470	185
	v %	9 %	60 %	14 %	12 %	5 %
Celkem	v %	11 %	51 %	17 %	15 %	6 %

Komentář:

Nesprávnost řešení této úlohy mohla být způsobena nepochopením herního algoritmu nebo jeho nesprávnou aplikací.

Úspěšnost této úlohy v kategorii Kadet byla o patnáct procent vyšší než u téže úlohy v kategorii Benjamin. Lze tudíž usuzovat o rozvoji algoritmického myšlení u dětí s rostoucím věkem, bez ohledu, zda se daná problematika vyučuje.

Úloha 4: CO JE PROGRAM?

Rozbor zadání:

Obtížnost řešení této otázky byla podle autorů střední.

Úloha spočívala v uvědomění si, jaké vlastnosti musí mít každý program, resp. algoritmus a poté vyhodnocení, která z uvedených vět tyto vlastnosti splňuje.

Zadání:

Jednoduché programy jsou sady příkazů. Příkazy určují, co se má dělat. Kterou z těchto vět může považovat za jednoduchý program?

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pole odpovědí:

1. *Dva plus dva jsou čtyři.*
2. *Vítej ve skutečném světě!*
3. *Co je informace?*
4. *Vejdi a zavři dveře.*
5. *Žádná odpověď*

Výsledky:

Tabulka 12: Kadet 2011, Co je program?

Odpověď	1.	2.	3.	4.	5.
Počet	1139	296	302	2887	186
(v %)	24 %	6 %	6 %	60 %	4 %

Komentář:

Nesprávnost řešení této úlohy byla způsobena neznalostí základních vlastností každého algoritmu.

Úloha 5: ČERNOBÍLÉ OBRÁZKY PÍSMEN

Rozbor zadání:

Řešení této úlohy bylo těžké.

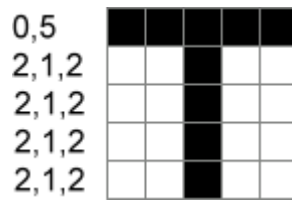
Byl dán algoritmus kódování. Pro zjednodušení byl uveden ještě příklad zakódování písmene T. Řešitelé měli tento algoritmus kódování použít na zakódované písmeno a dekodovat jej.

Zadání:

Obrázky na počítači jsou zobrazené v mřížce, která se skládá z malých bodů zvaných pixely. Černobílý obrázek je složený z černých nebo bílých pixelů. Počítač si ukládá do své paměti černobílé obrázky pomocí čísel, např. tímto způsobem:

První číslo zleva vždycky určuje počet bílých pixelů za sebou, další číslo určuje počet černých pixelů, případně opět počet bílých atd.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 21: Benjamin 2011, Černobílé obrázky písmen

Příklad (obrázek):

Spodní řádka začíná dvěma bílými pixely, pokračuje jeden černý a opět dva bílé.

První řádka tedy začíná 0 bílými pixely, pokračuje 5 černými pixely.

Které písmeno je popsáno následujícími čísly?

- 0,1,3,1
- 0,1,3,1
- 0,5
- 0,1,3,1
- 0,1,3,1

Pole odpovědí:

1. E
2. B
3. U
4. H
5. Žádná odpověď

Výsledky:

Tabulka 13: Benjamin 2011, Černobílé obrázky písmen

Odpověď	1.	2.	3.	4.	5.
Počty	766	631	465	3429	788
(v %)	13 %	10 %	8 %	56 %	13 %

Komentář:

Nesprávnost řešení této úlohy spočívala nejspíš v nepochopení algoritmu kódování.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Úloha 6: DUHOVÁ VEJCE

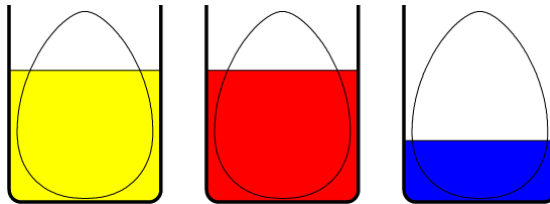
Rozbor zadání:

Tato úloha byla označena jako těžká.

Tato úloha byla postavena na stejném principu jako úloha Rozstříhané obrázky. Řešitelé měli najít algoritmus obarvení jednotlivých vajíček. Poté měli rozhodnout, které z nabídnutých vajíček mohlo vzniknout z daných barev. Aby to nebylo tak jednoduché, měli v zadání dáno několik podmínek pro vznik nových barev.

Zadání:

Linda obarvuje vejce a má tři hrníčky s barvami.



Obrázek 22: Benjamin 2009, Duhová vejce - zadání

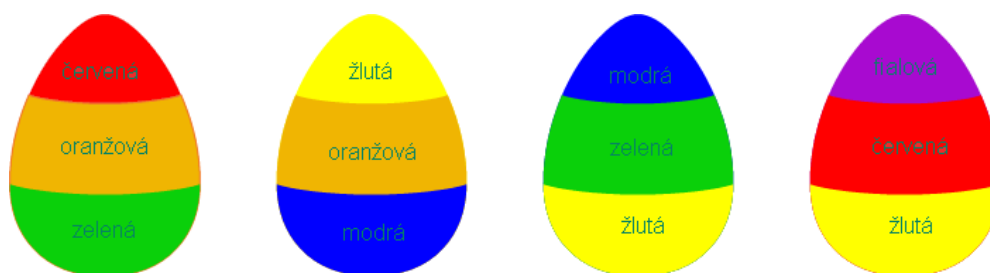
Červené a žluté barvy má hodně, takže do nich může vajíčko ponořit až do dvou třetin. Ale do modré barvy může vajíčko ponořit nanejvýš do třetiny. Linda ponořila vejce vždy až na dno. Když ponořila obarvené vejce do další barvy, tak

- *žlutá a červená se změnila v oranžovou;*
- *žlutá a modrá se změnila v zelenou;*
- *červená a modrá se změnila ve fialovou.*

Například: Kdyby ponořila jedno vejce do červené a potom do modré barvy, pak jej otočila a znovu ponořila do modré, dostala by fialovo-červeně-modré vejce.

Pouze jedno vejce z těchto mohlo být obarveno Lindou. Které?

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 23: Benjamin 2009, Duhová vejce - řešení

Pole odpovědí:

1. *Fialová – červená – žlutá*
2. *Červená – oranžová – zelená*
3. *Modrá – zelená – žlutá*
4. *Žlutá – oranžová – modrá*
5. *Žádná odpověď*

Výsledky:

Tabulka 14: Benjamin 2009, Duhová vejce

Odpověď	1.	2.	3.	4.	5.
Počty	614	1262	353	581	417
(v %)	19 %	39 %	11 %	18 %	13 %

Komentář:

Nesprávnost řešení této úlohy mohla být způsobena nesprávnou aplikací pravidel obarvení, nedostatečnou představivostí, nebo nepozorností, neboť správně bylo první vejíčko, ale v možnostech to byla pokaždé jiná odpověď.

Úloha 7: HONBA ZA POKLADEM

Rozbor zadání:

Tato úloha byla označena jako těžká. Algoritmický charakter této otázky spočíval ve schopnosti vyhodnocovat více podmínek. Po správném vyhodnocení zadaných podmínek řešitel zjistil, kde je na mapě ukryt poklad a poté měl pouze popsat souřadnice tohoto místa.

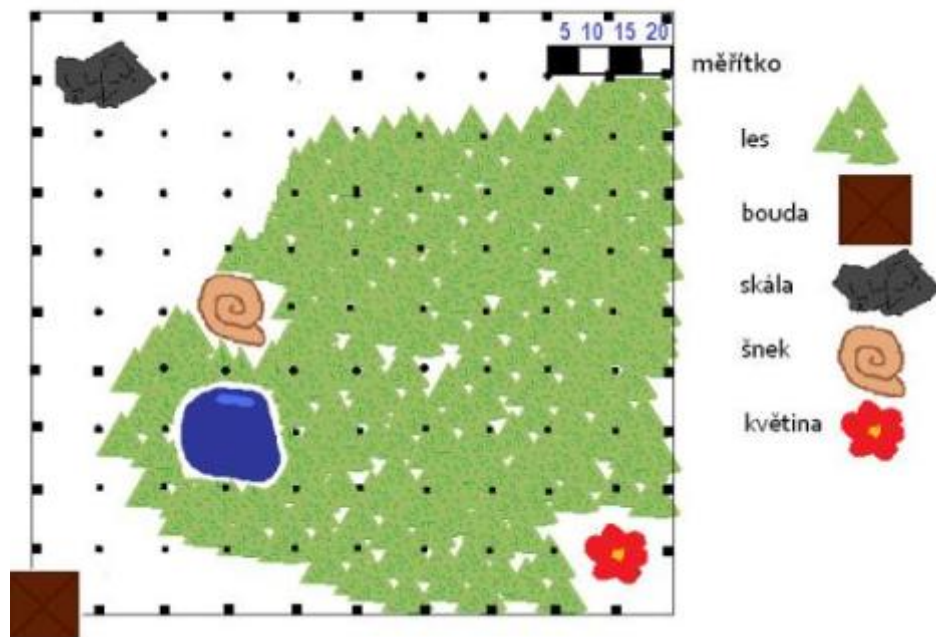
E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Zadání:

Jirka se chystá na honbu za pokladem. Honba začíná v boudě. Má tři informace:

- Poklad najdeš v lese.
- Poklad je v přímém směru od skály ke květině.
- Poklad je schovaný 20 kroků od šneka.

Který návod vede k pokladu?



Obrázek 24: Benjamin 2011, Honba za pokladem

Pole odpovědí:

1. Jdi 80 kroků východně (→) a 20 kroků severně (↑)
2. Jdi 30 kroků východně (→) a 70 kroků severně (↑)
3. Jdi 50 kroků východně (→) a 50 kroků severně (↑)
4. Jdi 30 kroků východně (→) a 30 kroků severně (↑)
5. Žádná odpověď

Výsledky:

Tabulka 15: Benjamin 2011, Honba za pokladem

Odpověď	1.	2.	3.	4.	5.
Počty	896	789	1947	1672	775
(v %)	15 %	13 %	32 %	27 %	13 %

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Komentář:

Nesprávnost řešení této úlohy by mohla spočívat v nesprávném vyhodnocení podmínek, nebo následně chybném určení souřadnic.

Úloha 8: INT VÝPOČET 0

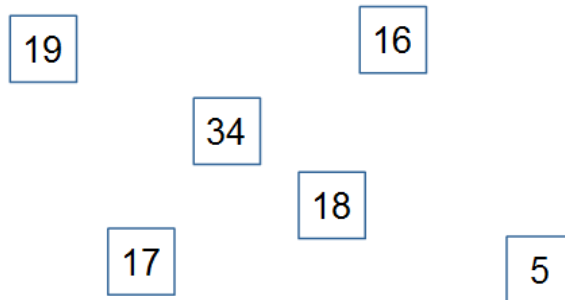
Rozbor zadání:

Třetí úloha měla matematické zaměření. Její obtížnost byla střední. Dalo by se říci, že tato úloha neměla s algoritmizací nic společného, ale zápis matematických operací například ve formě vývojového diagramu je typickým příkladem. V tomto případě si měli řešitelé tento vývojový diagram pouze představit. Z jiného pohledu se jedná o práci s celočíselným datovým typem, což opět do algoritmizace patří.

Zadání:

Doplň správná čísla, aby vyšla rovnost.

$$\square + \square - \square = 0$$



Obrázek 25: Benjamin 2009, INT výpočet 0

Pole odpovědí:

1. $18 + 16 - 34$ *nebo* $16 + 18 - 34 = 0$
2. *Jiná kombinace*
3. *Žádná odpověď*

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 16: Benjamin 2009, INT výpočet 0

Odpověď	1.	2.	3.
Počty	1847	1090	317
(v %)	57 %	33 %	10 %

Komentář:

Nesprávnost řešení této úlohy byla nejspíš způsobena špatnými matematickými a kombinačními dovednostmi.

Úloha 9: KORÁLKY

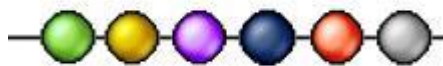
Rozbor zadání:

Obtížnost úlohy Korálky je těžká.

V případě této úlohy se jedná o typickou práci s datovým typem zásobník. Jedná se o obdobnou úlohu jako první algoritmická úloha v tomto ročníku Auta v garáži.

Zadání:

Janka navléká korálky. Na jedné niti má navlečenou zásobu korálků, každý jiný:



Obrázek 26: Benjamin 2009, Korálky – zadání 1

Druhá nit je zatím prázdná a na jednom konci má uzel.



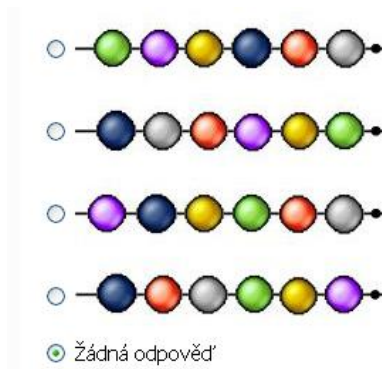
Obrázek 27: Benjamin 2009, Korálky – zadání 2

Z ní chce Janka udělat náramek. Bude na ni navlékat korálky tak, že vždy vezme některý korálek ze zásoby (ale může vzít jen ten, který je na levém nebo pravém konci nitě) a pak ho navleče na druhou nit (samozřejmě na ten konec, který není zauzlovaný).

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Který z náramků se dá vytvořit tímto způsobem?

Pole odpovědí:



Obrázek 28: Benjamin 2009, Korálky - odpovědi

Výsledky:

Tabulka 17: Benjamin 2009, Korálky

Odpověď	1.	2.	3.	4.	5.
Počty	788	312	1580	189	385
(v %)	24 %	9 %	48 %	6 %	12 %

Komentář:

Stejně jako u úlohy Auta v garáži byla nejspíš nesprávnost řešení způsobena neznalostí datové struktury zásobník, nesprávnou aplikací algoritmu výběru korálků, nebo nepozorností.

Úloha 10: NEJLEVNĚJŠÍ PODNÁJEM

Rozbor zadání:

Úloha Nejlevnější podnájem byla označena jako středně těžká.

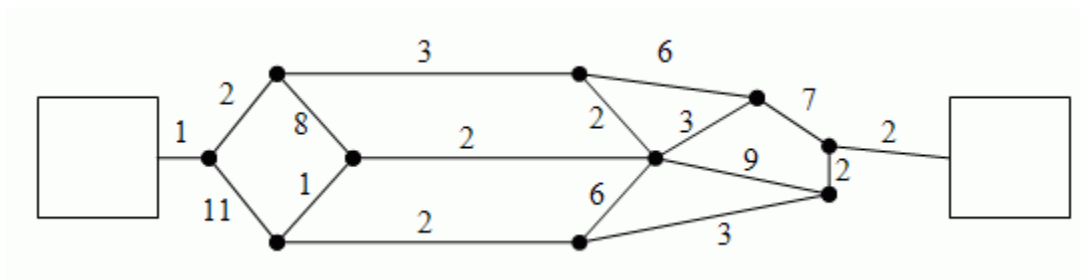
V tomto případě se jednalo o pochopení datové struktury ohodnoceného grafu. Žáci měli nalézt nejlevnější cestu z bodu A do bodu B.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Zadání:

Firma používá k síťovému spojení dvou svých vzdálených poboček pronajímané kabelové spojení. Na obrázku jsou úsečkami znázorněny jednotlivé kabely, které si firma může pronajmout, u každé je popsána týdenní cena za pronájem v Eurech. Firma si může vybrat, které kabely si pronajme.

Jaká je nejlevnější varianta týdenního pronájmu?



Obrázek 29: Benjamin 2009, Nejlevnější podnájem

Pole odpovědí:

1. 32
2. 21
3. 20
4. 17
5. Žádná odpověď

Výsledky:

Tabulka 18: Benjamin 2009, Nejlevnější podnájem

Odpověď	1.	2.	3.	4.	5.
Počty	185	1132	1091	368	478
(v %)	6 %	35 %	33 %	11 %	15 %

Komentář:

Nesprávnost řešení této úlohy mohla být způsobena povrchním přístupem, kdy žáci vypočítali cenu jedné nebo dvou cest, a když jim vyšlo číslo ze seznamu, dále nepokračovali.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Úloha 11: PARKING

Rozbor zadání:

První z těchto úloh byla povinná pro všechny řešitele. Obtížnost řešení této otázky je podle autorů střední.

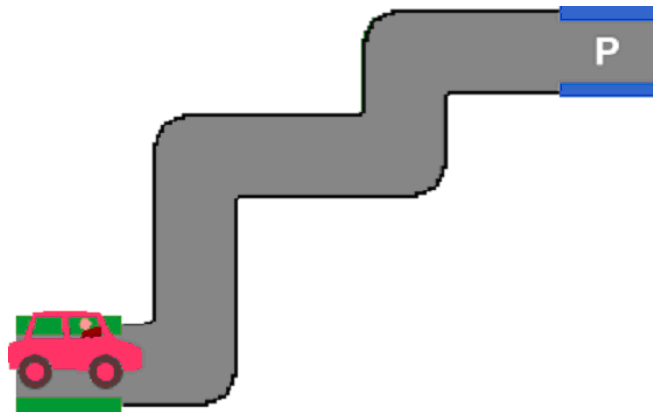
Úloha spočívala v sestavení algoritmu jízdy auta na parkoviště z předem daných příkazů. Jedná se tedy o sestavení jednoduchého nevětveného algoritmu.

Zadání:

Cestu auta ze startovního místa na parkoviště můžeš popsat následujícími příkazy:

- **dopředu** - jed' rovně k zatáčce nebo k parkovišti
- **vpravo** - v zatáčce zatoč na místě doprava
- **vlevo** - v zatáčce zatoč na místě doleva

Která sada příkazů správně popisuje cestu ze startovního místa na parkoviště?



Obrázek 30: Benjamin 2008, Parking

Pole odpovědí:

1. B: dopředu, vlevo, dopředu, vpravo, dopředu, vlevo, dopředu, vpravo, dopředu
2. A: dopředu, vlevo, dopředu, vlevo, dopředu, vlevo, dopředu, vpravo, dopředu
3. C: dopředu, vlevo, dopředu, vpravo, dopředu, vlevo, dopředu, vlevo, dopředu
4. D: vlevo, dopředu, vpravo, dopředu, vlevo, dopředu, vpravo, dopředu
5. Žádná odpověď

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 19: Benjamin 2008, Parking

Odpověď	1.	2.	3.	4.	5.
Počty	122	1201	192	196	47
v %	7 %	68 %	11 %	11 %	3 %

Komentář:

Nesprávnost řešení této úlohy bylo nejspíše způsobeno nepozorností a nedbalým přečtením zadání.

Úloha 12: PRAVIDLA OBLÉKÁNÍ

Rozbor zadání:

Řešení této otázky byla podle autorů středně obtížné.

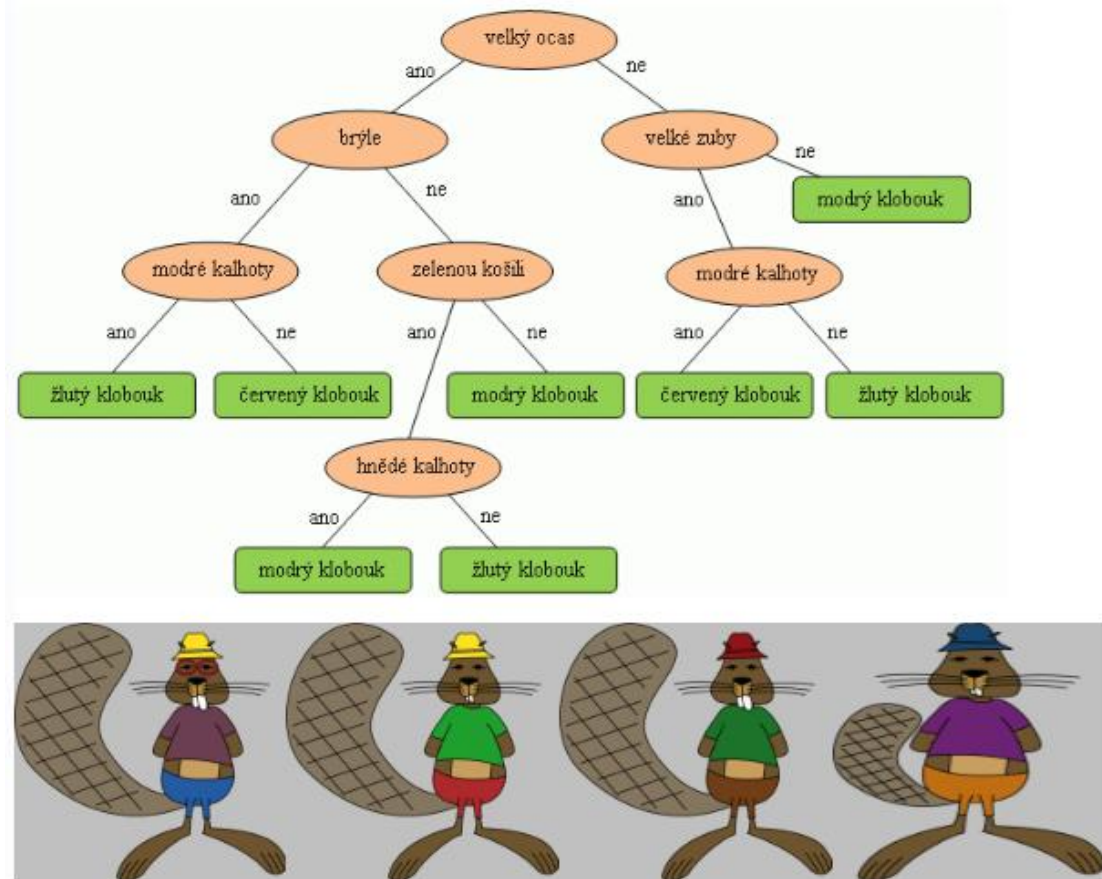
V této úloze byl daný vývojový diagram a řešitel měl pouze určit, který z nabízených výstupů nebyl možný. Jednalo se tedy o schopnost interpretace vývojového diagramu jako jedné z variant zápisu algoritmu a vyhodnocování podmínek.

Zadání:

Žáci školy v Bobřicích mají ve školním řádu pravidla oblékání. Pravidla pro všechny žáky školy jsou popsána v diagramu. Někteří bobři se však tento systém pravidel nedodržují.

Urči, který bobr není oblečen správně, tedy který nepoužívá správnou kombinaci oblečení.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 31: Benjamin 2011, Pravidla oblékání

Pole odpovědí:

1.
2.
3.
4.
5. Žádná odpověď

Obrázek 32: Benjamin 2011, Pravidla oblékání - odpovědi

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 20: Benjamin 2011, Pravidla oblékání

Odpověď	1.	2.	3.	4.	5.
Počty	1239	1965	710	1826	339
(v %)	20 %	32 %	12 %	30 %	6 %

Komentář:

Nesprávnost řešení této úlohy bylo nejspíš způsobeno nepochopením tohoto zápisu, nebo nepozorností při procházení diagramu pro jednotlivé bobříky.

Úloha 13: PROCHÁZKA ROBOTA

Rozbor zadání:

Tato úloha byla označena jako těžká.

V tomto případě se jednalo o programátorskou úlohu. Byly dány dva příkazy, které zajišťují pohyb robota. Dále byly nabídnuty možné výstupy. Řešitel měl rozhodnout, kterého pohybu robot nebyl schopen, pokud víme, kolikrát dané metody byly volány. Nejjednodušším řešením této úlohy bylo napsání posloupnosti těchto příkazů pro každý výstup a poté rozhodnout, kde neodpovídá počet jednotlivých příkazů zadání.

Zadání:

Jirka programuje robota, který jezdí po podlaze, cestou zakresluje svoji dráhu a vykonává pouze dva příkazy:

- *jed'_doleva. Robot udělá deset kroků vpřed a pak se otočí o 90 stupňů vlevo.*
- *jed'_doprava. Robot udělá deset kroků vpřed a pak se otočí o 90 stupňů vpravo.*

V počáteční pozici robot stojí ve směru ↑ (na sever).

Jirka v programu použil 3 x příkaz jed'_doleva a dvakrát příkaz jed'_doprava. Kterou čáru robot určitě Nevytvořil?

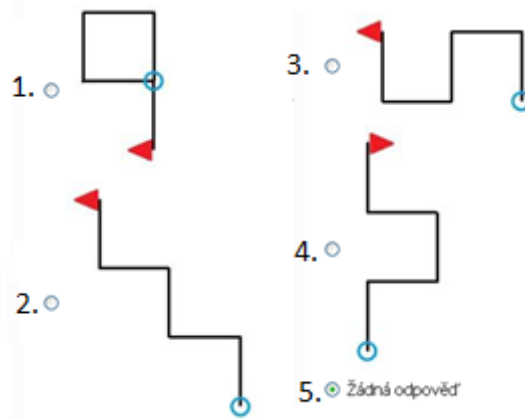
Vysvětlivky:

Počáteční pozice robota je značena modrým kolečkem.

Koncová pozice robota je označena červenou šipkou.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pole odpovědí:



Obrázek 33: Kadet 2011, Procházka robota

Výsledky:

Tabulka 21: Kadet 2011, Procházka robota

Odpověď	1.	2.	3.	4.	5.
Počet	1549	783	564	1651	263
(v %)	32 %	16 %	12 %	34 %	6 %

Komentář:

Nesprávnost řešení této úlohy nejspíš spočívala v nedbalém přečtení zadání, kde je řečeno, že se příkaz `jed' _doleva` provede třikrát a `jed' _doprava` dvakrát.

Úloha 14: PROJÍŽDKA MĚSTEM

Rozbor zadání:

Tato úloha byla označena jako těžká.

Úloha spočívala v nalezení chyby v mapě, která reprezentuje grafický záznam algoritmu cesty. Jednalo se o nalezení chyby v posloupnosti příkazů dopředu a vpravo.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 22: Benjamin 2010, Projížďka městem

Odpověď	1.	2.	3.	4.	5.
Počty	675	1565	1031	632	650
(v %)	15 %	34 %	23 %	14 %	14 %

Komentář:

Nesprávnost řešení této úlohy mohla být způsobena nedostatečnou představivostí, kdy si řešitelé nebyli schopni představit, modifikaci mapy při zalomení v různých bodech.

Úloha 15: PŘEHAZOVÁNÍ PÍSMEN

Rozbor zadání:

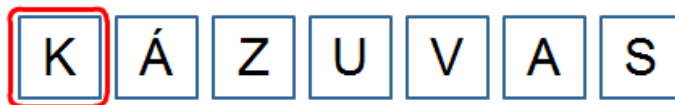
Jednalo se o lehkou úlohu. Princip této úlohy spočíval v nalezení co nejefektivnějšího algoritmu třídění pro daný problém.

Zadání:

Bruno si rád hraje se slovy. Nejrádší má hru, v které vezme první písmeno slova a vymění ho s některým jiným písmenem v tomto slově.

Například ze slova KÁZUVAS dostane jednou výměnou slovo VÁZUKAS.

Pomož mu. Klikěj tak, abys v co nejméně krocích (výměnách písmen) ze slova KÁZUVAS udělal slovo ZÁSUVKA.



Obrázek 35: Benjamin 2011, Přehazování písmen

Pole odpovědí:

1. 3 kroky
2. Jiný počet
3. Žádná odpověď

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 23: Benjamin 2011, Přehazování písmen

Odpověď	1.	2.	3.
Počty	1994	3558	527
(v %)	33 %	58 %	9 %

Komentář:

Nesprávnost řešení této úlohy byla způsobena náhodným přehazování písmen.

Úloha 16: RANNÍ BĚH

Rozbor zadání:

Jednalo se o těžkou úlohu.

Algoritmus běhání byl popsán prostřednictvím procedur a podprocedur. Řešitel měl určit, kolikrát se provede nejvnitřnější procedura a na základě toho vypočítat kolik kroků se při této aktivitě provedlo.

Zadání:

Mirek je baseballista a tak trénuje. Každé ráno běhá kolem domu a má předepsaný tréninkový plán, který přesně dodržuje. Na dnešní den má naplánováno:

- *Aktivita Běhání*
 - *Vykonej aktivitu Oběhni_blok*
 - *Vykonej aktivitu Oběhni_blok*
 - *Vykonej aktivitu Oběhni_blok*
- *Aktivita Oběhni_blok*
 - *Vykonej aktivitu Přeběhni_ulici*
 - *Vykonej aktivitu Přeběhni_ulici*
 - *Vykonej aktivitu Přeběhni_ulici*
 - *Vykonej aktivitu Přeběhni_ulici*
- *Aktivita Přeběhni_ulici*
 - *Uběhni 100 kroků*
 - *Zahni doleva*

Kolik Mirek uběhne kroků, vykoná-li aktivitu Běhání?

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pole odpovědí:

1. 100
2. 400
3. 300
4. 1200
5. Žádná odpověď

Výsledky:

Tabulka 24: Kadet 2010, Ranní běh

Odpověď	1.	2.	3.	4.	5.
Počet	206	550	497	2317	318
(v %)	5 %	14 %	13 %	60%	8 %

Komentář:

Nesprávnost řešení této úlohy mohla být způsobena nepochopením principu procedur.

Úloha 17: RAZÍTKOVÁNÍ

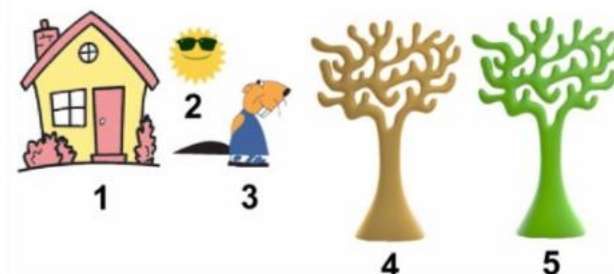
Rozbor zadání:

V tomto případě se jednalo o lehkou úlohu.

Řešitelé měli z předem daných částí obrázku jejich vrstvením sestavit výslednou koláž. Jednalo se o obdobnou úlohu, jako byla v minulém ročníku úloha Rozstříhané obrázky, kde se skládal „smajlík“.

Zadání:

Bobřík má pět razítek, očíslovaných od 1 do 5.



Obrázek 36: Benjamin 2009, Razítkování - zadání

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Natiskal krásný obrázek:



Obrázek 37: Benjamin 2009, Razítkování - řešení

V jakém pořadí použil razítka?

Pole odpovědí:

1. 5 – 2 – 3 – 4 – 1
2. 5 – 3 – 4 – 2 – 1
3. 5 – 2 – 4 – 3 – 1
4. 5 – 4 – 2 – 3 – 1
5. Žádná odpověď

Výsledky:

Tabulka 25: Benjamin 2009, Razítkování

Odpověď	1.	2.	3.	4.	5.
Počty	2298	160	402	195	199
(v %)	71 %	5 %	12 %	6 %	6 %

Komentář:

Nesprávnost řešení této úlohy byla způsobena nedostatečnou představivostí, nedostatkem algoritmického myšlení, kdy řešitel nebyl schopen sestavit algoritmus vrstvení.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Úloha 18: RGB MŘÍŽKA

Rozbor zadání:

Tato úloha byla označena jako lehká.

Jednalo se o doplnění chybějící RGB mřížky. Tato úloha není primárně algoritmická, ale každý programátor musí umět pracovat s maticí, kterou tato mřížka reprezentuje. Také se na tuto úlohu dá pohlížet jako na cyklus, kdy se v jednotlivých řádcích stále opakují stejná políčka a řádky se ob jeden opakují. Řešitel měl pouze tento algoritmus aplikovat na chybějící část mřížky.

Zadání:

V počítačových monitorech se pro zobrazování barev používá sada malinkých barevných plošek, z nichž je obrazovka složena. Každá ploška v mřížce je červená, zelená nebo modrá.

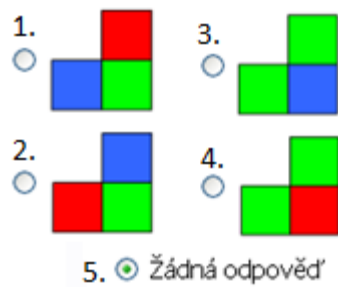
*Částečně vybarvená mřížka velikosti 8x11 pixelů je na obrázku. Pokud budeme předpokládat, že to bude pravidelně pokračovat dál, **jaké barvy budou v šedě vybarvené části v pravém dolním rohu?***



Obrázek 38: Benjamin 2011, RGB mřížka - zadání

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pole odpovědí:



Obrázek 39: Benjamin 2011, RGB mřížka - odpovědi

Výsledky:

Tabulka 26: Benjamin 2011, RGB mřížka

Odpověď	1.	2.	3.	4.	5.
Počty	4506	583	492	352	146
(v %)	74 %	10 %	8 %	6 %	2 %

Komentář:

Nesprávnost řešení této úlohy spočívala v neodhalení pravidla pro opakování, nebo nedbalém odpočítávání si políček.

Úloha 19: ROBOT - LAJNOVAČKA

Rozbor zadání:

Tato úloha byla označena jako těžká.

Úloha spočívala v sestavení algoritmu nakreslení kruhu z předem daných příkazů. Jedná se tedy o cyklus s daným počtem opakování. Tělo cyklu je tvořeno sekvencí příkazů. Pro lepší pochopení daného úkolu je zde uveden příklad nakreslení kruhu s dvojnásobným průměrem, než je správné řešení. Jednalo se tedy pouze o správnou interpretaci tohoto příkladu a uvědomění si, jaký parametr se má změnit.

Zadání:

Robot pro kreslení bílých čar na fotbalovém stadionu je ovládán programem, napsaném v počítači.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

- Příkaz *DOPŘEDU 1* znamená, že robot popojede o 1 decimetr dopředu.
- Příkaz *VLEVO 1* znamená, že robot se na místě otočí o úhel velikosti 1° .
- Příkaz *OPAKUJ 3 [PŘÍKAZY]* znamená, že robot třikrát vykoná příkazy v závorce.

Např.:

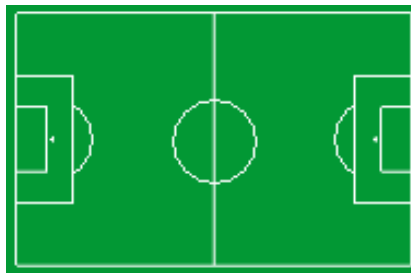
- *DOPŘEDU 10* znamená, že robot popoleze o $10 \text{ dm} = 1 \text{ metr}$.
- *VLEVO 10* - robot se na místě otočí o 10° .
- *DOPŘEDU 3 VLEVO 90 DOPŘEDU 2* - robot nakreslí písmeno *L*.
- *OPAKUJ 5 [DOPŘEDU 2]* znamená, že robot pětkrát pojede dopředu o 2 dm a ujede tak 1 m.

Středový kruh fotbalového hřiště se nakreslil příkazem:

OPAKUJ 360 [DOPŘEDU 2 VLEVO 1].

Pro hřiště na malý fotbal je potřeba kruh o polovičním průměru.

Kterým z příkazů robot nakreslí kruh polovičního průměru?



Obrázek 40: Benjamin 2008, Robot - lajnovačka

Pole odpovědí:

1. *OPAKUJ 720 [DOPŘEDU 2 VLEVO 1]*
2. *OPAKUJ 180 [DOPŘEDU 2 VLEVO 1]*
3. *OPAKUJ 360 [DOPŘEDU 1 VLEVO 1]*
4. *OPAKUJ 360 [VLEVO 1 DOPŘEDU 2]*
5. *Žádná odpověď*

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 27: Benjamin 2008, Robot - lajnovačka

Odpověď	1.	2.	3.	4.	5.
Počty	33	636	653	188	248
v %	2 %	36 %	37 %	11 %	14 %

Komentář:

Nesprávnost řešení této úlohy mohlo být způsobeno absencí algoritmického myšlení. Pochopení této datové struktury je již vcelku náročné. Pro řešitele, který nemá s programováním žádné zkušenosti, může být značně obtížné během soutěže vše pochopit a správně vyřešit.

Úloha 20: ROZSTŘÍHANÉ OBRÁZKY

Rozbor zadání:

Opět se jedná o těžkou úlohu.

Řešitelé měli z předem daných částí obrázku jejich vrstvením sestavit výsledného „smajlíka“. Nejsnazší možností, jak daný problém vyřešit, byla dekompozice výsledného obrázku. Dalo by se říci, že se jedná primárně o principy vektorové grafiky, ale dá se na tuto úlohu pohlížet i jako na algoritmickou, kdy řešitelé mají nalézt algoritmus vrstvení.

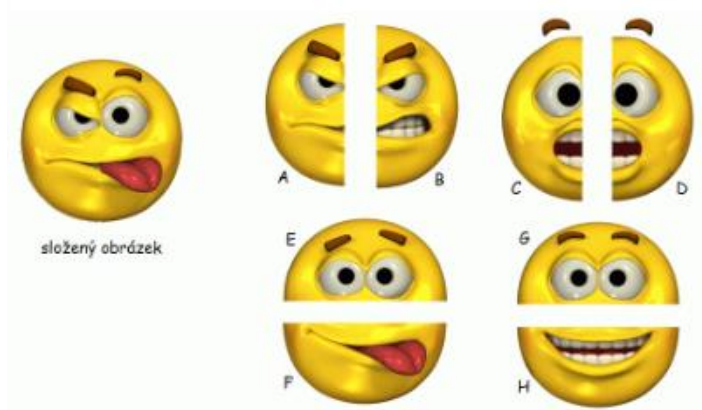
Zadání:

Ondatra bobrovi rozstříhala obrázky smajlíků a on se nyní snaží složit z nich původní obrázky.

Bobr má nyní skládáním ústřížků A, B, C, D, E, F, G, H sestavit obrázek podle předlohy. Poradíte mu tak, že napíšete, v jakém pořadí má které ústřížky na sebe pokládat. Např. ABD znamená, že se nejprve položí ústřížek A, na něj B a nakonec D. Části obrázků mohou ležet přes sebe.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Který ze zápisů vede ke správnému sestavení složeného obrázku?



Obrázek 41: Benjamin 2008, Rozstříhané obrázky

Pole odpovědí:

1. FAG
2. GAF
3. FEA
4. FGA
5. Žádná odpověď

Výsledky:

Tabulka 28: Benjamin 2008, Rozstříhané obrázky

Odpověď	1.	2.	3.	4.	5.
Počty	303	401	274	721	59
v %	17 %	23 %	16 %	41 %	3 %

Komentář:

Nesprávnost řešení této úlohy mohlo být způsobeno nedostatečnou představivostí a schopností kompozice a dekompozice.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Úloha 21: RYCHLOPRÁDELNA

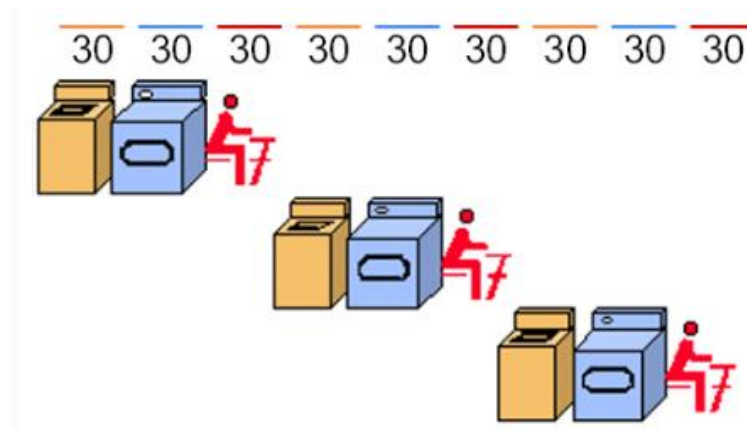
Rozbor zadání:

V případě Rychloprádelny se jednalo o středně těžkou úlohu. Z algoritmického hlediska šlo o nalezení nejefektivnějšího algoritmu využití prádelny, tj. s minimální časovou náročností. Určování efektivnosti algoritmu a vymyšlení co nejefektivnějšího algoritmu patří mezi základní programátorské dovednosti.

Zadání:

Pavel slavnostně otevřel novou prádelnu pro veřejnost se třemi stroji: pračkou, sušičkou a žehličkou. Každé zařízení bylo opatřeno časovým vypínačem, který dovolí každému zákazníkovi obsluhovat tento stroj na půl hodiny. Každý tedy potřebuje 90 minut pro vyprání, usušení a vyžehlení svého prádla.

Jestliže přijdou po sobě tři zákazníci a každý se začátkem praní počká, až předchozí z prádelny odejde, bude to celé trvat 270 minut (tato situace je znázorněna na grafu).



Obrázek 42: Benjamin 2011, Rychloprádelna - zadání

Nyní však přišli najednou tři zákazníci, kteří chvátají. Každý z nich chce své prádlo prát samostatně, shodují se však v tom, že chtějí celý proces co nejvíce urychlit.

Jak dlouho bude trvat, než všichni tři odejdou s vyžehleným prádlem?

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pole odpovědí:

1. 90 minut
2. 120 minut
3. 150 minut
4. 270 minut
5. Žádná odpověď

Výsledky:

Tabulka 29: Benjamin 2011, Rychloprádelna

Odpověď	1.	2.	3.	4.	5.
Počty	1623	535	1387	1992	542
(v %)	26 %	9 %	23 %	33 %	9 %

Komentář:

Nesprávnost řešení této úlohy byla nejspíš způsobena nepochopením zadání, nebo neuvědoměním si, že nelze nejprve prádlo žehlit a sušit, pokud není předtím vyprané, tudíž nemohou být současně od počátku využita všechna stanoviště.

Úloha 22: ŘAZENÍ KARET

Rozbor zadání:

Opět se jednalo o těžkou úlohu. Řešitel měl najít nejefektivnější algoritmus třídění karet a spočítat počet kroků, které je nutné provést k seřazení karet dle jejich hodnoty.

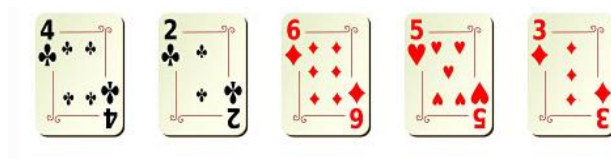
Zadání:

O přestávce mezi vyučováním si žáci hrají karetní hru – Řazení karet.

Na konci hry musí být karty seřazeny od nejmenší po největší. V jednom tahu lze prohodit pouze dvě sousední karty. Na kartách se počítají pouze hodnoty (na barvě nezáleží).

*Kolik nejméně tahů je třeba k seřazení karet **4 2 6 5 3**?*

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 43: Kadet 2010, Řazení karet

Pole odpovědí:

1. 5
2. 7
3. 6
4. 4
5. Žádná odpověď

Výsledky:

Tabulka 30: Kadet 2010, Řazení karet

Odpověď	1.	2.	3.	4.	5.
Počet	3176	164	372	1043	133
(v %)	65 %	3 %	8 %	21 %	3 %

Komentář:

Nesprávnost řešení této úlohy spočívala v nenalezení nejefektivnějšího algoritmu třídění.

Úloha 23: SKLÍZENÍ ÚRODY

Rozbor zadání:

V případě úlohy Sklizení úrody se jednalo o středně těžkou úlohu.

Tato úloha spočívala v pochopení cyklu a uvědomění si, že se stále střídají jedno a dvouřádkové záhony. Tudíž bobřík sklídí pět jednořádkových a pět dvouřádkových záhonů.

Zadání:

Maminka poslala bobříka sklídit zahrádku, aby měla z čeho vařit. Dala mu s sebou 10 semínek a řekla mu, že na každém záhonku, který sklídí, musí zasadit jedno semínko. Záhonky, které navštívil, měly střídavě jeden nebo dva řádky. Kolik řádků bobřík dohromady sklídl?

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“



Obrázek 44: Benjamin 2010, Sklizení úrody

Pole odpovědí:

1. Sklidil 15 řádků
2. Sklidil 20 řádků
3. Sklidil 10 řádků
4. Sklidil 5 řádků
5. Žádná odpověď

Výsledky:

Tabulka 31: Benjamin 2010, Sklizení úrody

Odpověď	1.	2.	3.	4.	5.
Počty	1767	556	1222	714	294
(v %)	39 %	12 %	27 %	16 %	6 %

Komentář:

Nesprávnost řešení této úlohy byla způsobena nejspíš málo pečlivým přečtením zadání a neuvědomění si opakovaného střídání různých záhonků.

Úloha 24: SLOUPEC TALÍŘŮ

Rozbor zadání:

Úloha Sloupec talířů byla označena jako lehká.

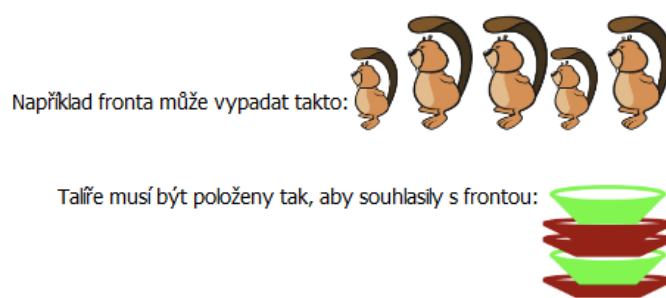
Jednalo se o nenásilnou formu seznámení žáků s datovými typy zásobníkem a frontou. Zásobník byl reprezentován sloupcem talířů a fronta zástupem bobříků. Řešitelé měli rozhodnout, který zásobník odpovídal dané frontě. Kombinovaly se zde tedy přístupy k datům

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

na principu LIFO (poslední přidaný prvek se odebírá jako první, z anglického last in first out) a FIFO (první přidaný prvek se odebírá jako první, z anglického first in first out).

Zadání:

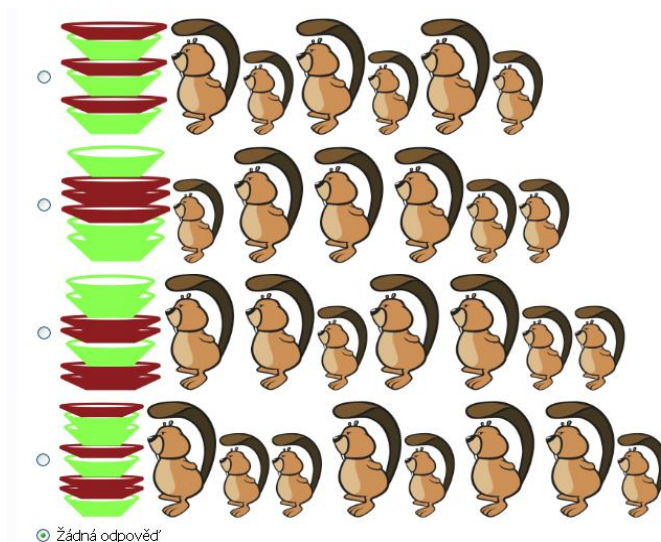
Ve školní jídelně jsou dva druhy talířů: zelené pro malé a hnědé pro velké žáky. Jednou měly kuchařky místo jen pro jeden sloupec talířů. Žáci se řadí do fronty na oběd a v kuchyni se musí dát talíře podle fronty do správného pořadí.



Obrázek 45: Benjamin 2010, Sloupec talířů - zadání

Ve kterém z následujících obrázků je nesouhlas mezi frontou a talíři?

Pole odpovědí:



Obrázek 46: Benjamin 2010, Sloupec talířů - odpovědi

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 32: Benjamin 2010, Sloupec talířů

Odpověď	1.	2.	3.	4.	5.
Počty	281	406	3437	354	75
(v %)	6 %	9 %	75 %	8 %	2 %

Komentář:

Nesprávnost řešení této úlohy byla nejspíš způsobena nepochopením principu přístupu k datům ve frontě a v zásobníku a nedostatečnou představivostí při odebrání prvků z obou datových struktur.

Úloha 25: ŠIFRA BOBRA EDUARDA

Rozbor zadání:

Opět se jednalo o těžkou úlohu.

V zadání byla dána pravidla kódování textu. Úkolem řešitele bylo aplikovat tento kódovací algoritmus na danou větu.

Zadání:

Šifra bobra Eduarda se používá tak, že

- *samohlásky a mezery se v textu nemění,*
- *každá souhláska se nahradí další souhláskou v jejich abecedě,*
- *poslední souhláska abecedy (Z) se nahradí první (B).*

*Jak vypadá zpráva: DNESKA V JEDNU ZA ZAHRADOU
zašifrovaná pomocí šifry bobra Eduarda?*

Pole odpovědí:

1. *FPFTLB W KFFPV BB BBJSBFPV*
2. *FPETLA W KEFPU BA BAJSAFOU*
3. *DNISKE V JIDNY ZE ZEHREDUY*
4. *EOFTLB W KFE OV AB ABISBEPV*
5. *Žádná odpověď*

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Výsledky:

Tabulka 33: Benjamin 2008, Šifra bobra Eduarda

Odpověď	1.	2.	3.	4.	5.
Počty	135	751	336	347	189
v %	8 %	43 %	19 %	20 %	10 %

Komentář:

Nesprávnost řešení této úlohy mohlo být způsobeno nesprávnou interpretací šifrovacího algoritmu.

Úloha 26: TALÍŘE A ČÍŠNÍK

Rozbor zadání:

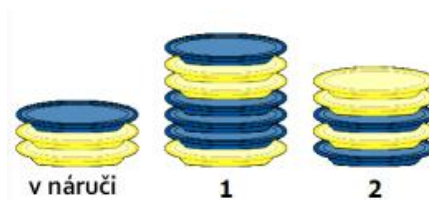
Řešení této úlohy bylo středně obtížné.

Tato úloha spočívala v aplikaci zadaného algoritmu na konkrétní případ. Jednalo se o zpětné sestavení talířů do sloupce ze dvou původních. Byl daný algoritmus tvořený čísly sloupce, určující pořadí, jak byly tyto dva sloupce vytvořeny a úkolem řešitele byl pouze opačný postup.

Zadání:

Číšník drží v náruči talíře a pokládá vždy jeden vrchní talíř na hromádku 1 nebo 2.

Pozoruje ho učeň, a protože se nudí, píše si na papírek, na kterou hromádku číšník právě talíř pokládá. Po chvíli měl napsáno $2 - 1 - 2 - 1 - 1 - 2 - 1$. A na hromádkách to vypadalo jako na obrázku:

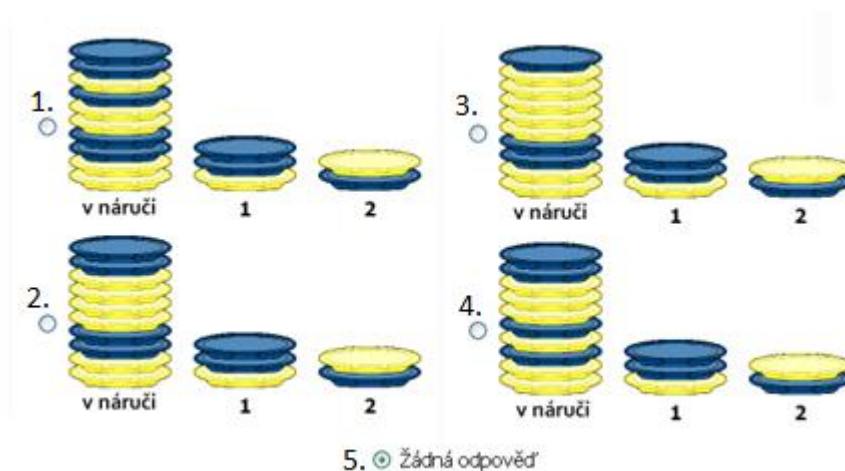


Obrázek 47: Kadet 2011, Talíře a číšník - zadání

Jak byly talíře umístěny na hromádkách předtím, než učeň začal psát čísla?

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pole odpovědí:



Obrázek 48: Kadet 2011, Talíře a číšník

Výsledky:

Tabulka 34: Kadet 2011, Talíře a číšník

Odpověď	1.	2.	3.	4.	5.
Počet	766	1847	583	1082	532
(v %)	16 %	39 %	12 %	22 %	11 %

Komentář:

Nesprávnost řešení této úlohy mohlo být způsobeno nesprávným pochopením algoritmu, nebo jeho nesprávnou interpretací.

Úloha 27: ŽELEZNICE

Rozbor zadání:

Jednalo se o těžkou úlohu.

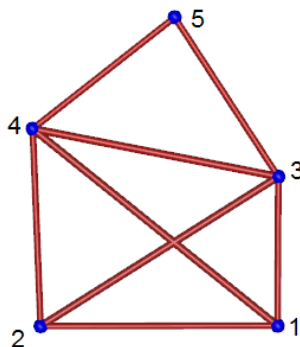
Úkolem v této úloze bylo nakreslení domečku jedním tahem. Princip této úlohy spočíval v pochopení datové struktury grafu a rozhodnutí, zda se jednalo o eulerovský graf (graf, který lze nakreslit jedním tahem) a jeho nakreslení.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Zadání:

Železniční síť spojuje 5 měst (viz plánek). Jednotlivé tratě se **křížují pouze ve městech** - mezi městy se míjejí pomocí nadjezdů a tunelů.

Je možné projet všemi tratěmi tak, aby se žádnou z nich nemuselo jet dvakrát? Ve kterém městě je třeba začít a kde skončit takovou cestu?



Obrázek 49: Benjamin 2010, Železnice

Pole odpovědí:

1. Je třeba začít i skončit v 5
2. Je třeba začít ve 4 a skončit v 3
3. Je možné začít i skončit kdekoliv
4. Je třeba začít v 2 a skončit v 1
5. Žádná odpověď

Výsledky:

Tabulka 35: Benjamin 2010, Železnice

Odpověď	1.	2.	3.	4.	5.
Počty	731	404	936	2208	274
(v %)	16 %	9 %	21 %	48 %	6 %

Komentář:

Nesprávnost řešení této úlohy mohla být způsobena nedostatečnou představivostí.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

3.1 Kategorizace

Při rozboru úloh bylo zjištěno, že u velkého množství úloh neodpovídá předpokládaná obtížnost výsledkům. Z tohoto důvodu bude provedena kategorizace úloh podle výsledků. Pro další zkoumání bude uvedena i kategorizace podle zaměření, aby bylo následně možné zjistit, zda mají například obtížné úlohy nějaký společný rys.

3.1.1 Podle obtížnosti

Následující tabulka shrnuje obtížnost jednotlivých úloh v kategorii Kadet. Jedná se o obtížnost dle autorů a podle výsledků.

Tabulka 36: Kategorizace dle obtížnosti

Pořadí	Název	Obtížnost dle zadání	Obtížnost dle výsledků
Úloha 1	Auta v garáži	Těžká	Snadná
Úloha 2	Autodráha	Těžká	Těžká A
Úloha 3	Brouk a jeho pouť	Střední/Lehká	Snadná
Úloha 4	Co je program?	Střední	Snadná
Úloha 5	Černobílé obrázky písmen	Těžká	Snadná
Úloha 6	Duhová vejce	Těžká	Střední
Úloha 7	Honba za pokladem	Těžká	Těžká B
Úloha 8	INT výpočet 0	Střední	Snadná
Úloha 9	Korálky	Těžká	Střední
Úloha 10	Nejlevnější podnájem	Střední	Těžká B
Úloha 11	Parking	Střední	Snadná
Úloha 12	Pravidla oblékání	Střední	Těžká B
Úloha 13	Procházka robota	Těžká	Těžká B
Úloha 14	Projíždka městem	Těžká	Těžká A

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pořadí	Název	Obtížnost dle zadání	Obtížnost dle výsledků
Úloha 15	Přehazování písmen	Lehká	Těžká A
Úloha 16	Ranní běh	Těžká	Snadná
Úloha 17	Razítkování	Lehká	Snadná
Úloha 18	RGB mřížka	Lehká	Snadná
Úloha 19	Robot - lajnovačka	Těžká	Těžká B
Úloha 20	Rozstříhané obrázky	Těžká	střední
Úloha 21	Rychloprádelna	Těžká	Těžká A
Úloha 22	Řazení karet	Těžká	Snadná
Úloha 23	Sklízení úrody	Střední	Střední
Úloha 24	Sloupec talířů	Lehká	Snadná
Úloha 25	Šifra bobra Eduarda	Těžká	Střední
Úloha 26	Talíře a číšník	Střední	Střední
Úloha 27	Železnice	Těžká	Střední

Sloupec **Obtížnost dle zadání** říká, jaká byla obtížnost úlohy podle autorů. Sloupec **Obtížnost dle výsledků** udává, jak bylo obtížné řešení podle procentuálního zastoupení správných a špatných odpovědí. Bližší specifikace podmínek pro zařazení do jednotlivých obtížnostních skupin je popsána kapitole věnované metodice výzkumu.

3.1.2 Podle zaměření

Každá úloha měla své zaměření. Jedná se například o úlohy algoritmické, o interpretaci algoritmu, o úlohy se zaměřením na datové struktury, programátorské aj. Následující tabulka shrnuje zaměření jednotlivých úloh.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Tabulka 37: Kategorizace dle zaměření

Pořadí	Název	Zaměření	Bližší specifikace
Úloha 1	Auta v garáži	Datové struktury	Zásobník
Úloha 2	Autodráha	Algoritmická	Sestavení alg.
Úloha 3	Brouk a jeho pouť	Algoritmická	Interpretace algoritmu
Úloha 4	Co je program?	Algoritmická	Vlastnosti alg.
Úloha 5	Černobílé obrázky písmen	Algoritmická	Interpretace kódovacího alg.
Úloha 6	Duhová vejce	Algoritmická	Interpretace algoritmu
Úloha 7	Honba za pokladem	Algoritmická	Interpretace algoritmu
Úloha 8	INT výpočet 0	Datové typy	Celočíselný DT
Úloha 9	Korálky	Datové struktury	Zásobník
Úloha 10	Nejlevnější podnájem	Datové struktury	Graf
Úloha 11	Parking	Algoritmická	Sestavení algoritmu
Úloha 12	Pravidla oblékání	Algoritmická	Vývojový diagram
Úloha 13	Procházka robota	Programátorská	Sestavení programu
Úloha 14	Projížďka městem	Algoritmická	Chyba v grafickém alg.
Úloha 15	Přehazování písmen	Datové struktury	Algoritmus třídění
Úloha 16	Ranní běh	Programátorská	Interpretace programu
Úloha 17	Razítkování	algoritmická	Algoritmus vrstvení
Úloha 18	RGB mřížka	Algoritmická	Cykly
Úloha 19	Robot - lajnovačka	programátorská	Modifikace algoritmu
Úloha 20	Rozstřihané obrázky	Algoritmická	Algoritmus vrstvení
Úloha 21	Rychloprádelna	Algoritmická	Časová náročnost

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pořadí	Název	Zaměření	Bližší specifikace
Úloha 22	Řazení karet	Datové struktury	Algoritmy třídění
Úloha 23	Sklízení úrody	Algoritmická	Cykly
Úloha 24	Sloupec talířů	Datové struktury	Zásobník, fronta
Úloha 25	Šifra bobra Eduarda	Algoritmická	Interpretace kódovacího alg.
Úloha 26	Talíře a číšník	Datové struktury / algoritmická	Zásobník / interpretace alg.
Úloha 27	Železnice	Datové struktury	Graf

3.2 Analýza kategorií

V této kapitole budou analyzovány jednotlivé úlohy zařazené do stejných obtížnostních kategorií. Bude snaha o nalezení společných rysů úloh zařazení do stejné obtížnostní kategorie. Jedná se o rysy jako například podobná délka zadání, práce s obrázkem, tematické zaměření, časová náročnost řešení, délka odpovědi aj.

3.2.1 Snadné úlohy

Do této kategorie byly zařazeny následující úlohy:

Tabulka 38: Snadné úlohy

Pořadí	Název	Obtížnost dle zadání	Zaměření
Úloha 1	Auta v garáži	Těžká	Datové struktury
Úloha 3	Brouk a jeho pouť	Střední/Lehká	Algoritmická
Úloha 4	Co je program?	Střední	Algoritmická
Úloha 5	Černobílé obrázky písmen	Těžká	Algoritmická
Úloha 8	INT výpočet 0	střední	Datové typy
Úloha 11	Parking	Střední	Algoritmická

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

Pořadí	Název	Obtížnost dle zadání	Zaměření
Úloha 16	Ranní běh	Těžká	Programátorská
Úloha 17	Razítkování	Lehká	Algoritmická
Úloha 18	RGB mřížka	Lehká	Algoritmická
Úloha 22	Řazení karet	Těžká	Datové struktury
Úloha 24	Sloupec talířů	Lehká	Datové struktury

Do této kategorie bylo zařazeno celkem jedenáct úloh. Z tabulky se dá vyvodit, že tyto úlohy nemají stejnou obtížnost podle autorů, ani nemají stejné zaměření.

Jedná se o otázky s krátkým zadáním i s dlouhým, otázky obrázkové i bez obrázků atd.

U této kategorie se nedají nalézt žádné společné rysy.

3.2.2 Středně těžké úlohy

Mezi algoritmickými úlohami bylo nalezeno šest středně těžkých úloh. Jejich seznam je uveden v následující tabulce:

Tabulka 39: Středně těžké úlohy

Pořadí	Název	Obtížnost dle zadání	Zaměření
Úloha 6	Duhová vejce	Těžká	Algoritmická
Úloha 9	Korálky	Těžká	Datové struktury
Úloha 20	Rozstříhané obrázky	Těžká	Algoritmická
Úloha 23	Sklízení úrody	Střední	Algoritmická
Úloha 25	Šifra bobra Eduarda	Těžká	Algoritmická
Úloha 26	Talíře a číšník	Střední	Datové struktury / algoritmická
Úloha 27	Železnice	Těžká	Datové struktury

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

V této kategorii je více než 70 % úloh dle autorů těžkých.

Zaměřením se tyto úlohy různí, jedná se o úlohy vyloženě algoritmické i úlohy více zaměřené na datové struktury.

Více než 70 % úloh je v zadání doplněno obrázkem, se kterým se musí dále pracovat.

U šesti z těchto úloh se při řešení dané úlohy nějak pracuje s pořadím. Ať se již jedná o pořadí korálků na niti, talířů v zásobníku, pořadí kroků při kresbě eulerovského grafu, vrstvení obrázků, obarvení vajec, nebo pořadí záhonků. Pouze u jediné úlohy – Šifra bobra Eduarda – se s pořadím dílčích kroků nesetkáme. Zde se jedná o aplikaci kódovacího algoritmu pro daný případ.

3.2.3 Těžké úlohy

Do této obtížnostní supiny bylo zařazeno celkem devět úloh. Tyto úlohy byly ještě dále rozděleny do dvou dílčích skupin. Jedná se o skupinu A, kde dominuje jedna nesprávná odpověď. Druhá skupina (B) se vyznačuje dvěma dominantními odpověďmi. Jedna z těchto odpovědí je správná a druhá chybná. U čtyř z pěti úloh mírně převažuje správná odpověď.

U sedmi z těchto úloh se nadprůměrný počet žáků vyhnul řešení těchto otázek. Celkový průměr nezodpovězených algoritmických otázek činil osm procent, u těchto úloh dosáhl tento průměr téměř jedenácti procent.

Těžké A:

Tabulka 40: Těžké úlohy A

Pořadí	Název	Obtížnost dle zadání	Zaměření
Úloha 2	Autodráha	Těžká	Algoritmická
Úloha 14	Projížďka městem	Těžká	Algoritmická
Úloha 15	Přehazování písmen	Lehká	Datové struktury
Úloha 21	Rychloprádelna	Těžká	Algoritmická

Tyto otázky byly nejspíš velmi složité a pro řešitele nepochopitelně zadané a ti je z tohoto důvodu nepřesně interpretovali. Procento špatných odpovědí u těchto otázek přesáhlo čtyřicet procent. Zatímco počet správně odpovídajících nepřesáhl dvacet pět procent.

E. Celorepublikový výzkum úspěšnosti v soutěži „Bobřík informatiky“

U všech těchto úloh se během řešení muselo pracovat s obrázkem.

Zajímavostí je, že autoři u jedné z těchto úloh předpokládali lehké řešení. Mezi algoritmickými úlohami byla ještě jedna zaměřená na třídící algoritmy. Tato úloha měla obdobný způsob řešení, ale její výsledky se od této velice lišily. Zatímco u úlohy Řazení karet bylo více než šedesát procent správných odpovědí, u úlohy Přehazování písmen jich bylo jen necelých třicet tři procent.

Těžké B:

Tabulka 41: Těžké úlohy B

Pořadí	Název	Obtížnost dle zadání	Zaměření
Úloha 7	Honba za pokladem	Těžká	Algoritmická
Úloha 10	Nejlevnější podnájem	Střední	Datové struktury
Úloha 12	Pravidla oblékání	Střední	Algoritmická
Úloha 13	Procházka robota	Těžká	Programátorská
Úloha 19	Robot - lajnovačka	Těžká	Programátorská

Jediný společný rys všech těchto úloh spočíval v delším řešení. Řešení všech těchto úloh trvá relativně dlouho a je velmi jednoduché se někde splést. Při jednom chybném kroku dojde sice k vyřešení celé úlohy, ale již ne ke správnému.

Diskuse

Tato práce se zabývá současným stavem výuky algoritmizace na základních školách. Na základě několika částí výzkumu bylo zjištěno následující:

V Českých Budějovicích působí šestnáct základních škol a algoritmizace se nevyučuje ani na jediné z nich. Na dvou místních školách se vyučuje alespoň programování, a to v rámci volitelného předmětu. S programováním se tedy seznámí pouze zájemci o danou problematiku, a to jen pokud je jiný nabízený předmět nenadchne více. Jedna z těchto škol je přímo zaměřená na výuku programování a při škole funguje volitelný kroužek. Tento kroužek je na přání rodičů nabízen již v první třídě. Dalo by se říci, že tato škola je svým zaměřením velmi ojedinělá.

V Českých Budějovicích také existuje zájmový kroužek při DDM, kde se věnují konstrukci robotů. Jedná se o konstrukci robotů po mechanické, elektronické i softwarové stránce. Zde se děti také věnují programování. Jedinou nevýhodou tohoto kroužku je, že je určen dětem, které už mají s programováním nějaké zkušenosti.

V celorepublikovém měřítku se dá říci, že se algoritmizace a programování příliš zájmu nevěnuje. Výskyt škol, kde se programování vyučuje je velmi ojedinělý. V celé republice se podařilo nalézt pouze padesátku škol, kde se této problematice věnují. Tato část výzkumu je pouze orientační, neboť se vycházelo pouze z informací uvedených na internetu. Proto tato část nemá tak velkou vypovídací hodnotu. Mnohem přesnějších výsledků by jistě bylo docíleno, pokud by se jednotlivé školy dotazovaly přímo. Tímto směrem by se dalo také dále v tomto výzkumu pokračovat.

Z výše uvedeného se dá vyvodit, že České Budějovice patří mezi několik málo měst, kde se programování vyučuje. Výskyt dvou škol v jednom městě je nadprůměrný. Pouze v Praze, Hradci Králové a Brně bylo nalezeno více škol, které se věnují výuce programování. Počítačově zaměřených základních škol je v České Republice jen několik a v Českých Budějovicích je dokonce škola zaměřená vysloveně na programování. Žáci této školy jsou i velmi úspěšní v soutěžích.

Co se týče využití učebnice při výuce, zde je situace velmi špatná. Na trhu je v současné době pouze jediná učebnice přímo určená pro základní školy, kde by byla programování věnována pozornost. Bylo zjištěno, že při výuce práce s počítačem se učebnice téměř nevyužívají. Pokud jsou využívány, má je k dispozici většinou pouze učitel. Plošné rozšíření

Diskuse

dané učebnice, která se věnuje programování, se nepodařilo nalézt a tento cíl nemohl být splněn. Daná učebnice se věnuje výuce programování v jazyce LOGO, které ani není nejčastěji používaným nástrojem pro výuku. Nejčastěji používaným nástrojem je Baltík, ke kterému jsou všechny materiály na internetu.

Další část výzkumu patřila soutěži Bobřík informatiky. Ze všech již proběhlých ročníků soutěže byly vybrány algoritmické úlohy z kategorií Benjamin a Kadet. Tyto kategorie jsou určeny pro žáky základních škol. Bylo zjištěno, že téměř třetina všech úloh v dané soutěži nějak souvisí s algoritmizací a programováním. Jednalo se zejména o úlohy zaměřené na interpretaci a sestavení algoritmu, práci s datovými strukturami jako například zásobník a fronta, vývojové diagramy, úpravy programu, určování výstupu programu aj.

Součástí byl i rozbor jednotlivých algoritmických úloh. Zde bylo zjištěno, že chybné řešení nebylo v mnoha případech způsobeno absencí či nedostatečně rozvinutým algoritmickým myšlením, ale pouze nepozorností při prostudování zadání, jeho chybnou interpretací, či příliš zbrklým výběrem odpovědi. Dle četnosti odpovědí u některých otázek se dá také usuzovat, že zadání bylo pro řešitele matoucí.

Z rozboru těchto úloh vyplynulo, že předpokládaná obtížnost otázek velmi často neodpovídá výsledkům. Například úloha, která byla označena jako těžká, měla v několika případech více než nadpoloviční převahu správných odpovědí a naopak zde byla i lehká otázka, kde byla pouze třetina správných odpovědí. Následnou kategorizací úloh dle výsledků nebyl nalezen žádný markantní znak, který by například nejtěžší úlohy spojoval. Jediným podstatnějším zjištěním bylo, že se jejich řešení vyhnulo většímu množství řešitelů. Tento rozdíl činil tři procenta (průměr u všech algoritmických úloh byl osm procent).

Výše zjištěné poznatky jsou zcela v souladu s předchozími zjištěními. Například Dr. Pitner se již více než před deseti lety zmiňuje o úpadku výuky algoritmizace na základních školách. V současné době však roste tlak na počítačovou gramotnost a tomuto by se také měly základní školy přizpůsobit a zvážit zařazení například programování do svých programů. Tomuto tlaku také odpovídá vznik různých projektů, které mají napomoci zařazení programování a algoritmizace do výuky. Jedná se například o projekt PROŠ, jehož nositelem je Univerzita Palackého v Olomouci, nebo Tvořivý učitel tvořivé informatiky a Mladí vývojáři. Všechny tyto projekty mají pomoci učitelům, aby byli schopni programování vyučovat. Jedná se například o kurzy, podporu výuky aj.

Diskuse

Výše zjištěné pouze částečně odpovídá hypotézám, které byly na začátku této práce předpokládány.

Jako první byla položena hypotéza, že se algoritmizaci věnuje malé procento českobudějovických základních škol. Pokud tato hypotéza bude brána doslova, pak byla zcela chybná, neboť výuce algoritmizace se nevěnuje žádná ze škol. Pokud je ale do této hypotézy zahrnuto i programování, pak lze říci, že byla splněna. V Českých Budějovicích se totiž věnuje výuce programování více než dvanáct procent základních škol.

Druhá hypotéza se vztahovala k zájmovým kroužkům. Předpokladem bylo, že se zde pořádá více kroužků zaměřených na výuku algoritmizace a programování. V tomto ohledu byl nalezen pouze jediný kroužek. Tento kroužek se primárně ani programování nevěnuje a jeho největším handicapem je, že je určen pouze dětem se základy programování. Tato hypotéza byla tedy z větší části chybná.

Následovala hypotéza věnující se celorepublikové výuce algoritmizace. Tato hypotéza se ukázala také jako ne příliš přesná. Koncentrace škol věnujících se programování je vcelku velká zejména v Praze a Brně, ale ostatní města na tom již tak dobře nejsou a je spíše výjimkou, než pravidlem, že by se algoritmizace a programování vyučovalo.

Další hypotéza se věnovala učebnicím. Byl předpoklad, že je na českém trhu větší množství učebnic výpočetní techniky pro ZŠ. Tento předpoklad sice splněn je, ale výuce algoritmizace se nevěnuje ani jediná z nich a programování pouze jedna. A nalézt vyučujícího, který by při výuce tuto učebnici aktivně používal se, jak již bylo řečeno, nepodařilo. Pokud ji učitelé používají, mají většinou pouze jeden výtisk a využívají ji pouze pro svou potřebu, nebo jako inspiraci pro svou výuku, ale školu, kde by s učebnicí pracovali i žáci, se nalézt nepodařilo.

Poslední hypotéza se zabývala úspěšností v soutěži Bobřík informatiky. Bylo zjištěno, že úspěšnost řešení algoritmických úloh je o pět procent nižší, než je celosoutěžní průměr. Předpokladem bylo, že tyto výsledky jsou stejné. Tento předpoklad by se dal považovat za téměř správný. Rozdíl pěti procent v tomto případě není až tak výrazný.

Celkově lze říci, že stanovené hypotézy vyzněly optimističtěji, než jsou skutečné závěry.

Shrnutí a závěr

V současné přetechizované době rostou nároky na počítačovou gramotnost. Algoritmické myšlení je její nedílnou součástí. Výuka algoritmizace podporuje nejen algoritmické myšlení, ale i zefektivňuje vyhledávání na internetu, rozvíjí tvořivost...

Výuka algoritmizace na základních školách je téměř zanedbatelná. Z tohoto důvodu nebylo možné splnit původní cíle této práce a bylo nutné je modifikovat. Modifikované cíle se podařilo z větší části splnit. Jediným nesplněným cílem bylo nalezení učitele využívající při své výuce učebnici. Z původních cílů nebylo splněno nalezení vzorku deseti škol, druhá část rozbor přístupů k výuce byl splněn alespoň u dvou škol, které v ČB programování vyučují.

Přínosem této práce je zhodnocení současného stavu výuky algoritmizace. Byla snaha o komplexní pohled na danou problematiku. Přínosem do budoucna by mohlo být po přečtení této práce zamyšlení nad současným stavem a snaha současných i budoucích učitelů výpočetní techniky o zařazení algoritmizace a programování do jejich výuky.

V tomto výzkumu je vhodné i nadále pokračovat a monitorovat, zda se opět algoritmizace posune více do popředí zájmu, nebo bude i nadále škol, kde se algoritmizace vyučuje, ubývat. Jak řekl dr. Pitner o výuce algoritmizace: „*Historie: Cesta ze středu pozornosti na periferii zájmu?*“

Seznam obrázků

Seznam obrázků

<i>Obrázek 1: Příklad slovního algoritmu</i>	6
<i>Obrázek 2: Vývojový diagram – výběr největšího čísla ze tří přirozených čísel</i>	7
<i>Obrázek 3: Karel 3D pro Windows</i>	19
<i>Obrázek 4: Imagine Logo</i>	20
<i>Obrázek 5: Baltík</i>	21
<i>Obrázek 6: Petr</i>	22
<i>Obrázek 7: Scratch</i>	23
<i>Obrázek 8: Graf - výsledky výzkumu A</i>	29
<i>Obrázek 9: MERKUR: Robotický Slídl ALFA</i>	37
<i>Obrázek 10: LEGO MINDSTORMS NXT 2.0</i>	38
<i>Obrázek 11: Mapa republiky s vyznačenými školami, kde se vyučuje algoritmizace</i>	39
<i>Obrázek 12: Podíl jednotlivých vývojových nástrojů na výuce</i>	40
<i>Obrázek 13: Tematické složení úloh</i>	43
<i>Obrázek 14: Průměrná úspěšnost řešení úloh</i>	45
<i>Obrázek 15: Průměrná úspěšnost v kategorii Benjamin</i>	47
<i>Obrázek 16: Průměrná úspěšnost v kategorii Kadet</i>	48
<i>Obrázek 17: Benjamin 2009, Auta v garáži - zadání</i>	49
<i>Obrázek 18: Benjamin 2009, Auta v garáži - odpovědi</i>	49
<i>Obrázek 19: Kadet 2010, Autodráha</i>	51
<i>Obrázek 20: Kadet 2010, Brouk a jeho pouť</i>	52
<i>Obrázek 21: Benjamin 2011, Černobílé obrázky písmen</i>	55
<i>Obrázek 22: Benjamin 2009, Duhová vejce - zadání</i>	56
<i>Obrázek 23: Benjamin 2009, Duhová vejce - řešení</i>	57
<i>Obrázek 24: Benjamin 2011, Honba za pokladem</i>	58
<i>Obrázek 25: Benjamin 2009, INT výpočet 0</i>	59
<i>Obrázek 26: Benjamin 2009, Korálky – zadání 1</i>	60
<i>Obrázek 27: Benjamin 2009, Korálky – zadání 2</i>	60
<i>Obrázek 28: Benjamin 2009, Korálky - odpovědi</i>	61

Seznam obrázků

<i>Obrázek 29: Benjamin 2009, Nejlevnější podnájem</i>	62
<i>Obrázek 30: Benjamin 2008, Parking</i>	63
<i>Obrázek 31: Benjamin 2011, Pravidla oblékání</i>	65
<i>Obrázek 32: Benjamin 2011, Pravidla oblékání - odpovědi</i>	65
<i>Obrázek 33: Kadet 2011, Procházka robota</i>	67
<i>Obrázek 34: Benjamin 2010, Projížďka městem</i>	68
<i>Obrázek 35: Benjamin 2011, Přehazování písmen</i>	69
<i>Obrázek 36: Benjamin 2009, Razítkování - zadání</i>	71
<i>Obrázek 37: Benjamin 2009, Razítkování - řešení</i>	72
<i>Obrázek 38: Benjamin 2011, RGB mřížka - zadání</i>	73
<i>Obrázek 39: Benjamin 2011, RGB mřížka - odpovědi</i>	74
<i>Obrázek 40: Benjamin 2008, Robot - lajnovačka</i>	75
<i>Obrázek 41: Benjamin 2008, Rozstříhané obrázky</i>	77
<i>Obrázek 42: Benjamin 2011, Rychloprádelna - zadání</i>	78
<i>Obrázek 43: Kadet 2010, Řazení karet</i>	80
<i>Obrázek 44: Benjamin 2010, Sklizení úrody</i>	81
<i>Obrázek 45: Benjamin 2010, Sloupec talířů - zadání</i>	82
<i>Obrázek 46: Benjamin 2010, Sloupec talířů - odpovědi</i>	82
<i>Obrázek 47: Kadet 2011, Talíře a číšník - zadání</i>	84
<i>Obrázek 48: Kadet 2011, Talíře a číšník</i>	85
<i>Obrázek 49: Benjamin 2010, Železnice</i>	86

Seznam tabulek

Seznam tabulek

<i>Tabulka 1: úroveň chápání algoritmických postupů žáky ZŠ</i>	14
<i>Tabulka 2 : Počet vzdělávacích institucí v ČB</i>	28
<i>Tabulka 3: Výsledky výzkumu A</i>	28
<i>Tabulka 4: Rozbor institucí vyučujících algoritmizaci</i>	29
<i>Tabulka 5: Bobřík inforamtky - počet soutěžících</i>	42
<i>Tabulka 6: Bobřík inforamtky - počet algoritmických úloh v ročnících</i>	43
<i>Tabulka 7: Průměrná úspěšnost v kategoriích</i>	44
<i>Tabulka 8: Průměrná úspěšnost v jednotlivých ročnících</i>	46
<i>Tabulka 9: Benjamin 2008, Auta v garáži</i>	50
<i>Tabulka 10: Kadet 2010, Autodráha</i>	51
<i>Tabulka 11: Benjamin 2010, Brouk a jeho pouť</i>	53
<i>Tabulka 12: Kadet 2011, Co je program?</i>	54
<i>Tabulka 13: Benjamin 2011, Černobílé obrázky písmen</i>	55
<i>Tabulka 14: Benjamin 2009, Duhová vejce</i>	57
<i>Tabulka 15: Benjamin 2011, Honba za pokladem</i>	58
<i>Tabulka 16: Benjamin 2009, INT výpočet 0</i>	60
<i>Tabulka 17: Benjamin 2009, Korálky</i>	61
<i>Tabulka 18: Benjamin 2009, Nejlevnější podnájem</i>	62
<i>Tabulka 19: Benjamin 2008, Parking</i>	64
<i>Tabulka 20: Benjamin 2011, Pravidla oblékání</i>	66
<i>Tabulka 21: Kadet 2011, Procházka robota</i>	67
<i>Tabulka 22: Benjamin 2010, Projížďka městem</i>	69
<i>Tabulka 23: Benjamin 2011, Přehazování písmen</i>	70
<i>Tabulka 24: Kadet 2010, Ranní běh</i>	71
<i>Tabulka 25: Benjamin 2009, Razítkování</i>	72
<i>Tabulka 26: Benjamin 2011, RGB mřížka</i>	74
<i>Tabulka 27: Benjamin 2008, Robot - lajnovačka</i>	76
<i>Tabulka 28: Benjamin 2008, Rozstříhané obrázky</i>	77

Seznam tabulek

<i>Tabulka 29: Benjamin 2011, Rychloprádelna</i>	<i>79</i>
<i>Tabulka 30: Kadet 2010, Řazení karet</i>	<i>80</i>
<i>Tabulka 31: Benjamin 2010, Sklizení úrody.....</i>	<i>81</i>
<i>Tabulka 32: Benjamin 2010, Sloupec talířů</i>	<i>83</i>
<i>Tabulka 33: Benjamin 2008, Šifra bobra Eduarda</i>	<i>84</i>
<i>Tabulka 34: Kadet 2011, Talíře a číšník</i>	<i>85</i>
<i>Tabulka 35: Benjamin 2010, Železnice</i>	<i>86</i>
<i>Tabulka 36: Kategorizace dle obtížnosti</i>	<i>87</i>
<i>Tabulka 37: Kategorizace dle zaměření</i>	<i>89</i>
<i>Tabulka 38: Snadné úlohy</i>	<i>90</i>
<i>Tabulka 39: Středně těžké úlohy.....</i>	<i>91</i>
<i>Tabulka 40: Těžké úlohy A</i>	<i>92</i>
<i>Tabulka 41: Těžké úlohy B</i>	<i>93</i>

Seznam použitých zkratek a pojmů

Seznam použitých zkratek a pojmů

<i>tj.</i>	To jest
<i>tzv.</i>	Tak zvané
<i>atd.</i>	A tak dále
<i>aj.</i>	A jiné
<i>např.</i>	Například
<i>vs.</i>	Versus
<i>resp.</i>	Respektive
<i>evt.</i>	Eventuálně
<i>alg.</i>	Algoritmus
<i>VT</i>	Výpočetní technika
<i>RVP ZV</i>	Rámcový vzdělávací program pro základní vzdělávání
<i>RVP G</i>	Rámcový vzdělávací program pro gymnázia
<i>ŠVP SR</i>	Štátny vzdelávací program (Slovenské RVP)
<i>IKT</i>	informační a komunikační technologie
<i>MŠMT</i>	Ministerstvo školství, mládeže a tělovýchovy
<i>DDM</i>	Dům dětí a mládeže
<i>DT</i>	Datový typ

Seznam použitých zdrojů

Seznam použitých zdrojů

BALADA, Jan, aj. 2007a. **RÁMCOVÝ VZDĚLÁVACÍ PROGRAM PRO ZÁKLADNÍ VZDĚLÁVÁNÍ.** [online]. 8-8-2007, [cit. 2011-03-26]. Dostupný z: http://www.vuppraha.cz/wp-content/uploads/2009/12/RVPZV_2007-07.pdf

BALADA, Jan, aj. 2007b. **RÁMCOVÝ VZDĚLÁVACÍ PROGRAM PRO GYMNÁZIA.** [online]. 13-9-2007, [cit. 2011-09-05]. Dostupný z: http://clanky.rvp.cz/wp-content/upload/prilohy/9639/rvp_g.pdf

BERKI, Jan. 2009. **STRASTI DIDAKTIKA INFORMATIKY.** *Sborník konference Počítač ve škole 2009* [online]. 20-03-2009, [cit. 2011-05-20]. Dostupný z: <http://gynome.nmnm.cz/konference/files/2009/sbornik/berki.pdf>

BLICHOVÁ, Slávka, ŠESTÁKOVÁ, Eva. 2008. **ŠTÁTNY VZDELÁVACÍ PROGRAM,** Inforamtika, ISCED 2. [online]. 2008, [cit. 2011-09-02]. Dostupný z: http://www.iedu.sk/vyucovanie_a_studium/vyucovacie_predmety/xKatalog_Dokumenty/Informatika%20ISCED%202.pdf

ČECHOVÁ HUMPOLCOVÁ, Tereza. 2012. **SCRATCH VE VÝUCE.** [online]. 26-01-2012, [cit. 2012-02-16]. Dostupné z: <http://spomocnik.rvp.cz/clanek/14905/SCRATCH-VE-VYUCE.html>

DDM ČESKÉ BUDĚJOVICE. 2010. **ROBOZOR:** Robozor - o nás. [online]. 2010 [cit. 2012-04-08]. Dostupné z: <http://www.robozor.cz/klub-robozor-o-nas/>

DOHNAL, Pavel. 2009. **PROGRAMOVÁNÍ NA ZÁKLADNÍCH ŠKOLÁCH.** [online]. Diplomová práce 2009, [cit. 2011-02-15]. Dostupný z: http://is.muni.cz/th/72886/pedf_m/Diplomova_prace_Pavel_Dohnal.pdf

FUKOVÁ, Alena. 2007. **ŠKOLNÍ VZDĚLÁVACÍ PROGRAM PRO ZÁKLADNÍ VZDĚLÁVÁNÍ:** Základní škola a Mateřská škola, Nerudova 9, České Budějovice. [online]. 20-6-2007 [cit. 18-2-2012]. Dostupný z: http://www.zsnerudova.cz/texty/10_11/SVP_zari_2010.pdf

HOŠPESOVÁ, Alena. 2011. **DOTAZNÍK, ROZHOVOR.** [online]. 2-12-2011, [cit. 2011-12-26]. Dostupný z: http://eamos.pf.jcu.cz/amos/kat_ped/externi/kat_ped_76269/5_Dotaznik_rozhovor.pdf

Seznam použitých zdrojů

HRONOVÁ, Milada. 2011. **PROJEKT PEDAGOGICKÉ FAKULTY PŘINESE ZÁKLADY PROGRAMOVÁNÍ DO ŠKOL.** *Žurnál UP* [online]. 10-3-2011, [cit. 2011-03-26]. Dostupný z: <http://www.zurnal.upol.cz/aktualita/clanek/projekt-pedagogicke-fakulty-prinese-zaklady-programovani-do-skol/>

CHRÁSKA, Miroslav. 2007. **METODY PEDAGOGICKÉHO VÝZKUMU: základy kvantitativního výzkumu.** Vydání 1. Praha: Grada Publishing, 2007, 265 s. ISBN 978-80-247-1369-4.

KLABAČKA, Radek. 2011. **ALGORITMIZACE A ZÁKLADY PROGRAMOVÁNÍ.** [online]. 15-02-2011, [cit. 2011-03-05]. Dostupný z: <http://www.jardaz.cz/soubory/referaty/25/klabacka.doc>

KLÍMA, Karel. **PROGRAMOVACÍ JAZYK KAREL: CO JE KAREL.** [online]. [cit. 2011-12-27]. Dostupné z: <http://karel.webz.cz/co-je-karel>

KRAUS, Josef. 2011. **NÁSTROJ PETR: PROGRAMOVÁNÍ I PRO ZAČÁTEČNÍKY.** *Živě.cz* [online]. 20-09-2011, [cit. 2011-12-27]. Dostupné z: <http://www.zive.cz/clanky/nastroj-petr-programovani-i-pro-zacatecniky/sc-3-a-158822/default.aspx>

KRČEK, Břetislav, KREML, Pavel. 1993. **ALGORITMIZACE A PROGRAMOVÁNÍ V JAZYKU PASCAL.** 1. vyd. Ostrava: VŠB – Technická univerzita, 1993.

KRONICK, JANE C. 1997. **ALTERNATIVNÍ METODOLOGIE PRO ANALÝZU KVALITATIVNÍCH DAT.** *Sociologický časopis: Metodologická rubrika* [online]. 1997, [cit. 2011-12-26]. Dostupné z: http://sreview.soc.cas.cz/uploads/d9ea8b94ec58c839306f61835d149215d1a90b3f_276_057KRONI.pdf

MELICHAR, Jan. 2006. **ALGORITMY NA 1. STUPNI ZÁKLADNÍ ŠKOLY.** [online]. 20-12-2006, [cit. 2011-02-15]. Dostupný z: http://pf.ujep.cz/files/KMA_poznamkydidamat04.pdf

MŠMT. 2007. **RÁMCOVÝ VZDĚLÁVACÍ PROGRAM PRO OBOR VZDĚLÁNÍ: 2345L01 MECHANIK SEŘIZOVAČ.** [online]. 16-4-2007, [cit. 2011-09-05]. Dostupný z:

Seznam použitých zdrojů

<http://ebookbrowse.com/rvp-2345101-mechanik-serizovac-pdf-d71922679>

PITNER, Tomáš. 2000. **VÝUKA PROGRAMOVÁNÍ NA ZÁKLADNÍ A STŘEDNÍ ŠKOLE.** [online]. 13-3-2000, [cit. 2011-04-15]. Dostupný z: http://www.fi.muni.cz/~tomp/semuc/text_pitner.html

PRŮCHA, Jan, Eliška WALTEROVÁ a Jiří MAREŠ. 2003. **PEDAGOGICKÝ SLOVNÍK.** 4., aktualiz. vyd. Praha: Portál, 2003, 322 s. ISBN 80-717-8772-8.

PŘIBYL, Vladimír. 2012. **BOBŘÍK INFORMATIKY** [online]. 3/2012 [cit. 2012-03-03]. Dostupné z: <http://www.ibobr.cz/>

PŠENČÍKOVÁ, Jana. 2009. **ALGORITMIZACE.** 2. vyd. Kralice na Hané: Computer Media, 2009. ISBN 978-80-7402-034-6.

SGP systems. a. **SGP BALTÍK 3.** [online]. [cit. 2011-12-27]. Dostupné z: http://www.sgp.cz/cz/Product_B3.asp

SGP systems. b. **SGP BALTIE 4 C#.** [online]. [cit. 2011-12-27]. Dostupné z: http://www.sgp.cz/cz/Product_B4.asp

STRAKA, Jaromír, et al. 2007. **ŠKOLNÍ VZDĚLÁVACÍ PROGRAM PRO ZÁKLADNÍ VZDĚLÁVÁNÍ.** [online]. 31-8-2007, [cit. 2011-03-26]. Dostupný z: www.zshanusovice.cz/uredni_deska/SVP.doc

TIB, občanské sdružení. 2008. **TVOŘIVÝ UČITEL TVOŘIVÉ INFORMATIKY.** [online]. 31-8-2008, [cit. 2012-02-15]. Dostupný z: <http://www.tib.cz/tvorivyucitel/projekt.htm>

TLAPÁKOVÁ, Helena. 2010. **ŠKOLNÍ VZDĚLÁVACÍ PROGRAM PRO ZÁKLADNÍ VZDĚLÁVÁNÍ: Základní škola, L. Kuby 48, České Budějovice.** [online]. 1-10-2010 [cit. 16-2-2012]. Dostupný z: <http://www.zsroznov.cz/dokumenty/svp.pdf>

VANÍČEK, Jiří. 2011. **IMAGINE LOGO: Web na podporu výuky.** [online]. 2011, [cit. 2012-02-16]. Dostupné z: <http://www.pf.jcu.cz/imagine/>

ZŠ L.Kuby. 2011. **Baltík a výuka na škole: Jak využíváme tyto programy na naší škole?.** **BALTÍK V ROŽNOVĚ** [online]. 20-11-2011, [cit. 2012-02-16]. Dostupné z: <http://www.baltik.wz.cz/baltik.html>

Přílohy

Přílohy a jejich obsah:

Příloha 1: CD-ROM

- Diplomová práce ve formátu PDF
- Seznam základních škol a nižších gymnázií působících na území města České Budějovice.
- Mapa České Republiky s vyznačenými školami, kde se vyučuje algoritmizace nebo programování.

Příloha 2: Seznam základních škol a nižších gymnázií působících na území města České Budějovice.

Příloha 3: Mapa České Republiky s vyznačenými školami, kde se vyučuje algoritmizace nebo programování.