

UNIVERZITA PALACKÉHO V OLOMOUCI

PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

Diplomová práce

Bc. Libor Jurča

Vývoj interaktivní webové aplikace pro podporu výuky
informatických předmětů na 2. stupni ZŠ

Olomouc 2023

Vedoucí práce

Mgr. Tomáš Dragon

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně, uvedl a citoval všechny použité zdroje, které uvádím v jejich seznamu.

V Olomouci dne 16. 4. 2023

Podpis autora práce

Poděkování

Děkuji vedoucímu práce Mgr. Tomáši Dragonovi a Mgr. Lucii Bryndové za vstřícný přístup a odborné vedení mé diplomové práce.

Dále dovoluji, abych zde poděkoval Ing. Dušanovi Fojtů, Ph.D., který mě odborně vedl v mých prvních krocích programování v C# a za jeho odborné rady při vývoji interaktivní aplikace.

Největší část mého poděkování patří mé rodině, a hlavně manželce Oldřišce, která mi byla oporou po celou dobu studia a za její trpělivost, se kterou se starala o rodinnou pohodu.

Obsah

Úvod	7
1. Strategie 2030+	8
1.1. Strategie vzdělávací politiky České republiky do roku 2030	8
1.1.1. Strategický cíl 1: Zaměřit vzdělávání více na získávání kompetencí potřebných pro aktivní občanský, profesní i osobní život	8
1.1.2. Strategický cíl 2: Snížit nerovnosti v přístupu ke kvalitnímu vzdělávání a umožnit maximální rozvoj potenciálu dětí, žáků a studentů.....	9
2. Digitální vzdělávání na základních školách	10
2.1. RVP	10
2.2. RVP ZV	11
2.2.1. Informatika	11
2.2.2. Klíčové kompetence	12
2.2.3. Digitální kompetence.....	12
2.3. ŠVP – Školní vzdělávací program	12
3. Informatika a informatické myšlení	14
3.1. Informatika.....	14
3.2. Informatické myšlení	14
3.2.1. Principy, které lze využít při výuce informatického myšlení.....	15
3.2.2. Informatické myšlení a programování.....	15
3.3. Podpora rozvíjení informatického myšlení (PRIM)	16
3.4. Modelové školní vzdělávací programy pro základní vzdělávání.....	17
3.4.1. Modelový ŠVP: Opatrně vpřed	18
3.4.2. Modelový ŠVP: Nebojácně vpřed	19
3.4.3. Modelový ŠVP: Progresivně vpřed	20
3.4.4. Modelový ŠVP: Kreativně vpřed	22
4. Učebnice	24
4.1. Rozcestník učebnic	24
4.2. Programování ve Scratch pro 2. stupeň základní školy	25
5. Programování.....	26
5.1. Sekvenční programování	26
5.2. Strukturované programování.....	27
5.3. Objektové programování	28
6. Jak učit programování na ZŠ	31

6.1.	Nesdělovat žákům správná řešení	31
6.2.	Neučit definice a pamětní poznatky	31
6.3.	Obrnit se trpělivostí.....	31
6.4.	Očekávat živé a rušné prostředí	31
6.5.	Nechat žáky pracovat ve dvojicích	32
6.6.	Hodnotit praktické dovednosti	32
7.	Webová aplikace.....	33
7.1.	Jak funguje webová aplikace	33
7.1.1.	Zpracování statických stránek	33
7.1.2.	Zpracování dynamických stránek	34
8.	Doména a webhosting	35
8.1.	Doména	35
8.2.	Webhosting	37
9.	Technologie využité při vývoji interaktivní webové aplikace.....	38
9.1.	Inkscape	38
9.2.	HTML	38
9.3.	CSS	41
9.4.	Brackets.....	45
9.5.	Microsoft Visual Studio	47
9.6.	ASP.NET a .NET	48
9.7.	Web Forms	50
9.8.	IIS.....	52
9.9.	C#.....	52
10.	Vývoj webové aplikace	54
10.1.	Volba projektu.....	54
10.2.	Název aplikace	54
10.3.	Psaní kódu v Brackets	55
10.4.	Práce ve Visual studio 2022	60
10.5.	Volba webhostingu a publikování na internet.....	65
10.6.	Popis aplikace.....	66
Závěr.....		71
Seznam použitých zkratk		72
Seznam použitých zdrojů.....		73

Seznam grafů	80
Seznam tabulek.....	81

Úvod

Volbu tématu této diplomové práce ovlivnila celosvětová pandemie koronaviru, způsobená virem SARS-CoV-2, která dostala název COVID-19. V České republice byla zaváděna různá opatření proti šíření nemoci, kdy jedním z opatření, které velmi ovlivnilo život studentů, bylo uzavření všech typů škol. Do této doby jsme se v našem školství nesetkali s tím, že by někdo na základní či střední škole studoval online formou. Přechod na tento typ výuky byl nečekaný a školy i rodiče na toto museli zareagovat. Rodiny, které neměly v domácnosti počítač či chytré zařízení, dostaly prostředky pro online výuku zapůjčeny od školy, případně si rodiče chodili pro studijní materiály do školy. Dovolujeme si říct, že pandemie pomohla učitelům objevit mnoho aplikací, které lze k výuce využít a urychlila jejich rozhodnutí, že tyto moderní technologie do vzdělávání patří.

S rozvojem informačních technologií jde ruku v ruce rozvoj aplikací, které lze na internetu nalézt a využít ve vzdělávání. Výhodou je přístup na mnoho interaktivních stránek, které jsou zdarma, a které si připravují učitelé sami a navzájem je sdílejí. Nevýhodou jsou počáteční náklady na nákup počítačového vybavení, ale i programového vybavení nebo systém, na kterém tyto programy běží.

Hlavním cílem této diplomové práce je vyvinout interaktivní webovou aplikaci pro podporu výuky informatických předmětů na 2. stupni ZŠ a ukázat, že i učitel bez velkých zkušeností s programováním může vyvinout aplikaci pro výuku v předmětech informatiky a dále ji poskytnout k využití ostatním učitelům.

Pro mnoho pedagogů se jeví jako nejlepší aplikace, které běží online, a které lze spustit v internetovém prohlížeči. Výhodou je přístup z jakéhokoli místa, kde je přístup k internetu a není potřeba instalace programu do počítače. Naopak nevýhodou je potřeba připojení k internetu a v případě, že chcete takovou aplikaci vyvinout a mít online, je nutnost si zaplatit webhosting.

V teoretické části představíme pojmy, jako jsou projektové vyučování, výukové metody, digitální gramotnost, informatické myšlení apod.

Praktická část práce bude zaměřena na samotný vývoj webové aplikace, která bude využívaty .NET a jeho webové formuláře.

1. Strategie 2030+

1.1.Strategie vzdělávací politiky České republiky do roku 2030

Tento dokument byl schválen vládou ČR dne 19. 10. 2020 a navazuje na Strategii vzdělávací politiky ČR do roku 2020. Tento dokument má pouze obecnou, zastřešující povahu, ve kterém se popisují priority, které je potřeba v daném desetiletí řešit. Byly stanoveny dva hlavní cíle. Prvním cílem je změna obsahu vzdělávání a druhým cílem je snížení sociálních nerovností ve vzdělávání. Vize této strategie říká *„Implementace Strategie 2030+ povede k vytvoření a rozvoji otevřeného vzdělávacího systému, který reaguje na měnící se vnější prostředí a poskytuje relevantní obsah vzdělávání v celoživotní perspektivě. Cílem vzdělávání v následující dekádě je základními a nepostradatelnými kompetencemi vybavený a motivovaný jedinec, který dokáže v co nejvyšší míře využít svůj potenciál v dynamicky se měnícím světě ve prospěch jak svého vlastního rozvoje, tak s ohledem na druhé a ve prospěch rozvoje celé společnosti“* (Fryč, 2020).

1.1.1. Strategický cíl 1: Zaměřit vzdělávání více na získávání kompetencí potřebných pro aktivní občanský, profesní i osobní život

Nelze opomenout, že v současné době nabraly na rychlosti změny v oblastech technologických, hospodářských, sociálních, kulturních, ale i environmentálních. Z tohoto důvodu se mění požadavky na dovednosti, které jsou pro výkon tradičních i nových povolání potřebné. Mění se i formy komunikace. Typickým znakem pro dnešní generaci je využití digitálních technologií, neomezený přístup k informacím, které je potřeba kriticky hodnotit a dále s nimi pracovat (Fryč, 2020).

Proměny ve vzdělávání

Ve vzdělávání je potřeba upravit vzdělávací obsahy, ale i způsob, kterým se bude učivo předávat. Předání znalostí se bude zaměřovat na méně memorování, a naopak se přejde na jejich pochopení, využití a vzájemné propojení, dále na rozvoj gramotností a zvyšování kompetencí. Učitelé by se měli zaměřit na náročnější úkoly, které budou vyžadovat hlubší porozumění a praktickou aplikaci, a také na dovednosti žáků při vzájemné spolupráci u hledání společných řešení. Žáci se naučí reagovat na výzvy a znalosti a dovednosti, které získají v procesu vzdělávání, využijí i mimo školní prostředí (Fryč, 2020).

Inovace obsahu a způsobu vzdělávání

Inovace obsahu učiva a způsobu vzdělávání je nutné na všech stupních. U dětí a dospělých je cílem dosažení vyšší úrovně klíčových kompetencí a gramotností. Snahou bude podpora při zavádění nových a inovativních metod ve vzdělávání u všech typů škol, ale i u mimoškolních zařízení (Fryč, 2020).

Rozvoj digitálního vzdělávání

V rozvoji digitálního vzdělávání je potřeba vytvořit podmínky nejen pro žáky, ale i pro učitele s cílem zvýšení jejich kompetencí v užívání digitálních technologií, informatického myšlení a digitální gramotnosti. Systém vzdělávání v oblasti digitálních technologií bude obsahovat datovou, informační a mediální gramotnost, dále potom bezpečnost v online prostředí, komunikaci a spolupráci, ale i kritické myšlení a řešení problémů. (Fryč, 2020).

Provzdušnění učiva

Je potřeba zamezit přetěžování žáků informacemi. Učivo, kde je potřeba hlubšího porozumění, aťproberou v širokých souvislostech, ale bez zbytečných poznatků a informací. Učitelé tak získají možnost probrat učivo dostatečně a žáci budou mít prostor informacím porozumět a využít je v reálných situacích (Fryč, 2020).

1.1.2. Strategický cíl 2: Snížit nerovnosti v přístupu ke kvalitnímu vzdělávání a umožnit maximální rozvoj potenciálu dětí, žáků a studentů

Regionální disparity

Napříč celou republikou jsou velké rozdíly ve vzdělávání, které se projevují diferenciací ve výsledcích žáků z různých škol a regionů. Důsledkem toho je vysoký počet žáků, kteří na konci základní školy dosahují velmi nízké úrovně základních znalostí. Velký vliv na nerovnosti v kvalitě vzdělávání má mimo jiné i nedostatek kvalitních pedagogů v různých regionech ČR. (Fryč, 2020).

2. Digitální vzdělávání na základních školách

2.1.RVP

Rámcový vzdělávací program je závazný kurikulární dokument, který dle systému kurikulárních dokumentů patří na státní úroveň. Tento dokument vymezuje výsledky vzdělávání a soubor učiva, které jsou školy povinny zařadit do svých vzdělávacích programů (Zormanová, 2021). Tyto dokumenty byly zavedeny do vzdělávání ČR zákonem č. 561/2004 Sb. o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon), který byl přijatý dne 24. září 2004.

Rámcové vzdělávací programy dle portálu EDU.CZ dělíme na:

- RVP PV – rámcový vzdělávací program pro předškolní vzdělávání
- RVP ZV – rámcový vzdělávací program pro základní vzdělávání
- RVP ZUV – rámcový vzdělávací program pro základní umělecké vzdělávání
- RVP G* – rámcový vzdělávací program pro gymnázia
- RVP SOV – rámcové vzdělávací programy středního odborného vzdělávání
- RVP SPV – rámcový vzdělávací program pro speciální vzdělávání
- RVP ICT – rámcové vzdělávací programy v oblasti informatiky a informačních a komunikačních technologií

Zato rozdělení, které je uvedeno v aktuálním RVP ZV (dále jen RVP ZV) vypadá takto:

- RVP PV – rámcový vzdělávací program pro předškolní vzdělávání
- RVP ZV – rámcový vzdělávací program pro základní vzdělávání
- RVP ZŠS – rámcový vzdělávací program pro základní školy speciální
- RVP G – rámcový vzdělávací program pro gymnázia
- RVP GSP – rámcový vzdělávací program pro gymnázia se sportovní přípravou
- RVP DG – rámcový vzdělávací program pro dvojjazyčná gymnázia
- RVP SOV – rámcové vzdělávací programy středního odborného vzdělávání
- RVP SPV – rámcový vzdělávací program pro speciální vzdělávání
- RVP ZUV – rámcový vzdělávací program pro základní umělecké vzdělávání
- RVP JŠ – rámcové vzdělávací programy pro jazykové školy s právem státní jazykové zkoušky (RVP ZV, 2021).

Tvorbu a oponenturu rámcových vzdělávacích programů provádí příslušná ministerstva, která využívají odborníky z vědy a praxe, včetně psychologie a pedagogiky. Vydání RVP spadá do kompetencí ministerstva školství, mládeže a tělovýchovy (dále jen MŠMT), a to po projednání s příslušnými ministerstvy (EDU.CZ, 2022).

2.2.RVP ZV

V současné době je v platnosti RVP ZV, který byl vydán v lednu 2021, s platností od 1. 9. 2021, s výjimkou kapitoly 11 - Zásady pro zpracování, vyhodnocování a úpravy školního vzdělávacího programu. Tato kapitola nabyla účinnosti k 1. 2. 2022. Z opatření ministra školství, mládeže a tělovýchovy lze vyčíst, že se jedná již o 13. změnu od roku 2004. Hlavním důvodem změny RVP ZV je to, aby školy měly dostatek času se připravit na novou podobu výuky informatiky (MŠMT, s. 1).

Revize RVP ZV vychází ze strategie 2030+ a zachovává základní strukturu současného RVP ZV. Tato revize byla schválena v roce 2021 a bude postupně zaváděna do vzdělávání, a to nejpozději pro první stupeň základní školy od 1. 9. 2023 a pro druhý stupeň základní školy od 1. 9. 2024. V revizi RVP ZV dochází k začlenění nové vzdělávací oblasti – Informatika (RVP ZV, 2021).

2.2.1. Informatika

Tato oblast se zaměřuje na rozvoj inforatického myšlení a na porozumění základním principům digitálních technologií. Žáci by se měli do výuky začlenit více aktivně a využívat inforatické postupy a pojmy. Žáci by měli porozumět tomu, jak digitální technologie fungují, a tím získat povědomí o tom, jak je efektivně, bezpečně a eticky využít.

Žáci prvního stupně vzdělávání mají zvýšenou časovou dotaci v plánu z jedné hodiny na dvě hodiny, kdy tento čas využijí prostřednictvím her, experimentů, diskuzí a dalších aktivit k utvoření představy, jak data a informace zaznamenat a objevovat počítačové aspekty světa kolem nich. Žáci budou postupně rozvíjet schopnost popsat problém, analyzovat jej a najít řešení tohoto problému, ale současně si osvojí dovednosti a návyky pro bezpečné zacházení s technologiemi, vedoucí k prevenci rizikového chování.

V druhém stupni by žáci měli tvořit, experimentovat, ověřovat si své hypotézy, navrhnout a ověřit si různá řešení, která mohou prodiskutovat s ostatními, a tím se zdokonalit v porozumění principů fungování digitálních technologií. Při analýze problému vyberou

důležité aspekty, které potřebují pro řešení tohoto problému. Po poznání principu funkce digitálních technologií žáci chápou základy kódování, modelování a dokážou více chránit své soukromí, data i zařízení (RVP ZV, 2021).

2.2.2. Klíčové kompetence

Klíčová kompetence – v RVP ZV je definována takto: „*Klíčové kompetence představují souhrn vědomostí, dovedností, schopností, postojů a hodnot důležitých pro osobní rozvoj a uplatnění každého člena společnosti*“ (RVP ZV, 2021, s. 10).

Cílem a smyslem vzdělávání je vybavit všechny studenty souborem klíčových dovedností na dané úrovni, která je jim přístupná a připravuje je k dalšímu studiu a uplatnění ve společnosti. Osvojit si klíčové kompetence je dlouhý a složitý proces, který začíná již v předškolním vzdělávání, dále pokračuje přes základní a střední školu a postupně se utváří v průběhu života.

Klíčové dovednosti neexistují samostatně, různě se prolínají, mají mezipředmětovou podobu a lze je vždy získat pouze jako výsledek celkového procesu vzdělávání.

Do základního vzdělávání byly zařazeny tyto klíčové kompetence: kompetence k učení, kompetence k řešení problému, kompetence komunikativní, kompetence sociální a personální, kompetence občanské, kompetence pracovní a v novelizaci přibyla kompetence digitální (RVP ZV, 2021).

2.2.3. Digitální kompetence

Žák, který ovládá tuto kompetenci, je schopen běžně používat zařízení, aplikace a služby, dokáže samostatně rozhodnout, které technologie a pro jakou činnost nebo řešený problém využít. Žák je schopen získat, vyhledat, kriticky posoudit, spravovat a sdílet data, informace a digitální obsah, k tomu dokáže zvolit postupy pro konkrétní situaci a účel. K usnadnění práce využívá digitální technologie, chápe jejich význam pro společnost, minimalizuje situace ohrožující bezpečnost dat a zařízení (RVP ZV, 2021).

2.3. ŠVP – Školní vzdělávací program

Tento dokument je oproti RVP na školní úrovni a vychází z RVP. To znamená, že za každý tento dokument je zodpovědná každá škola sama. Podle ŠVP se uskutečňuje výuka a je specifický pro každou školu. Jsou v něm zohledněny potřeby a možnosti žáků, ale také možnosti školy, založené na jejich představách a zkušenostech z praxe. Tento program vytváří

pedagogové na škole a již nepodléhá schvalování ze strany MŠMT, a podle něj učitelé na svých školách vyučují. Nejsou tedy již vázáni na osnovy, kterých by se museli držet. ŠVP popisuje dovednosti, jež mají žáci mít. Proto mohou některé pasáže látky vypustit, či zredukovat nebo naopak některé projekty prodloužit, a tím splnit základní cíl výuky (Zormanová, 2014).

Za tvorbu ŠVP na jednotlivých školách odpovídá ředitel školy, který může určit koordinátorem svého zástupce nebo jiného člena pedagogického sboru. Návrh ŠVP předkládá a vyjadřuje se k němu školská rada. ŠVP je součástí povinné dokumentace a musí být zpřístupněn veřejnosti, aby se případný zájemce do něj mohl kdykoliv podívat, pořídit si výpisky či kopie (RVP ZV, 2021).

Pro tvorbu jednotlivých ŠVP je pro školy připraven manuál, který pochází z MŠMT. Obsahuje kompletní návod, jak by měl takový program vypadat, co by měl obsahovat také a postup tvorby ŠVP (Zormanová, 2014).

3. Informatika a informatické myšlení

3.1. Informatika

Pojem informatika ve své knize Gála, Pour a Šedivá charakterizuje jako vědu, která se zabývá vyjádřením, zpracováním a přenášením informací v určitém systému (Gála, Pour, Šedivá, 2015), zato Bělohávek definoval pojem informatika takto: „*Informatika se zabývá studiem procesů zpracovávajících informace, jejich teoretickými základy, analýzou, návrhem, efektivitou, implementací a aplikacemi, ať už jde o informace uložené ve formě bitů v paměti počítače, nacházející se v dokumentech na internetu nebo zapsané v genech živých organismů. Základní otázkou, která se promítá do všech oblastí informatiky, je: Co vše lze efektivně mechanicky spočítat?*“ (Bělohávek, 2016, s. 301).

3.2. Informatické myšlení

Daniel Lessner ve svém článku přiděluje prvotní zmínku o informatickém myšlení (computational thinking) Seymouru Papertovi, který je autorem programovacího jazyka LOGO. Tento jazyk slouží ke konstruktivistickému vzdělávání. Zmínka o informatickém myšlení napsal ve svém článku, pojednávajícím o možnosti využití počítačů při výuce matematiky, který vyšel v roce 1996 v časopise International Journal of Computers for Mathematical Learning. Významně se o rozvoj pojmu informatické myšlení zasloužila Jeanette Wing v roce 2006, článkem o informatickém myšlení, který vyšel v časopise Communications of the ACM. Wing přiřazuje informační myšlení ke schopnostem základním, jako je čtení, psaní a počítání, a zároveň zde pojmenovala důležitou stránku výuky. Nepoužívat technologie a dovednosti IT profesionálů, ale použít jejich způsob myšlení, který díky informatice můžeme získat (Lessner, 2014). Wing v roce 2010 v dalším článku cituje informatické myšlení takto: „*Informatické myšlení jsou myšlenkové postupy zapojené při takovém formulování problémů a jejich řešení, které umožní tato řešení efektivně provést agentem zpracovávajícím informace.*“ (Wing, 2010, s. 1). Autorka agentem myslela, jak stroj, tak i člověka.

Jak lze definovat informatické myšlení? Informatické myšlení může být postup řešení problému. Tento způsob myšlení je zaměřen na popis problému, analýzu a efektivní řešení problému. Pokud se jej naučíme používat, lze jej využít např. při plánování a řízení činností, rozdělit si velký problém na více malých, snáze řešitelných, vybrat podstatné aspekty problému, které lze vynechat a v neposlední řadě použít programovací jazyky, díky kterým se domluvíme s počítači a roboty.

Toto myšlení je potřeba se naučit nejprve na jednoduchých a postupně složitějších úkolech, či problémech. Informatické myšlení lze efektivně využít u komplexního problému, kde sice na začátku ztratíme trochu času při analýze, či popisu problému, ale naopak čas získáme při jeho řešení. Při práci lze navíc využít počítačů nebo dalších strojů řízených počítači (imysleni.cz, 2021).

3.2.1. Principy, které lze využít při výuce informatického myšlení

Pokus – omyl je cesta vpřed

Žáci se nesnaží postupovat dle známých postupů, ale jsou vyzýváni k tomu, aby se snažili nalézt nová, vlastní řešení. Chyba, kterou žáci udělají, je součástí procesu učení a žákům ukáží, že tato cesta není správná. Poučí se ze svých chyb a pokračují dalším pokusem. Zde se nejvíce osvědčí testování, které nám poskytuje zpětnou vazbu (imysleni.cz, 2021).

Učíme se tím, že to děláme

Cílem učitele je předat žákům jistotu, aby začali důvěřovat svému úsudku a svým schopnostem. Zde je jádrem vlastní aktivní práce žáků, kdy se snaží přemýšlet a kombinovat postupy, které již znají s těmi nově vymyšlenými, a tím je jim umožněno vyřešit nové a doposud neřešené postupy (imysleni.cz, 2021).

Podstatná je vytrvalost

Žák musí pochopit, že má smysl na něčem pracovat dlouhodoběji, třeba i několik dní a najít zábavu v soustředěné práci, a nejen rychle střídat různé druhy prací. Nevzdat se, pokud to nepůjde hned, ale naopak se snažit přidat ve své práci (imysleni.cz, 2021).

Spolupracujeme

K řešení problému se postaví týmy, které budou spolupracovat, využívat silných stránek různých lidí a budou spolu v týmu komunikovat nad řešením daného problému. Každý v týmu může konkretizovat svoji myšlenku a správně formulovat. Díky tomu porozumí problému a komunikaci v praxi (imysleni.cz, 2021).

3.2.2. Informatické myšlení a programování

Představa o programování je různorodá, ale obvykle si člověk představí programování jako soubor činností, které souvisí s tvorbou aplikací nebo také zapisování zdrojového kódu.

Na základní škole se při výuce informatiky žáci seznámí se základy programování, aby získali všeobecný rozhled.

Jednou z výukových metod bude programování. Žáci zjistí, že strojům lze zadat i poměrně složité úkoly, ale musí své řešení přesně zformulovat a zapsat. Zápis je pro jejich začátky udělán jednoduše, kde pomocí předpřipravených bloků sestavují daný kód. Tím odpadá starost s přesností zápisu kódu, a tím se můžou žáci soustředit na postupu řešení (imysleni.cz, 2021).

3.3.Podpora rozvíjení informatického myšlení (PRIM)

Národní ústav pro vzdělávání (NÚV) charakterizuje tento projekt takto: „*Projekt PRIM si klade za úkol inovovat obsah vzdělávací oblasti Informatika a ICT akcentováním výuky zaměřené na rozvoj informatického myšlení žáků*“ (Národní ústav pro vzdělávání, 2022).

Hlavním úkolem projektu je předpoklad vytvoření ucelených sad výukových materiálů pro všechny stupně škol, ale i systém vzdělávání učitelů, kteří se věnují výuce informatiky. Současně se snaží zpopularizovat témata týkající se informatického myšlení, zejména programování, porozumění informacím a robotiku (Národní ústav pro vzdělávání, 2022).

Mezi hlavní důvody realizace projektu je narůstající potřeba IT profesí a rozpor mezi potřebou a dosavadní orientací našeho školství, které nereflektuje na změnu potřeb na trhu práce. Dalším důvodem je zavádění informačního obsahu do školního kurikula ve vyspělých zemích. Jádrem projektu jsou pedagogické fakulty, které se podílejí na doktorském studijním programu Informační a komunikační technologie ve vzdělávání, ale také pracoviště, jež se dlouhodobě zabývají výzkumem v oblasti didaktiky informatiky a pracoviště, která se podílí na tvorbě učebnic informatiky (KVD, 2022).

Projekt byl spolufinancován Evropskou unií a probíhal od 1. 10. 2017 do 30. 11. 2020 a lze jej dohledat pod registračním číslem projektu CZ.02.3.68/0.0/0.0/16_036/0005322 (imysleni).

Hlavním příjemcem projektu je Jihočeská univerzita v Českých Budějovicích a k projektu je připojeno mnoho partnerů. Mezi tyto partnery spadá Masarykova univerzita, Ostravská univerzita, Univerzita Jana Evangelisty Purkyně v Ústí nad Labem, Univerzita Karlova, Univerzita Palackého v Olomouci, Technická univerzita v Liberci, Univerzita Hradec Králové, Západočeská univerzita v Plzni a Národní ústav pro vzdělávání, školské poradenské zařízení pro další vzdělávání pedagogických pracovníků (imysleni.cz, 2021).

3.4. Modelové školní vzdělávací programy pro základní vzdělávání

Jak již bylo zmíněno výše, v projektu PRIM byla jedním z hlavních úkolů tvorba ucelených sad výukových materiálů pro všechny druhy škol a tyto materiály jsou volně dostupné na stránkách imysleni.cz v sekci ŠVP, a dle zaměření je na výběr ŠVP pro základní vzdělávání nebo pro gymnázia. Jelikož se naše práce zaměřuje na tvorbu webové aplikace pro druhý stupeň základní školy, budeme se zmiňovat pouze modelovými ŠVP základní vzdělávání. Tyto ŠVP nejsou určeny pro přechodné období, ale ukazují, jak by mohly ŠVP vypadat po roce 2024. Pro přechodné období lze své ŠVP upravit vybráním témat a nahrazením za své stávající. Tím lze docílit postupného zavádění témat do výuky. V přechodném období lze témata zařazovat i do jiných ročníků, než jsou primárně určeny, jelikož se informatika nebude v přechodném období učit ve všech ročnících a v důsledku toho může učitel použít stejné téma v různých ročnících (imysleni.cz, 2021).

Autorství a proces tvorby

Modelové školní vzdělávací programy vznikaly v průběhu roku 2020 a podíleli se na nich autoři jednotlivých učebnic pod vedením a finální autorizací Jiřího Vaníčka a Jana Berkiho. Jejich znění bylo projednáno a připomínkováno odborným panelem pro inovaci kurikula při národním pedagogickém institutu (imysleni, 2021).

Na těchto stránkách jsou k dispozici modelové ŠVP pro schválenou hodinovou dotaci 2 + 4 a ve čtyřech verzích. První až třetí varianta má sestaven učební plán jako časový a je uspořádán po ročnících a tématech. Tím je dosažena návaznost témat, pokud dodrží navržené pořadí.

V tabulkách učebních plánů jsou témata, která je nutné zařadit pro naplnění všech výstupů v RVP, označena písmenem A. Pokud by se škola rozhodla toto téma nezařadit, měla by očekávané výstupy zabezpečit jiným způsobem např. výukou v jiném programovacím prostředí.

Očekávané výstupy RVP jsou u témat konkretizovány a jsou rozepsány do očekávaných výstupů ŠVP a tyto výstupy jsou formulovány a terminologií přizpůsobeny vzdělávacímu materiálu, který je v tématech uveden jako doporučený.

V jednotlivých variantách ŠVP se konkrétní témata neliší svými výstupy, ale rozdíl může být v dotaci hodin. Některá témata umožňují se zabývat problematikou více do hloubky nebo zvolit i časově náročnější metody výuky.

U některých témat je položka zdroje prázdná, protože se většinou jedná o tradiční témata, která se již v informatice vyučují (imysleni, Modelový ŠVP – Opatrně vpřed, 2021).

3.4.1. Modelový ŠVP: Opatrně vpřed

Tento modelový ŠVP je určen školám, které se rozhodnou přejít na výuku informatiky bez pořízení pomůcek a učebnic, ale chtějí digitální gramotnost rozvíjet v ostatních předmětech. V informatice se zaměří na rozvoj informatického myšlení a informatických témat. Jediným požadavkem je, aby byla škola vybavena standartní počítačovou učebnou, kterou měla již před přechodem na nový ŠVP. V tabulce č. 1 je ukázka modelového učebního plánu.

Tab. 1: Učební plán – Opatrně vpřed

ročník	téma	hodiny	je třeba počítač
4.	Ovládání digitálního zařízení	12	A
	Práce ve sdíleném prostředí	11	A
	Úvod do kódování a šifrování dat a informací	10	
5.	Úvod do práce s daty	4	A
	Základy programování – příkazy, opakující se vzory	6	A
	Úvod do informačních systémů	3	
	Základy programování – vlastní bloky, náhoda	7	A
	Úvod do modelování pomocí grafů a schémat	7	
	Základy programování – postavy a události	6	A
6.	Kódování a šifrování dat a informací	9	
	Práce s daty	10	A
	Informační systémy	5	
	Počítače	9	A
7.	Programování – opakování a vlastní bloky	14	A
	Modelování pomocí grafů a schémat	7	
	Programování – podmínky, postavy a události	12	A
8.	Programování – větvení, parametry a proměnné	15	A
	Hromadné zpracování dat	18	A
9.	Programovací projekty	12	A
	Digitální technologie	15	
	Závěrečné projekty	6	A

(Zdroj: Modelový ŠVP: Opatrně vpřed)

Z tabulky lze vyčíst, že výuka začíná ve čtvrtém ročníku, dále jednotlivé hodinové dotace v daných ročnících. Dále lze vidět v jakých tématech je potřeba pro výuku využít počítače, a u kterých naopak počítače potřeba nejsou. Při podrobnějším zkoumání modelového příkladu naleznete, které zdroje jsou k dispozici pro využití ve výuce. U daných témat je uveden seznam

zdrojů, které lze využít pro výuku s odkazem na jejich stažení. Více se tímto tématem budeme zabývat v další kapitole nazvané Učebnice (imysleni, Modelový ŠVP – Opatrně vpřed, 2021).

3.4.2. Modelový ŠVP: Nebojácně vpřed

V tomto modelovém ŠVP se škola zaměřuje na informatiku a technické směřování rozvoje žáků. Do výuky jsou přidány základy robotiky jako aplikovaná oblast, jsou zde propojeny informatika a programování s technikou, tím je jim umožněno řešit praktické problémy. Informatické myšlení je rozvíjeno projektovou činností a podporou tvořivosti. Opět zde žáci používají počítačovou učebnu, v některých případech lze učit bez použití počítače. Rozdíl oproti předchozímu modelovému ŠVP je v použití pomůcek, které musí škola pořídit, jedná se zejména o robotické stavebnice a programovatelné pomůcky.

U některých témat je žádoucí práce ve dvojicích u jednoho počítače, aby zde docházelo k diskuzi a spolupráci. Žáci pracují individuálním tempem. Výuka je zde orientována činnostně, kde se vyžaduje aktivita žáků, který objevuje, experimentuje, snaží se ověřit své hypotézy, řešit problémy a pracovat s využitím projektů. Není zde důraz na pamětné učení a reprodukci. V tabulce č. 2 lze vidět návrh učebního plánu (imysleni, Modelový ŠVP – Nebojácně vpřed, 2020).

Tab. 2: Učební plán – Nebojácně vpřed

roč.	téma	hodiny	nutné k naplnění RVP	je třeba počítač	nutný nákup pomůcek
4.	Ovládání digitálního zařízení	10	A	A	A
	Práce ve sdíleném prostředí	6	A	A	
	Základy robotiky se stavebnicí	8		A	
	Úvod do kódování a šifrování dat a informací	9	A		
5.	Úvod do práce s daty	4	A	A	
	Základy programování – příkazy, opakující se vzory	6	A	A	
	Úvod do informačních systémů	3	A		
	Základy programování – vlastní bloky, náhoda	7	A	A	
	Úvod do modelování pomocí grafů a schémat	7	A		
	Základy programování – postavy a události	6	A	A	
6.	Kódování a šifrování dat a informací	9	A	A	
	Práce s daty	10	A		
	Informační systémy	3	A		

roč.	téma	hodiny	nutné k naplnění RVP	je třeba počítač	nutný nákup pomůcek
	Programování – opakování a vlastní bloky	11	A	A	
7.	Programování – podmínky, postavy a události	9	A	A	
	Modelování pomocí grafů a schémat	6	A		
	Programování – větvení, parametry a proměnné	13	A	A	
	Počítače	5	A	A	
8.	Programování robotické stavebnice (alt. Programování hardwarové desky)	20 (20)		A A	A A
	Hromadné zpracování dat	13	A	A	
9.	Programovací projekty	12	A	A	
	Digitální technologie	15	A		
	Závěrečné projekty	6		A	

(Zdroj: Modelový ŠVP – Nebojácně vpřed)

V tomto modelovém ŠVP, jak již bylo zmíněno, se objevují témata základy robotiky ve čtvrtém ročníku a postupně budou témata nabalovány až do devátého ročníku. Pro výuku základy robotiky jsou doporučovány robotické stavebnice LEGO WeDo 2.0, LEGO Mindstroms EV3, alternativně lze využít programovatelnou desku Micro:bit. (1 stavebnice pro dva žáky). I pro tento modelový ŠVP jsou připraveny učebnice, které mohou učitelé využít pro výuku (imysleni, Modelový ŠVP – Nebojácně vpřed, 2020).

3.4.3. Modelový ŠVP: Progresivně vpřed

Tento modelový ŠVP má jako jediný v návrhu zvýšenou hodinovou dotaci, a to v podobě 3 + 5 hodin (3 hodiny na prvním stupni a 5 hodin na stupni druhém). Jelikož je zde zvýšena hodinová dotace, tak výuka informatiky začíná již ve třetím ročníku. Tyto hodiny jsou čerpány z disponibilních hodin. Výuka informatiky na druhém stupni má 2 hodiny týdně a je jen na rozhodnutí školy, zda ji realizuje ve druhém pololetí 8. ročníku nebo v prvním pololetí 9. ročníku. Do výuky je třeba pořídit robotické stavebnice a edukační software. V tabulce č. 3 lze vidět učební plán – progresivně vpřed (imysleni, Modelový ŠVP – Progresivně vpřed, 2020).

Tab. 3: Učební plán – Progresivně vpřed

roč	téma	hodiny	nutné k naplnění RVP	je třeba počítač	nutný nákup pomůcek
3.	Základy algoritmizace s robotickou hračkou Základy algoritmizace Ovládání digitálního zařízení	5 12 16	A	A A	A A
4.	Práce ve sdíleném prostředí Základy robotiky se stavebnicí Úvod do kódování a šifrování dat a informací Řešení problémů pomocí algoritmizace	6 10 10 7	A A	A A A	A A
5.	Úvod do práce s daty Základy programování – příkazy, opakující se vzory Úvod do informačních systémů Základy programování – vlastní bloky, náhoda Úvod do modelování pomocí grafů a schémat Základy programování – postavy a události	4 6 3 7 7 6	A A A A A A	A A A A A	
6.	Kódování a šifrování dat a informací Práce s daty Informační systémy Programování – opakování a vlastní bloky	9 10 3 11	A A A A	A A A	
7.	Programování – podmínky, postavy a události Modelování pomocí grafů a schémat Programování – větvení, parametry a proměnné Počítače	9 6 13 5	A A A A	A A A A	
8.	Programování robotické stavebnice Hromadné zpracování dat	20 13	A	A A	A
8./9.	Programování hardwarové desky Programovací projekty	20 13	A	A A	A
9.	Programovací projekty Digitální technologie Závěrečné projekty	5 15 13	A A	A A	

(Zdroj: Učební plán – Progressivně vpřed)

Organizační a obsahové vymezení vyučovacího předmětu vychází z modelu ŠVP – Nebojácně vpřed, ale jak již bylo zmíněno výše, je zde navýšena hodinová dotace, a tím pádem je potřeba nakoupit více výukových pomůcek jako jsou: robotická hračka Bee-bot nebo Blue-bot a deska Tac-Tile (po 6 kusech do třídy), LEGO WeDo, LEGO Mindstorms EV3(1 deska na 2 žáky), edukační software pro algoritmizaci Emil 3 a Emil 4 (licence pro třídu + žákovské

sešity) a programovatelná deska Micro:bit (imysleni, Modelový ŠVP – Progresivně vpřed, 2020).

3.4.4. Modelový ŠVP: Kreativně vpřed

Zde není stanoven klasický vzdělávací program, ale obsahuje přehled nabízených tematických celků, který tvoří náplň předmětů. Každý tematický celek uvádí, pro který stupeň je určen, a zda naplňuje požadavky na očekávané výstupy RVP v daném okruhu témat. Počet hodin není pevně stanoven, ale ukazuje předpokládanou dobu k naplnění očekávaných výstupů. Škola si sestavuje svůj učební plán sama z nabídky a lze k nim doplnit vlastní tematické celky. Tabulka č. 4 uvádí seznam témat pro první stupeň a tabulka č. 5 uvádí seznam pro druhý stupeň (imysleni, Modelový ŠVP – Kreativně vpřed, 2020).

1. stupeň

Tab. 4: Učební plán – Kreativně vpřed, 1. stupeň

Téma	hodiny	nutné k naplnění RVP	je třeba počítač	nutný nákup pomůcek
Základy algoritmizace s robotickou hračkou	5			A
Základy algoritmizace (verze Tomáš)	5		A	
Základy algoritmizace (verze Emil)	20		A	A
Základy programování	22	A	A	
Základy robotiky se stavebnicí	8		A	A
Úvod do kódování a šifrování dat a informací	9	A		
Úvod do modelování pomocí grafů a schémat	8	A		
Úvod do informačních systémů	3	A		
Úvod do práce s daty	5	A	A	
Ovládání digitálního zařízení	16	A	A	
Práce ve sdíleném prostředí	6	A	A	

(Zdroj: Učební plán – Progresivně vpřed)

2. stupeň

Tab. 5: Učební plán – Kreativně vpřed, 2. stupeň

Téma	hodiny	nutné k naplnění RVP	je třeba počítač	nutný nákup pomůcek
Programování	33	A	A	
Programování robotické stavebnice	22		A	A
Programování hardwarové desky	15		A	A
Programovací projekty	12	A	A	
Kódování a šifrování dat a informací	9	A		
Modelování pomocí grafů a schémat	6	A		
Informační systémy	3	A		
Práce s daty	10	A	A	
Hromadné zpracování dat	13	A	A	
Digitální technologie	20	A	A	

(Zdroj: Učební plán – Progresivně vpřed)

4. Učebnice

Učebnice a vzdělávací materiály vznikly v projektu PRIM, na kterých se podíleli učitelé z vysokých, středních a základních škol. Ověření učebnic zabralo dva a půl roku a probíhalo ve třech fázích na desítkách škol. Další úpravy probíhaly na základě poznatků učitelů a autoři učebnic provedli potřebnou úpravu.

Učebnice ukazují, jak lze rozvíjet u žáků informatické myšlení a jak se zaměřit na určitou oblast informatiky. Učitel má možnost se rozhodnout, zda bude realizovat celou výuku podle těchto materiálů nebo si může zvolit pouze určitou oblast a svoji výuku jimi doplnit, a tím naplnit obsah výuky, aby vyhovovala novým RVP.

Učebnice jsou zcela zdarma ke stažení nebo je lze spustit online.

Ke každé učebnici jsou k dispozici žákovské listy, metodické listy pro učitele nebo další soubory, které budou učitelům nápomocny k výuce, jako jsou pracovní soubory či návody ke konstrukci apod.

V metodických listech pro učitele naleznete pasáž učit na co si dát pozor, cíle a řešení úloh. Oproti tomu v žákovských materiálech naleznete zadání úkolů nebo projektů či pracovní soubory, se kterými mohou žáci pracovat.

Učebnice jsou k dispozici pouze elektronicky z důvodu jejich aktualizací a vylepšení. Učebnice žáci do ruky nepotřebují, protože žákovské listy jim učitel může promítnout nebo si je mohou otevřít na svém počítači (imysleni, 2021).

4.1. Rozcestník učebnic

Na obrázku 1 je znázorněn rozcestník učebnic, kde jednotlivé názvy je možné rozkliknout a fungují jak hypertextový odkaz na dané materiály, které lze využít pro výuku v předmětech. Učebnice jsou volně ke stažení a k využití pro učitele, kteří budou přecházet na novou výuku. Jak můžete na obrázku 1 vidět, k dispozici jsou učebnice pro první, druhý stupeň a střední školu.

Použité odstíny barev mají svůj význam a snaží se učiteli napovědět, do kterého ročníku je primárně učebnice určena. Tmavá barva znázorňuje ročník, ve kterém by měl učitel učebnici použít. Světlejší barvu znázorňují ročníky, ve kterých lze ještě učebnici využít, ale zde je na učiteli, aby provedl výběr nebo vlastní úpravy (imysleni, 2023)

	MŠ	ZŠ / 1. stupeň					ZŠ / 2. stupeň				SŠ			
		1	2	3	4	5	6	7	8	9	1	2	3	4
Programování a algoritmizace	Tomáš													
	Robotické hračky Bee-bot													
	Scratch 1. st.													
	Scratch 2. st.													
	Scratch 2. st. (pokročil)													
Informatika (ostatní témata)	Python													
	Základy informatiky 1. st.													
	Základy informatiky 2. st.													
	Práce s daty													
Základy robotiky	Základy informatiky SŠ													
	LEGO WeDo													
	LEGO Mindstorms													
	Micro:bit s Makecode													
	Micro:bit s Pythonem													
Arduino														

Obr. 1 Rozcestník učebnic

(Zdroj: imysleni.cz/ucebnice)

Pro naši diplomovou práci jsme využili zejména knihu zaměřenou na programování a algoritmizaci a určenou pro 2. stupeň základní školy.

4.2. Programování ve Scratch pro 2. stupeň základní školy

Učebnice se skládá z materiálů pro žáka a z materiálů pro učitele. Obsahuje žakovské listy (zadání úloh k promítnutí na projekci), pracovní soubory s projekty (projekty si žáci pouští na počítači a pracují s nimi) a metodickou příručku pro učitele. Učebnice je v rozsahu 32 vyučovacích hodin a je připravena pro žáky 7. ročníku ZŠ, ale je možné ji využít i v sousedních ročnících, popřípadě je možnost si látku rozdělit do dvou ročníků. Výuka je rozdělena do deseti tematických celků. Každá kapitola má základní i rozšiřující učivo. Doporučení autorů v případě dvouhodinovky je střídat programování s jinou činností z důvodu náročnosti činnosti a udržení pozornosti dětí (Vaníček a kol., 2020).

5. Programování

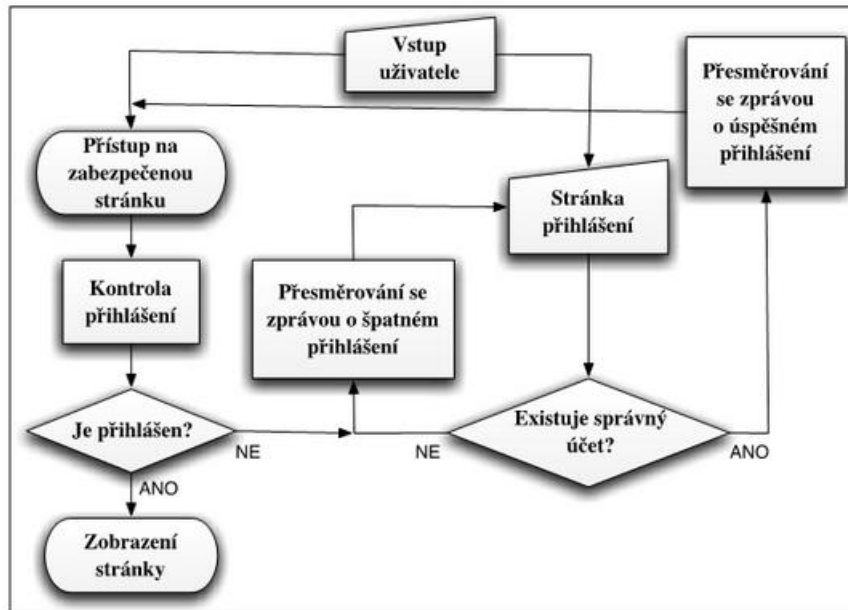
O programování můžeme hovořit jako o procesu, který zahrnuje činnosti od návrhu algoritmu, přes psaní, testování a ladění počítačového programu. Program jako takový představuje návod k vyřešení nějakého programu. Počítač sám neumí najít způsob, jak problém vyřešit, to musí udělat programátor. Pokud tedy znáte cestu, jak najít řešení problému, můžete mu tu cestu ukázat pomocí programu, musíte mu napsat program. Počítač bude umět jen to, co ho naučíme, tzn. na co má program (Virus, 2021).

Při programování se můžeme setkat s pojmem algoritmus. Zjednodušeně můžeme říct, že se jedná o návod, který chceme předložit počítači, a který musí mít určité vlastnosti. Základní podmínkou je, aby návod vedl k požadovanému výsledku. Do kýmženeho cíle vede mnoho cest, ale je potřeba najít takovou, která povede ke správnému výsledku a zda splní předpoklady pro data, se kterými bude návod pracovat. Návod by se měl skládat z kroků, kterým bude počítač rozumět, těmto krokům říkáme elementární kroky. Počet kroků by neměl být zbytečně veliký. Není problém napsat krátký program, který nikdy neskončí, jelikož v něm lze opakovat donekonečna určitá skupina instrukcí (Virus, 2021).

Při psaní algoritmu lze využít dvě metody. A to metodu shora dolů nebo metodu zdola nahoru. U metody shora dolů postupujeme tak, že rozložíme malé úseky na menší, až dospějeme k elementárním krokům. Při dělení se nám může stát, že některé rozdělení bude potřeba upravit, některé kroky spojit a některé naopak přidat. Oproti tomu metoda zdola nahoru znamená naučit počítač nové elementární kroky, které poskládáme z kroků, které již počítač zná. Pokud provádíme často nějakou operaci, můžeme si naprogramovat algoritmus, dáme mu jméno a později ho budeme pouze využívat (Virus, 2021).

5.1. Sekvenční programování

V začátcích programování psal programátor sekvenci příkazů, které počítač prováděl jeden za druhým, proto se mu říkalo sekvenční programování. Průběh sekvence šel samozřejmě narušit a předat řízení jinam, ale bylo to nepohodlné a pro další druhy úprav programu nešťastné řešení. Při tomto druhu programování se používali jednoduché podmínky a skoky na jiné místo ve zmíněné sekvenci příkazů. Programátoři rádi k tomuto využívali vývojový diagram, který můžete vidět na obr. č. 2.

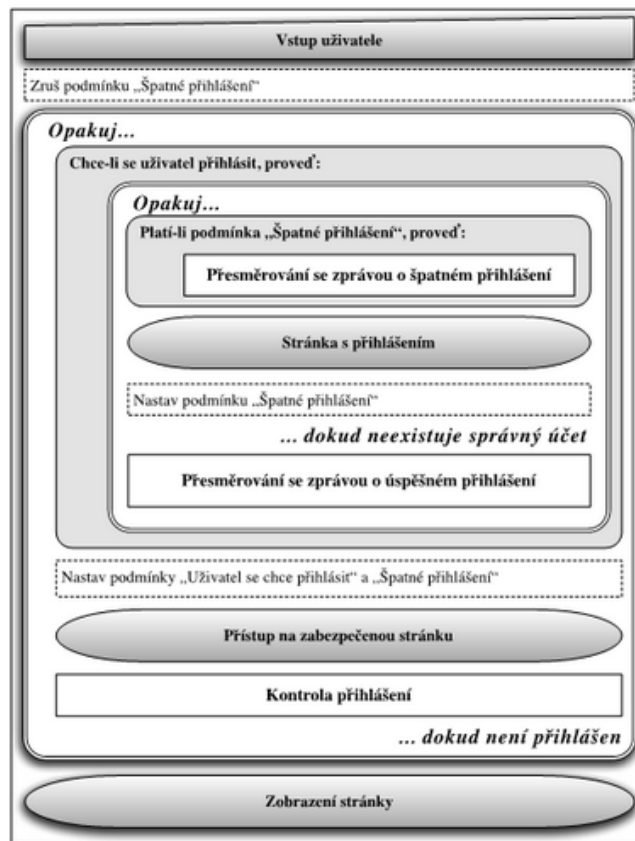


Obr. 2 Vývojový diagram přihlášení k webové aplikaci

(Zdroj: Čada, 2009 Objektové programování – naučte se pravidla objektového myšlení, str. 17)

5.2. Strukturované programování

Oproti tomu strukturované programování využívá striktně hierarchickou strukturu stromového typu. Zde přestalo existovat řízení stylem „jdi na“, ale namísto toho začali skládat hotové stavební bloky, které využívali podmínky. Jako příklad můžeme uvést opět přihlášení. Pokud platí podmínka P, proved' blok A, pokud neplatí, tak proved' blok B nebo prováděj blok A dokud platí podmínka P. Tento příklad můžeme vidět graficky na obrázku č. 3.



Obr. 3 Strukturovaný diagram

(Zdroj: Čada, 2009 Objektové programování – naučte se pravidla objektového myšlení, str. 18)

Velkým rozdílem oproti sekvenčnímu programování je v mnohem jednodušší úpravě strukturovaného kódu a minimální riziko zanesení nových chyb a problémů. Jednotlivé bloky jsou jasně definované, a pokud dodržíme tato pravidla, lze uvnitř bloku provádět úpravy, změny i zdokonalení. Na stejném způsobu programování funguje mnoho programovacích jazyků dodnes. Nevýhodou tohoto programování je jeho rigidní hierarchie (Čada, 2009).

5.3. Objektové programování

Někde se lze setkat s výrazem objektově orientované programování, které v současné době představuje nejvíce rozšířenou metodu vytváření programů. Tato metoda napodobuje vzhled a chování objektů z reálného světa s velkou možností abstrakce.

Objekty – základními prvky jsou útvary, jež nazýváme objekty. Mohou obsahovat útvary z reálného světa, jako jsou osoby, předměty, dokumenty, události. Každý z těchto objektů obsahuje informaci o svém stavu a má schopnost své stavy na požádání měnit. Objekty obsahují jak data, tak procedury, které s těmito daty pracují. Průběh výpočtu je určen posíláním zpráv

mezi jednotlivými objekty. Pomocí zpráv můžete změnit stav objektů a na zaslanou zprávu bude objekt nějak reagovat. Tuto reakci definujeme v kódu, který se označuje jako metoda. Proto se výměně zpráv mezi objekty říká, že volá jeho metodu. Reakce na metodu záleží, kým byla zpráva odeslána a na tom v jakém stavu se objekt v době obdržení kódu nacházel (Perzina, 2021).

Třídy – ve světě se objekty často opakují, proto vytváříme tzv. třídy, které jsou vlastně šablonami, jež definují, jak se budou objekty vytvářet, které značíme instance dané třídy. Šablona tedy definuje, jaké budou mít instance atributy a jaké budou mít metody. Třídy mohou být objektem a v praxi se tohoto využívá. Pomocí tohoto dochází k zjednodušení systému, ve kterém je zapotřebí pracovat s prvky dvou odlišných typů, jak s objekty, tak i třídami, v tomto případě pracujeme pouze s objekty, z nichž jsou některé třídami (Čada, 2009).

Instance – jedná se o objekty, které nejsou třídou, ale mohou být od třídy odvozeny. Mohou být složeny z dalších instancí tříd. Všechny instance dané třídy mají stejné atributy, ale mohou mít rozdílné hodnoty (Čada, 2009).

Dědění – mezi základní zásady při programování patří zásada neopakování se. Při programování kódy, které se opakují stále dokola, neopakujeme, ale využíváme dědění nebo se můžete setkat s pojmem dědičnost (inheritance), který umožňuje od existující třídy odvodit třídu novou. Odvozená třída bude obsahovat všechny složky a všechny metody, od níž byly odvozeny a k nim lze přidat nové datové složky a nové metody. Při dědění nelze odstranit žádnou datovou složku ani žádnou metodu odstranit, ale můžeme ji překrýt neboli změnit (override) některou z metod předka (Virus, 2021).

Zapouzdření – zapouzdření (encapsulation), spojuje kód a data dohromady a chrání je před zásahem zvenčí. Kombinací kódu s daty vznikne objekt, který řídí zapouzdření. Uvnitř těchto objektů může být obsah soukromý (privat) nebo veřejný (public) V případě privátního přístupu nepůjdou volat zvenku metody a tím ochráníte svá data a vnitřní stav objektu (Barášek, 2023).

Polymorfismus – metoda se stejným názvem může měnit své chování v závislosti na počtu nebo typu parametrů. Výhoda polymorfismu spočívá v omezení složitosti tím, že určena obecná třída procedury povoluje pouze jediné rozhraní. Pak již záleží pouze na překladači, kterou konkrétní proceduru vybere pro danou situaci (Virus, 2021).

Přetěžování – jedna třída může obsahovat více metod se stejným identifikátorem, pokud mají jiný počet, typ nebo pořadí parametrů. Výběr metody záleží na overload resolution, což je systém, který vybírá nejvhodnější přetížení na základě předdefinovaných pravidel. Overload

resolution obsahuje seznam argumentů a množinu použitelných kandidátů. V případě, že množina obsahuje jednoho kandidáta, je tento kandidát zvolen. Pokud jich obsahuje více, je zvolen takový, který je lepší než ostatní. Může nastat případ, že je objeven více než jeden vhodný kandidát, který je lepší než ostatní, pak není výběr jednoznačný a nastane chyba (Janáček, 2014).

6. Jak učit programování na ZŠ

Programování je praktická činnost, kdy žáci vidí ihned výsledek své práce ihned po spuštění programu. Učitel by měl využít výhody zpětné vazby, kterou mu počítač poskytuje. Program žákovi vyhodnotí, zda pracoval správně nebo naopak zda udělal nějakou chybu. Vaníček a kolektiv doporučují ve své knize při výuce programování dodržovat tyto zásady. Zásady jsou postaveny pro práci v programu Scratch, avšak je lze využít pro výuku programování všeobecně. (Vaníček a kol., 2020).

6.1. Nesdělovat žákům správná řešení

Učitel by měl být spíše moderátorem, než říkat žákům správné řešení, ale nechat ukázat výsledek. Pokud program vyhodnotí, že žák nepostupoval správně a v jeho kódu bude chyba, může žákovi doporučit spustit scénář znova ať si jej zkontroluje, tím se žák bude učit samostatnosti. Učitel by neměl vyžadovat striktní řešení dle jeho názoru nebo dle postupu v učebnici, protože většinou neexistuje jedno správné řešení. Klidně se může stát, že žák vymyslí své vlastní řešení, které bude obsahovat více bloků, ale výsledek bude stejný. Na druhou stranu není vhodné nechat žáka používat bloky, které ještě řádně nezná, ale můžou mu usnadnit řešení úkolu (Vaníček a kol., 2020).

6.2. Neučit definice a pamětní poznatky

Zaměřit se na praktické úkoly místo učení a opakování definic. Je tedy zbytečné, žáky vyvolávat, přezkušovat z definic, zbytečně tím budete ztrácet čas (Vaníček a kol., 2020).

6.3. Obrnit se trpělivostí

Žákům bude určitou dobu trvat, než se budou orientovat v programech využitých pro výuku programování. Může to trvat klidně až tři měsíce, ale jakmile pochopí celý systém, budou již pracovat automaticky a samostatně (Vaníček a kol., 2020).

6.4. Očekávat živé a rušné prostředí

Žáci si budou rádi ukazovat návrhy svých řešení, při práci ve skupině. Učitel by si měl pohlídat, aby žáci diskutovali pouze na dané téma výuky nebo zadaného úkolu (Vaníček a kol., 2020).

6.5. Nechat žáky pracovat ve dvojicích

Při programování je vhodné pracovat v menších skupinkách nebo dvojicích. Žáci si mohou radit, argumentovat a spolupracovat spolu. V tomto případě může být nevýhodou zdání, že slabší žáci budou opisovat od zdatnějších, ale i kdyby k tomu docházelo, vždy se něco naučí a můžou se spolužáky spolupracovat. Druhý případ může být ten, že žák bude chtít pracovat sám. I toto není problém nechat jej pracovat, a pokud narazí na složitější úkol, může využít své spolužáky. (Vaníček a kol., 2020).

6.6. Hodnotit praktické dovednosti

Známku zde určit by mohlo být jednoduché, pokud scénář funguje na první pokus, zaslouží si žák jedničku, pokud na druhý dvojku atd. Tento způsob však je sice jednoduchý, ale v praktickém pojetí není moc vhodný. Velmi vhodné je spíše pozorování žák při práci ve skupině nebo při konzultaci s žákem o jeho rozpracované práci. Svůj příklad hodnocení žáka uvádí Vaníček takto: „*Tomáši, vidím, že se snažíš, mám radost z toho, jak znovu a znovu zkoušíš úlohu opravit. Postavit domek se ti opravdu povedlo. Ještě ti nejde dobře poznat, jak mají být bloky ve scénáři uspořádány, nevíš, jak přijít na chybu ve scénáři. Tady se potřebuješ zlepšit. Za dosavadní práci ti dávám dvojku (do žákovské knížky)*“ (Vaníček a kol., 2020, s.13).

7. Webová aplikace

Webovou aplikaci můžeme stručně definovat jako software, který je zprostředkovaný internetovým prohlížečem. Výhodou je, že nemusíte oproti desktopovému software nic instalovat do svého počítače. Vývoj webové aplikace je levnější než vývoj software na zakázku, kde pro vývoj na android nebo iOS platíte zvlášť. Pro její chod stačí připojení k internetu, a když do internetového prohlížeče zadáte požadovanou adresu. Vytváření webových aplikací funguje na stejných technologiích jako webové stránky. Používají tedy HTML, CSS, JavaScript, PHP a další. Právě z toho důvodu webové aplikace od webových stránek nemusíte na první pohled rozeznat, ale webové aplikace jsou doplněné o pokročilejší funkcionality. Webové aplikace se více blíží k počítačovým programům (Kad'ousková, 2021).

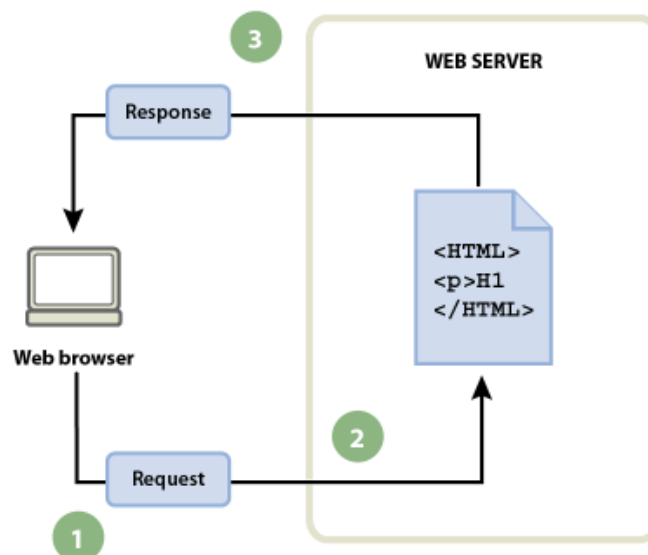
Oproti tomu nevýhodou webových aplikací je závislost na připojení k internetu, případně přístup webových aplikací k hardware, na kterém běží, nemůžete například použít fotoaparát nebo posílat „push“ notifikace. U distribuce programu může být rizikem ukončení podpory či dalšího vývoje. Webové aplikace můžete používat do té doby, dokud poběží na serveru, oproti tomu software na desktopu vám poběží, dokud bude program kompatibilní s počítačem a jeho systémem, na kterém běží (Kad'ousková, 2021).

7.1. Jak funguje webová aplikace

Webová aplikace je soubor statických a dynamických webových stránek. Statická webová stránka se nezmění, když si to návštěvník webu vyžádá: webový server odešle stránku do webového prohlížeče, který si to vyžádal, beze změny. Naproti tomu dynamická webová stránka je před odesláním do prohlížeče, který si ji vyžádal, upravena serverem. Měnící se povaha webu je důvod, proč se nazývá dynamický (Adobe, 2021).

7.1.1. Zpracování statických stránek

Zpracování statických webových stránek můžete vidět na obrázku 4. Webový prohlížeč požádá o statickou stránku (1), webový server vyhledá stránku (2) a nakonec webový server odešle stránku do prohlížeče (3) (Adobe, 2021).

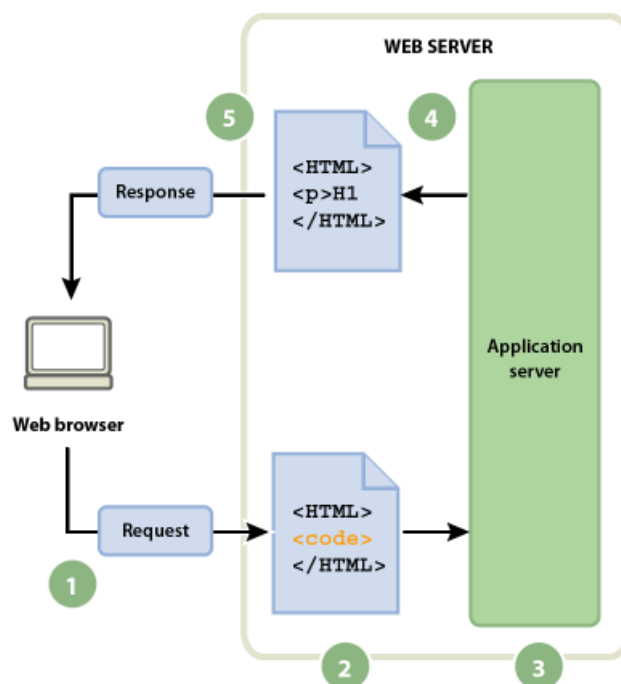


Obr. 4 Zpracování statické webové stránky

(Zdroj: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>)

7.1.2. Zpracování dynamických stránek

Oproti tomu zpracování dynamických stránek potřebuje pro zpracování aplikační server. Webový prohlížeč pošle dotaz na zobrazení dynamické stránky (1) Webový server stránku vyhledá a předá ji aplikačnímu serveru (2) Tento server si přečte kód na stránce a dokončí stránku dle instrukcí v kódu (3), a nakonec tento kód ze stránky odstraní a předá ji opět webovému serveru (4). Výsledkem je opět statická stránka, kterou webový server pošle do prohlížeče, který si o zobrazení této stránky požádal (5). Princip můžete vidět na obrázku 5 (Adobe, 2021).



Obr. 5 Zpracování dynamických stránek

(Zdroj: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>)

8. Doména a webhosting

8.1. Doména

Pokud chcete provozovat své stránky, je potřeba si pro ně zaregistrovat vlastní doménu, která je vlastně identifikací vytvářeného webu. Podmínkou pro označení domény je její jedinečnost. V síti se nesmí objevit dva naprosto stejné názvy domén. Registrace se provádí přes internet, kde se ověří, zda navrhovaný název domény již někdo používá nebo je volně k dispozici (Bureš, 2021).

Celosvětový systém domén je tvořen do struktury, která má tři úrovně a tomu odpovídá i název domény. Ač nelogicky jednotlivé části názvu domény odpovídají jednotlivým úrovním domény. Proto poslední část názvu domény je doménou prvního řádu a často označuje označení státu, na kterém je web provozován. Pro českou republiku je tedy označení CZ, pro Slovensko SK, Německo má DE a známou je také doména TO, jež patří Tongu nebo TV, které patří Tuvalu. Mezi známé domény prvního řádu, které nepatří žádnému státu, můžeme vzpomenout

ORG (neziskové organizace), COM (komerční organizace) nebo GOV (instituce vládní správy) (Bureš, 2021).

Doména druhého řádu označuje vlastní provozovatele webu a nachází se na předposledním místě. Na začátku názvu domény najdeme doménu třetího řádu a ta může mít dvojí charakter. Velmi často je označena písmeny www (world wide web) nebo může rozšiřovat název stránky kdy, pokud nechcete platit za registraci domény, využijete již domény zaregistrované např. libor.webnode.cz (Bureš, 2021).

Jak vybrat správný název domény radí Fiala Tomáš ve svém článku na webu nastartujtese.cz kde doporučuje dodržovat pět základních pravidel. I když je článek z roku 2017 stále může dobře posloužit pro správný výběr názvu domény, jelikož název domény vás bude provádět po celou dobu funkčnosti webu.

- 1. Název webu stručný, ale výstižný** – při volbě názvu domény jsou k dispozici všechna písmena anglické abecedy, velká písmena, latinské číslice a pomlčky. Není potřeba přehánět délku názvu domény, krátké názvy se dají lépe zapamatovat. Pokusit se můžete vymyslet název spojením podstatného jména s předložkou např. „zavodou.cz“ nebo s přídavným jménem „nejlepsiinformatika.cz“ případně lze zkusit u cílové skupiny vzbudit emoční odezvu ještě dříve, než vůbec váš web navštíví „spokojenzak.cz“. Lidé si dokáží taky dobře zapamatovat zkratky a jejich vyslovování „imak.cz“ (Fiala, 2017).
- 2. Název dle účelu projektu** – v začátcích rozvoje internetu a názvu internetových stránek se lidé snažili vložit klíčové slovo do názvu domény, čímž chtěli dosáhnout prvních příček ve vyhledávačích, ale moderní vyhledávače již hledají weby s relevantním obsahem. Tím pádem, pokud je již název domény obsazen není potřeba zoufat, ale stačí mít na webu kvalitní obsah. Doménu vybírejte podle cílové skupiny, pro koho je určena, aby z názvu šlo pochopit jaké je téma webu, a čím se zabývá např. slevomat.cz (Fiala, 2017).
- 3. Nadčasová varianta** – při registraci domény je potřeba uvažovat dopředu, protože pokud si zaregistrujete název domény např. naucsescratch.cz a později byste chtěli na stejné doméně učit python nemusí být žákům jasné, že to najdou u vás (Fiala, 2017).
- 4. Nesnažit se být světový za každou cenu** – často se stává, že při registraci domény se snažíme užít angličtinu, která samozřejmě nemusí být špatně např. při doméně, jež se zabývá IT oborem, protože angličtina je pro programátory základním jazykem. Většina Čechů angličtinu na dobré úrovni neovládá a může název webu různě komolit,

proto je výhodnější pojmenovat web česky. Pokud máte v plánu stránky prezentovat i v zahraničí můžete zvolit národní doménu s příslušnou koncovkou. Tím dostanou vaše stránky více na důvěryhodnosti. Pokud budete stránky používat pouze u nás je dobré se přesvědčit dopředu, zda jsou pod vaším názvem domény .cz, .com a .eu (Fiala, 2017).

5. **Obsazenou doménu lze odkoupit** – ve světě internetu je již hodně domén zakoupených, a ne všechny jsou plně využívány, popřípadě jsou stránky prázdné a vy se můžete pokusit s majitelem domény spojit a název od něj odkoupit. Také lze sledovat, kdy vyprší expirace domény a pokud majitel nezaplatí za její prodloužení, můžete jí koupit. Před koupí takovéto domény není k zahzení si prověřit historii stránek, zda předchozí obsah nebyl nějak závadný (Fiala, 2017).

Rada navíc: Pomlčka nebo bez pomlčky – Fiala Tomáš doporučuje koupit doménu jak s pomlčkou, tak bez pomlčky a doménu s pomlčkou přeměrovat na stránky bez pomlčky. Někteří jedinci si totiž nechávají zaregistrovat různé názvy domén, aby pak za stránky dostali královsky zapláceno, nebo se spoléhají na podobnost v názvu domény a využijí dobrého jména. Ověřit volnost domén lze na stránkách, kde si budete doménu registrovat (Fiala, 2017).

8.2. Webhosting

Webhosting nebo někde uváděno i jako hosting je pronájem prostoru pro své webové stránky, které budou na cizím serveru. Pronajímatel se označuje jako poskytovatel webhostingu nebo provider. Webhosting vám umožňuje umístit své webové stránky na internet a nemusíte mít vlastní server. Ceny za pronájem serverů se pohybují v rozmezí pár korun až po několik tisíc za měsíc. Lze sehnat i webhosting zdarma, avšak tento je určen například pro testování nebo jen nenáročné projekty. Cenou za webhosting zdarma je ovšem vložená reklama na vašich stránkách, omezená velikost webu, požadovaná aktivita na webu. Pro nejnáročnější stránky, kde požadujeme spolehlivost, rychlost, podporované služby a technologie je lépe si pronajmout celý server (Procházka, 2012).

Při výběru poskytovatele webhostingu je dobré si dopředu ověřit, zda podporuje programovací jazyk PHP, přístup k databázi a instalaci některého redakčního systému pro tvorbu webu (Bureš, 2021).

9. Technologie využité při vývoji interaktivní webové aplikace

Základním programovacím jazykem je HTML, který postupně prochází vývojem, ale i ten má své limity. Ne všechny požadavky na web je schopen zvládnout, a proto se k tomuto jazyku přidaly další, a to především JavaScript a PHP.

9.1. Inkscape

Je bezplatný a open source (software s otevřeným zdrojovým kódem) editor vektorové grafiky při GNU/Linux, Windows i macOS. V inkscape je možné tvořit umělecké i technické ilustrace jako například ikony, kliparty, loga, diagramy, vývojové diagramy. Jeho standardizovaným formátem je SVG, který podporují mnohé další aplikace včetně webových prohlížečů. Dokáže naimportovat a vyexportovat různé druhy formátů (SVG, AI, EPS, PDF, PS, PNG). Má vícejazyčnou podporu a je rozšiřitelný pomocí doplňků. Na rozvoji této aplikace se podílí komunita napříč celým světem a je zpracováno mnoho výukových materiálů, které lze při tvorbě obrázků využít (Inkscape, 2023).

9.2. HTML

Html neboli HyperText markup Language je značkovací jazyk pro hypertext. Patří mezi základní programovací jazyky pro tvorbu stránek v systému World Wide Web a tento jazyk umožňuje aplikaci dokumentů na internetu (Lalík, 2013).

Webová stránka, která je psaná pomocí značkovacího jazyka HTML se skládá ze dvou částí. První část jsou texty, které budou na stránkách zobrazeny a řídicích příkazů jazyka HTML, které se značí jako elementy, ale hojně používané je pojmenování tagy. Text, který při tvorbě webové stránky zapíšeme, slouží jako informace pro webový prohlížeč. Pouze elementy mohou ovlivnit způsob, jakým bude prohlížeč stránky zobrazovat.

Při zobrazení zapsaného textu se prohlížeč řídí těmito pravidly:

- prohlížeč ignoruje ENTER,
- prohlížeč nezobrazuje prázdné řádky,
- mezeru prohlížeč vyhodnotí jako oddělovač mezi slovy, nikoliv jako znak, proto se více mezer zobrazí jako jedna mezer,
- tabulátor akceptuje jako jednu mezeru.

Tato pravidla jsou stanovena kvůli různým počítačům a jejich vybavení. Každý počítač má jiné periferní zařízení (monitory), nastavené rozlišení, grafické karty, a proto stejný odstavec se bude různě zobrazovat a může zabrat jiný počet řádků.

Programovací jazyky dokáží v případě chyby, chybu vyhodnotit a odmítne dále pokračovat. Toto ovšem neplatí pro webové prohlížeče.

Elementy jazyka HTML – elementy se uzavírají do znaků `<p>`, mezi něž píšeme identifikační kód zvoleného elementu. Elementy se rozdělují na dva typy:

Párové elementy – těmto elementům je nutné vyznačit jejich začátek i jejich konec. Pro zapsání konce elementu se používá stejný element jako u začátečního, ale je doplněn o lomítko.

Příklad párového elementu může vypadat takto: `<body> </body>`

Nepárové elementy – v dokumentu stojí samostatně, bez koncového elementu. Jak bylo zmíněno, prohlížeče nereflktují enter, a proto v případě potřeby odskoku na další řádek se zapíše element `
`

Pro správnou funkci elementů řada z nich potřebuje další informace, které dostanou ve formě atributů. Těmto atributům je přiřazena potřebná hodnota. Přiřazení se provádí pomocí rovnítko, za kterým následuje hodnota zapsaná v uvozovkách. Atribut i s hodnotou se zapisuje uvnitř elementu mezi `< >` a od kódu je oddělen mezerou. Pro jeden element není problém, aby obsahoval více elementů a jejich pořadí v zápise je libovolný. Kód elementu musí stát na prvním místě (Laurenčík, 2019).

Pro jednodušší práci při psaní kódu má řada atributů nastavenou výchozí hodnotu. Při vynechání atributu u elementu bude použita tato výchozí hodnota (Laurenčík, 2019).

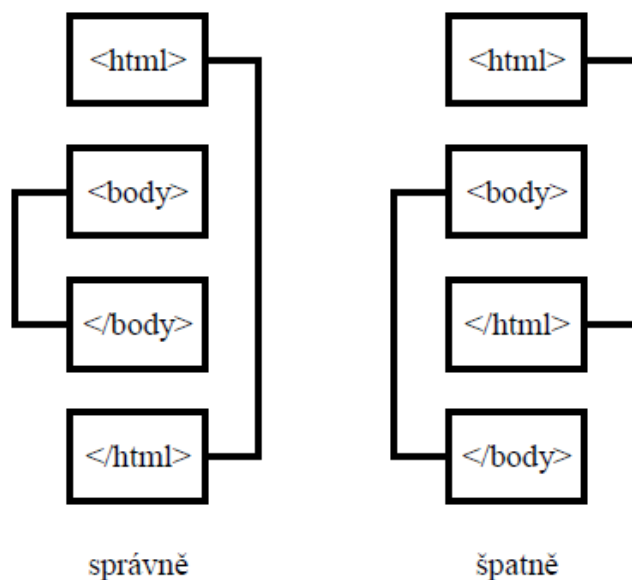
Při psaní elementů se nemusíme obávat, když zápis nevyjde na jeden řádek, protože prohlížeče ignorují přechody na nový řádek (enter). Proto lze element zapsat samostatně na jeden řádek společně s textem nebo jej rozdělit do více řádků.

Webovým prohlížečům bývá zpravidla jedno, zda je zápis zapsán malými nebo velkými písmeny. Pro korektní zápis platí nepsaná zásada, aby počáteční i koncový element byl zapsán stejně, buď malými, nebo velkými. Většinou se používá zápis s malými písmeny (Laurenčík, 2019).

Příklad zápisu párového elementu s nastavením atributu může vypadat takto:

`<strong style="color:red">Text zobrazený na webových stránkách`

Vnořování – párové elementy je možné do sebe vnořovat a tím vytvořit vícestupňové hierarchie, ale musíte si pohlídat, aby se dané elementy mezi sebou nekřížily. To znamená, že element, který je otevřen jako první, musí skončit jako poslední. Této hierarchie se hojně využívá při tvorbě struktury dokumentu HTML. Přehled křížení elementů je na obr. 6.



Obr. 6 Vnořování elementů

(Zdroj: Vlastní snímek obrazovky)

Co HTML zvládne

HTML obsahuje elementy, díky nimž lze vytvářet, formátovat či upravovat webové stránky. Můžeme jej využít k nastavení vzhledu, velikosti zvolenému textu, vkládat obrázky, animace ve formátu GIF, vytváření formulářů, dotazníků, tabulek, vložení odkazů na další stránky nebo na dokumenty na jiných stránkách, odkazy na video či audiosoubory (Písek, 2014).

Co naopak HTML nezvládne

HTML nedokáže vytvořit ověřování uživatele, vytváření dynamických nabídek, vysouvací menu apod. HTML využívá statické obsahy, proto se dokumenty nedokáží automaticky měnit. V případě, že budete něco z tohoto na svých stránkách potřebovat, budete muset sáhnout k jiným webovým technologiím (Písek, 2014).

9.3. CSS

V počátcích psaní webu, kdy ještě neexistovali kaskádové styly, se obsah stránky formátoval pomocí HTML elementů. To sebou přinášelo další tagy např. ``, který měl za úkol nastavit barvu a velikost písma. Tyto tagy neumožňovaly složitější formátování, které bylo potřeba pro lepší vzhled nebo vlastnosti stránky. Oproti tomu vznik kaskádových stylů umožnil rozdělení vzhledu dokumentu od jeho obsahu. Tímto oddělením došlo k lepší přístupnosti webu a zlepšení použitelnosti. Mezi hlavní výhody patří zejména větší možnosti formátování, rychlejší načítání stránek, menší zatížení serveru a snazší správa větších prezentací pomocí CSS šablon (Štráfelda, 2023).

Pro příklad použití kaskádových stylů využijeme předchozí nadpis h1 a předvedeme tři možné způsoby úpravy nadpisu pomocí kaskádových stylů. Výsledek vzhledu nadpisu bude vždy stejný, ale cesta k výsledku bude jiná.

Inline styl – takto psaný kaskádový styl je napsán přímo pomocí atributu **style**. Zápis bude vypadat takto:

```
<h1 style="color: blue">Modrý nadpis</h1>.
```

Tento způsob zápisu není moc vhodné řešení, protože je součástí základního kódu a dále při použití u každého prvku dochází k narůstání objemu kódu, což může způsobit zpomalení webu a k nepřehlednosti kódů (Štráfelda, 2023).

Interní styly – i v tomto způsobu se využívá tag `styl`, ale zápis je napsán v hlavičce dokumentu, ale aplikují se na všechny odpovídající elementy na dané stránce.

```
<style>
h1 {
    color: blue;
}
</style>
```

Výhodou a zároveň nevýhodou je, že takto napsané pravidlo se použije u všech nadpisů první úrovně. Při změně stylu u nějakého prvku na webu, musíme upravit CSS v HTML kódu všech stránek a při načítání stránek načítáme kaskádové styly znovu (Štráfelda, 2023).

Externí styly – pro tento způsob zápisu je potřeba vytvořit soubor s koncovkou css (style.css) a v hlavičce HTML zápisu musíte tento soubor připojit. Pro připojení se používá zápis: `<link rel="stylesheet" href="style.css">`

Tímto se bude po zápisu externí soubor načítat při spuštění stránky pouze jednou, takzvaně se nakešuje do prohlížeče a tímto dojde ke zrychlení při procházení dalších stránek u daného webu.

Zápis v souboru style.css souboru bude vypadat takto:

```
h1 {  
    color: blue;  
}
```

Můžete vidět, že tento způsob zápisu je nejjednodušší. I v tomto případě se bude změna stylu používat pro všechny nadpisy první úrovně, což není vždy vhodné, a pokud potřebujete upravit každý nadpis založený na tagu `<h1>` musíte využít selektory (Štráfelda, 2023).

Selektory – slouží ke konkrétnímu označení prvku, který chceme v kaskádových stylech upravovat. Kaskádové styly nabízí řadu selektorů a vlastnosti jim lze přiřazovat podle typu elementu, podle atributů (id nebo class) podle umístění elementu v hierarchickém stromu dokumentu. Jak můžete vidět na příkladu, určili jsme nadpisu h1 třídu (class) s názvem nadpis1. Název třídy se uvádí za uvozovky.

```
<h1 class="nadpis1">Nadpis h1</h1>
```

V souboru style.css bude zápis vypadat takto:

```
.nadpis1 {  
    color: blue;  
}
```

Tečka se užívá u atributu class. Mezi další užívané atributy v kaskádových stylech patří.

atribut „id“

div pro všechny elementy typu „div“

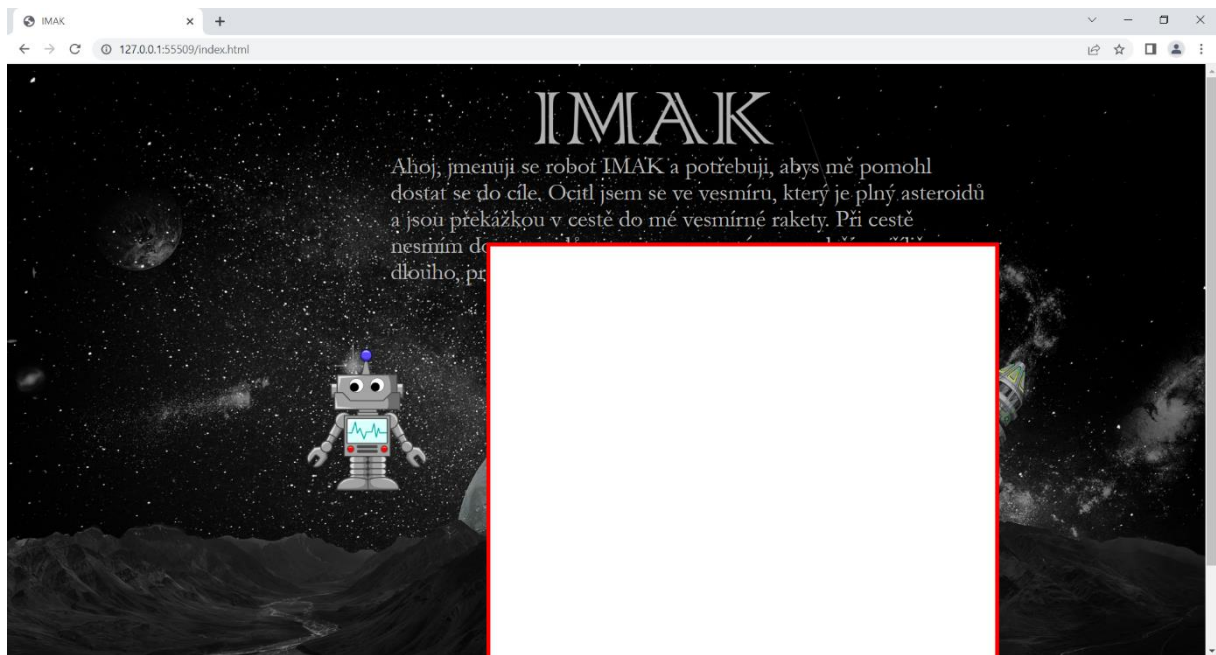
Kaskádování v CSS – kaskádování znamená, že jednotlivá pravidla se dědí k dalším prvkům, které jsou v hierarchickém stromu dokumentu položené níže. Vlastnost nám tímto kaskáduje. Pokud definujete textu odstavce zelenou barvu pak tuto barvu zdědí prvek, který se nachází v odstavci (Štráfelda, 2023).

Specifika selektorů v CSS – jedná se o pravidla, která všechny prohlížeče implementují stejně a jsou určující, jak se mají zachovat v případech kdy dojde ke křížení selektorů. Zde platí, že pravidlo přiřazené pomocí třídy je důležitější než pravidlo přiřazené podle typu elementu (Štráfelda, 2023).

Komentáře – při psaní v kaskádových stylech je vhodné používat komentáře, které slouží pro lepší přehlednost a v případě, že bude potřeba stránku upravit po delším časovém horizontu. Pro zápis se používá lomítko / v kombinaci s hvězdičkou * a na konci v opačném pořadí. Zápis komentáře v CSS vypadá takto:

```
/*Toto je zápis komentáře*/
```

Responzivita – pro zobrazení stránek na různých obrazovkách nebo zařízeních jako je mobilní telefon, tablet nebo obrazovka počítače je potřeba nastavit responzivitu stránek. Pokud bychom tento krok neudělali, vaše stránky by se na různých zařízeních a v různém přiblížení zobrazovaly s překrytím. Jak vypadají stránky bez nastavené responzivity můžete vidět na obrázku č. 7. Pro příklad jsme do stránek vložili čtverec o velikosti 650 x 650px, který nemá nastavenou responzivitu.



Obr. 7 Responzivita

(Zdroj: Vlastní snímek obrazovky)

Pro správné nastavení responzivity David Šetek ve svém videu na youtube doporučuje pro nastavení responzivity tyto kroky.

1. v indexu

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!--Viewport: okno prohlížeče, width=device-width: šířka zařízení, initial-scale:
výchozí zoom je 100% -->
</head>
```

2. v CSS nastavené šířky z „px“ (pixely) nastavit na „%“ a používat max-width. Například nastavení nadpisu stránky bude vypadat jako text níže.

```
#nadpis img {
    position: absolute; /*prvek je umístěn k jeho prvnímu umístění*/
    top: 2%; /*umístění od horního okraje stránky*/
    left: 44%; /*umístění od levého okraje stránky*/
    width: 20%; /*stanovená šířka v %*/
    margin: 20px auto; /*nastavení vnějšího okraje*/
```

```
}
```

3. v CSS používat media queries. Tento příkaz řekne od kdy má platit pravidlo responzivity. Například pokud bude obrazovka menší než 1190px. Pro příklad uvádíme tento zápis:

```
@media only screen and (max-width: 1190px){  
    #nadpis img{  
        top: 3%;    /*umístění nadpisu od horního okraje stránky*/  
        left: 44%; /*umístění nadpisu od levého okraje stránky*/  
        width: 20%; /*stanovená šířka v %*/  
    }  
}
```

Tento zápis bude fungovat pouze v případě, že rozlišení obrazovky bude 1190px a méně (Šetek, 2020).

9.4.Brackets

Jedná se o moderní open-source textový editor kódu pro HTML, CSS a JavaScript. Výhodou tohoto editoru je v jeho jednoduchosti, možnosti instalace doplňků, které vám pomůžou při tvoření stránek. Brackets je synchronizován s vaším prohlížečem a umožňuje živý náhled na průběh tvorby stránek a okamžité úpravy kódu (Github, 2023).

Po stažení instalačního souboru ze stránek brackets.io je potřeba spustit instalaci. Po nainstalování a prvním spuštění doporučujeme doinstalovat, přes správce doplňků, doplněk s názvem HTML skeleton. Tento doplněk vám pomůže snadno vkládat do dokumentu různé prvky HTML kamkoliv, kde budete mít umístěný kurzor (Triangle717, 2023). Na obrázku č. 8 je ukázka vloženého skeletonu pro začátek tvorby HTML kódu.

```

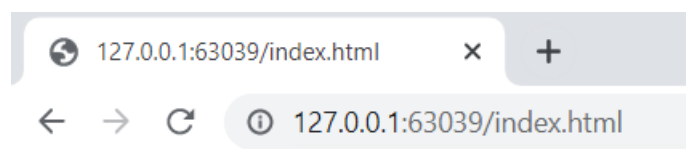
1 <!DOCTYPE html>
2 <html lang="">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title></title>
7     <link rel="stylesheet" href="">
8 </head>
9
10 <body>
11     <script src=""></script>
12 </body>
13 </html>
14 |

```

Obr. 8 Vložený skeleton HTML

(Zdroj: Vlastní snímek obrazovky)

Dalším nezbytným doplňkem je EMMET, tento doplněk nám pomáhá dopisovat s klávesou TAB zbytek rozepsaného Tagu. Bez tohoto doplňku je potřeba tag psát celý což vede ke zbytečné ztrátě času a chybám v zápise. Tagy se používají pro nadpisy, odstavce, seznamy, odkazy, citace a další položky. Často jsou tagy párové a názvy značek a jejich vlastnosti jsou uzavírány mezi úhlové závorky. Například pro hlavní nadpis se používá tag `<h1>Nadpis h1</h1>` a pro podnadpis `<h2>Podnadpis h2</h2>`. Výsledek zápisu těchto tagů můžete vidět na obrázku č. 9.



Nadpis h1

Podnadpis h2

Obr. 9 Ukázka základních tagů

(Zdroj: Vlastní snímek obrazovky)

Takto vytvořené nadpisy mají základní vzhled, který jim určuje zvolený tag. Pro úpravu vzhledu tagu a celkové formátování stránek webu se používají tkz. CSS (z anglického Cascading Style Sheets) neboli kaskádové styly.

9.5. Microsoft Visual Studio

Na svých stránkách Microsoft charakterizuje Visual studio 2022 takto „*Integrované vývojové prostředí (IDE) Visual Studio je kreativní spouštěcí panel, který můžete použít k úpravám, ladění a sestavování kódu a k publikování aplikace. Nad rámec standardního editoru a ladicího programu, které poskytuje většina integrovaných vývojových prostředí (IDE), obsahuje Visual Studio kompilátory, nástroje pro doplňování kódu, grafické návrháře a mnoho dalších funkcí, které zlepšují proces vývoje softwaru.*“ (Visual Studio, 2023).

Tento program se dá stáhnout ve třech verzích a to Community, Professional a Enterprise.

Verze Community je zcela zdarma a v podstatě se jedná o stejnou verzi jako Professional s tím rozdílem, že verzi Community nemůžete provozovat v podnikových sítích (více než 250 PC nebo roční výnos 1 milion USD). Jinak nezbyvá než si zaplatit verzi Enterprise či případně Professional a počítat s omezením (Žůrek, 2023).

Předplatné si lze objednat přímo na stránkách Microsoftu a pohybuje se 45\$ za měsíc u verze Professional nebo v případě licence Enterprise 250\$ za měsíc. Pro potřeby vývoje webové aplikace jsme zvolili verzi Community, která je dostatečná a hlavně zdarma.

Microsoft Visual Studio 2022 je podporováno v těchto 64bitových operačních systémech, pro naše potřeby uvedeme jen ty nejzákladnější a podrobnější informaci o podporovaných operačních systémech můžete nalézt na stránkách Microsoftu. Samozřejmě s vydáním Windows 11 je podpora zabezpečena i zde od verze 21H2 nebo novější v edicích od Home až po Education. Windows 10 je podporován od verze 1909 a podporuje edice Home, Professional, Education a Enterprise. Dalšími operačními systémy, kde může program běžet, jsou Windows Server 2016, 2019, 2022 jak v edici Standard, tak i Datacenter. (Visual Studio, 2023)

Kde naopak program nepoběží, to jsou 32bitové operační systémy a dále Windows 11 v režimu S, Windows Enterprise IoT, Windows 10 IoT Core, Windows 10 Enterprise LTSC, Windows 10 S a edice Windows 10 Team. (Visual studio, 2023)

Minimálními požadavky na hardware je Procesor ARM64 nebo x64. Microsoft doporučuje alespoň čtyřjádrový procesor. Procesory ARM 32 se nepodporují. Paměť RAM je vyžadována nejméně 4 GB, ale pro profesionální řešení se doporučuje 16 GB. Potřeba volného místa na disku záleží podle nainstalovaných funkcí a může se pohybovat v rozmezí 850 MB až 210 GB. Pro běžného uživatele je rozpětí 20-50 GB volného místa. Rozlišení obrazovky bude

fungovat s grafickou kartou, která podporuje minimální rozlišení zobrazení WXGA (1366 x 768), ale studio nejlépe pracuje v rozlišení 1920 x 1080 (full HD). Zajímavostí je, že program může běžet na Windows 365 (což je cloudové řešení), ale zde jsou vyšší nároky na procesor a RAM, minimálně požaduje pro svůj běh 2 virtuální procesory a 8 GB RAM, avšak doporučuje se mít 4 virtuální procesory a 16 GB RAM. (Visual studio, 2023)

Program je dostupný v těchto 14 jazycích: angličtina, čínština (tradiční), čínština (zjednodušená), čeština, francouzština, italština, japonština, korejština, němčina, polština, portugalština (Brazílie), ruština, španělština a turečtina. Jazyk si může uživatel zvolit při instalaci, popřípadě bude odpovídat jazyku nastavenému v systému Windows (Visual studio, 2023).

Pro instalaci programu je potřeba mít práva správce systému. Pro instalaci je potřeba mít nainstalován nejméně .NET Framework 4.5.2 nebo vyšší a ke spuštění programu je potřeba mít nainstalováno rozhraní .NET Framework 4.8, toto rozhraní je součástí instalačního souboru proto je rozdíl ve verzích pro instalaci a spuštění (Visual studio, 2023).

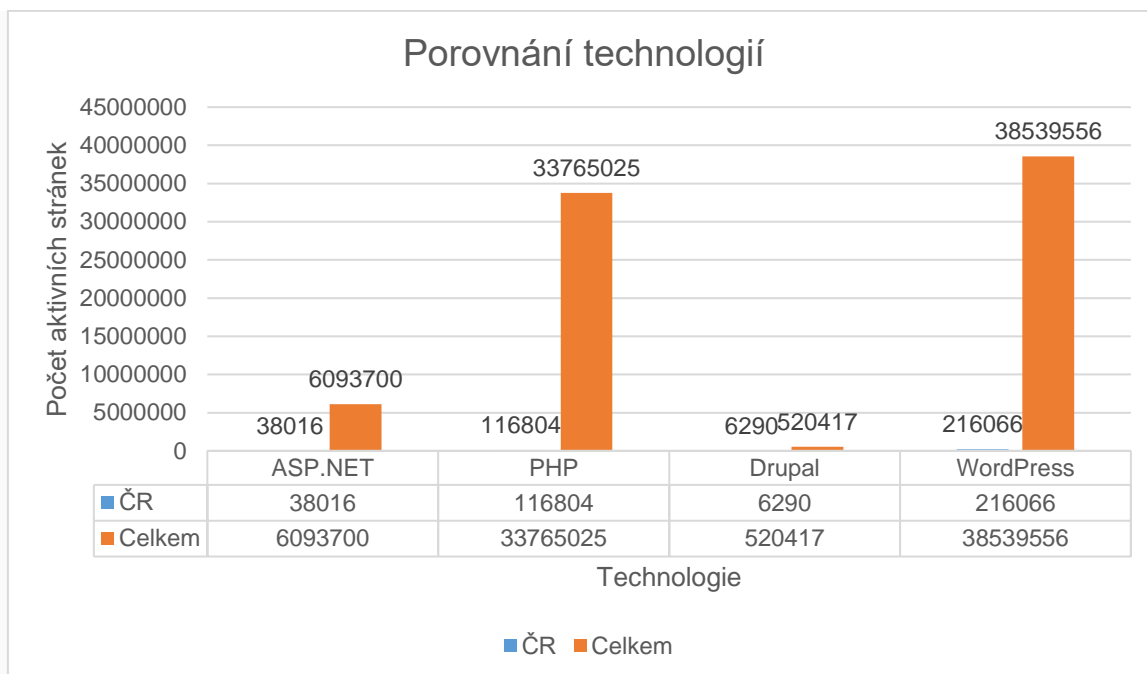
9.6. ASP.NET a .NET

Microsoft na svých stránkách definuje ASP.NET takto: „*NET je vývojářská platforma složená z nástrojů, programovacích jazyků a knihoven pro vytváření mnoha různých typů aplikací.*“ (Microsoft, 2023).

V platformě .NET lze nalézt programovací jazyky C#, F# a Visual Basic, dále knihovny pro práci s řetězci a další editory a nástroje pro Windows, Linux, macOS, a Docker. ASP.NET je rozšiřující platformou .NETu, který je zaměřen na nástroje a knihovny pro vývoj webových aplikací kde mezi rozšíření patří základní framework pro zpracování webových požadavků v C# nebo F#, knihovny pro běžné webové vzory např. Model View Controller (MVC), autentizační systém v němž jsou knihovny, databáze a stránky šablon pro zpracování přihlášení, včetně vícefaktorové autentizace nebo autentizace třetích stran jako je Google, Twitter a další. Pro vývoj webových stránek obsahuje také rozšíření editoru, který zvýrazňuje syntaxe či dokončování kódu, dále syntaxe šablon webových stránek známá jako Razor, který slouží pro vytváření dynamických webových stránek pomocí C#. (Microsoft, What is ASP.NET,2023).

V ASP.NET je tedy vhodný jak pro malé osobní weby, tak i pro velké projekty. Tuto výhodu ovšem vyvažuje nevýhoda v podobě webhostingů, které podporují ASP.NET

a zejména jež jsou zadarmo. Na stránkách BuiltWith.com lze vyhledat statistiky o využití různých druhů technologií při tvorbě webových stránek. Pro porovnání jsme použili technologie ASP.NET, PHP a dva redakční systémy WordPress a Drupal jejichž výsledky můžete vidět v grafu č.1.



Graf 1 Technologie pro tvorbu webových stránek

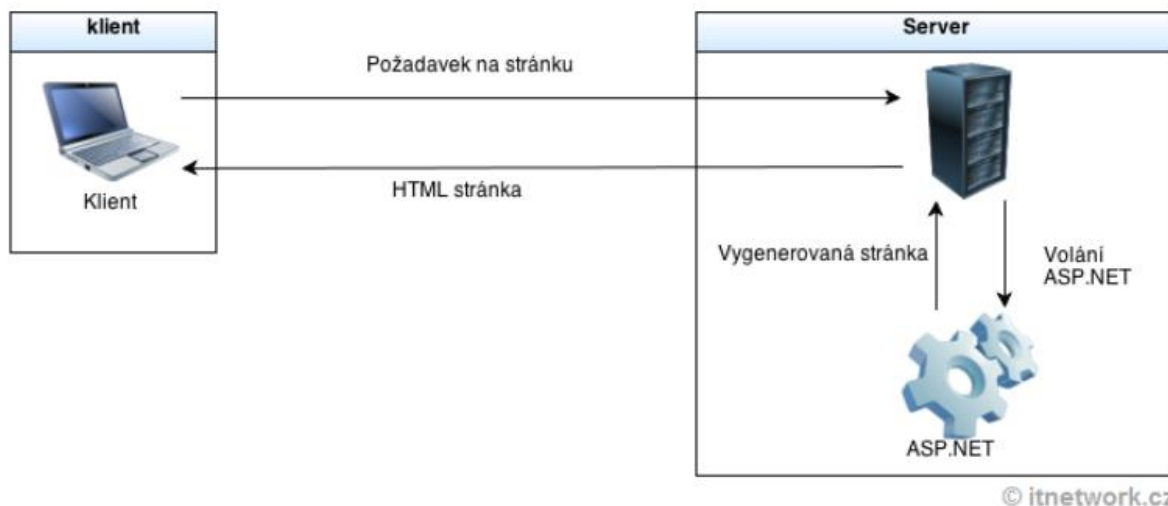
Z uvedeného grafu lze vyčíst, že v současné době má při psaní webů přednost redakční systém WordPress a to pro jeho jednoduchost.

ASP.NET má v ČR k 20. 3. 2023 aktivních 38016 stránek a ve světě 6093700, rozdíl v počtu stránek ve světě a ČR je tak velký, že v grafu nejde skoro zaznamenat sloupce.

Principem ASP.NET je architektura založena na principu klient-server. ASP.NET běží na straně serveru. (itnetwork.cz)

Jelikož má statický web hodně omezení je požadavkem, aby web pracoval dynamicky. V principu se jedná o změnu HTML stránky ještě předtím, než bude klientovi odeslána. Lze tak již do stránek přidávat aktivní prvky jako je přihlašování uživatelů, diskuzní fóra apod.

ASP.NET běží na serveru a na základě požadavku klienta vygeneruje webovou stránku, kterou odešle klientovi. Klientovi se zobrazí pouze výsledné HTML, ze kterého nepozná, že stránka byla zpracována APN.NET. Stránka užívá koncovku .aspx místo .html. Na obrázku č. 10 lze vidět, jak situace vypadá znázorněna graficky (itnetwork.cz, 2023).



Obr. 10 Dynamický web ASP.NET

(Zdroj:https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net?gclid=CjwKCAjw_MqgBhAGEiwAnYOAejuDzOcONwNhSBA6Qz3wm7067j7G5EC2jhXDDsFf_vGA3PQsbbNSaxoCD8wQAvD_BwE)

ASP.NET běží na serveru IIS (Internet Information Services), ale teoreticky je možné jej rozběhnout na serveru Apache jako jeho modul, podobně jako konkurenční PHP. Při instalaci plné verze Visual studia se nainstaluje i server IIS (itnetwork.cz, 2023).

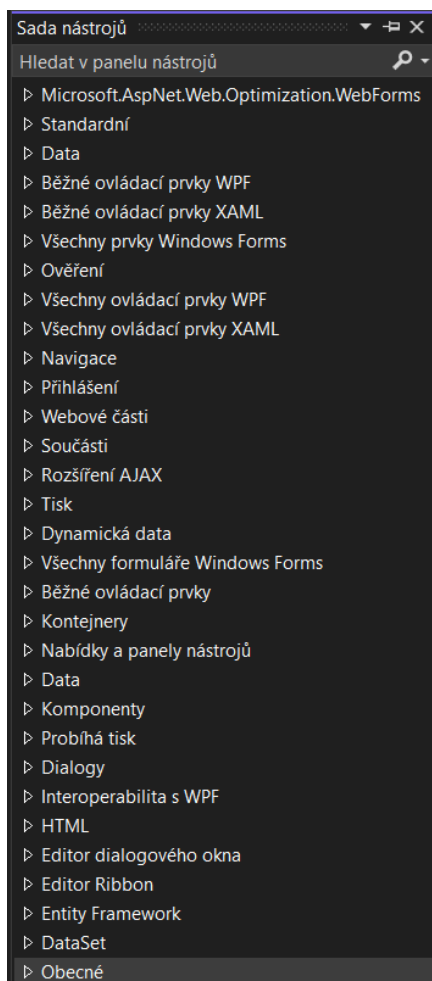
9.7. Web Forms

WebForm je jedním ze čtyřech programovacích modelů pro tvorbu webových stránek ASP.NET a webové aplikace. Tento model je nejstarším z programovacích modelů ASP.NET s událostmi jež jsou řízeny webovými stránkami napsanými kombinací HTML, serverových ovládacích prvků a kódu serveru. Tyto formuláře se kompilují a spouštějí na serveru, který generuje HTML, a zobrazují je webové stránky (w3schools.com).

Programování webových aplikací pomocí webových formulářů je více podobné programování klasických aplikací Windows, jako je Visual Basic než programování klasických stránek ASP. Například při běžném vývoji webových formulářů se nemusíme starat o to, jak, kdo a kdy přesně zavolá překladatel, aby přeložil programový kód formuláře a podobně. A pokud vytvoříme webové stránky pomocí vizuálního programovacího nástroje, jako je Visual Studio 2022, pak může být vše neuvěřitelně jednoduché. Web Forms umožňují tvorbu webových aplikací nezávislých na prohlížeči, protože webový formulář a jednotlivé

ovládací prvky jsou schopny generovat výstupní HTML kód podle typu klienta/prohlížeče, který daný kód požaduje. Web Forms také poskytuje automatickou správu a ukládání informací o stavu stránky (state management). Další velkou výhodou pro programátora je, že Web Forms umožňuje zcela oddělit uživatelské rozhraní webu od jeho programového kódu a jeho silná podpora cache nebo skvělé možnosti ladění (debuging) a sledování(tracing). (Furman, 2002)

Příklad ovládacích prvků můžete vidět na obrázku č. 11.



Obr. 11 Sada nástrojů

(Zdroj: vlastní snímek obrazovky)

Jak lze vidět na obrázku nabídka je opravdu široká a záleží jen na tvůrci webové aplikace, co si vybere a zvolí pro svoji práci. Po vložení ovládacích prvků na stránky jim lze přiřadit potřebné události.

9.8. IIS

Internet Information Services vytvořila společnost Microsoft, proto tato služba běží pouze v operačním systému Windows. IIS obsluhuje požadované HTML stránky nebo soubory. Webový server přijímá požadavky ze vzdálených klientských počítačů a vrací příslušnou odpověď. Tato služba běží jak na lokální síti LAN, tak v internetu. (Techtarget.com)

Služba IIS používá řadu standardních jazyků a protokolů. HTML používá pro vytváření prvků, mezi něž patří text, tlačítka, umístění obrázků a hypertextové odkazy. K výměně informací mezi webovými servery a uživateli využívá protokoly, jako jsou HTTP (Hypertext Transfer Protocol) nebo jeho zabezpečenou verzi HTTPS (HTTP s použitím šifrování pro zvýšení bezpečnosti přenosu dat Secure Sockets Layer neboli SSL). K přenosu souborů používá FTP (File Transfer Protocol) nebo jeho zabezpečená varianta FTPS. Dalším významným protokolem, který běží v IIS je protokol pro odesílání a příjem e-mailů, který se nazývá SMTP (Simple Mail Transfer Protocol) (TechTarget.com, 2023).

9.9. C#

Tento programovací jazyk je nástupcem jazyka C, a jelikož je posazen o půl tónu výše, byla k jeho označení použita značka z hudebního prostředí, která se označuje jako křížek # a patří do skupiny posuvníků. Označení C# neboli cis se přepisuje jako C sharp (Virus, 2021).

Programovací jazyk C# byl představen firmou Microsoft v roce 2002 a jeho původní určení bylo k programování pro Windows, ale v současné době i díky nástrojům jako je Microsoft Visual Studio, slouží k vytváření aplikací pro jiné platformy, mezi nimiž lze vyjmenovat Linux, Adroid i macOS (Virus, 2021).

Tento programovací jazyk se řadí mezi objektový programovací jazyk a je určen k vývoji aplikací pro prostředí .NET. V prvotních fázích byl .NET nástavbou operačního systému Windows, avšak v současné době je jeho součástí (Virus, 2021).

Mezi hlavní výhody tohoto programovacího jazyka patří zejména jeho jednoduchost, obrovská komunita a je blízký dalším programovacím jazykům jako je C#, C++ a JAVA (W3School.com, 2023).

C# pro překlad zdrojového kódu využívá univerzálního pomocného jazyka označovaného Common Intermediate Language (CIL) a ten s využitím dalšího překladače převede do strojového kódu cílového počítače, ale až v době, kdy program spustíme.

Z uvedeného vyplývá, že cílový počítač musí obsahovat překladač, který z CIL provede předklad do strojového kódu. Tomuto počítači říkáme JIT (Just In Time), protože překládá

v době, kdy je zrovna potřeba. Podmínkou je nainstalované prostředí .NET na tomto počítači, které bude kromě JIT obsahovat i další součásti potřebné pro běh programů, které byly vytvořeny v C#. Nevýhodou je, že překladače, které překládají z jakéhokoliv programovacího jazyka, zároveň kontrolují syntaktickou správnost programu – jestli je program zapsán podle jistých formálních pravidel. Tato pravidla zaručují, že mu počítač porozumí. Syntaktická správnost ovšem není zárukou, že program bude dělat to, co si přejeme (Virus, 2021).

V C# lze vytvářet programy pro různé operační systémy, mezi ty nejznámější patří již zmiňované Windows v různých verzích a edicích, Linux, macOS, IOS a Android, tzn. je pro ně k dispozici prostředí .NET (Virus, 2021).

10. Vývoj webové aplikace

V této kapitole si popíšeme, jak jsme postupovali při tvorbě webové aplikace od jejího začátku až po uveřejnění na web a zprovoznění.

Podmínky stanovené na začátku: Jednoduchost, přehlednost, webhosting zdarma.

10.1. Volba projektu

Prvním a zásadním krokem bylo vymyslet, jak daná webová aplikace bude vypadat, co bude dělat a jaké nástroje použijeme k vývoji webové aplikace. Zvolená webová aplikace odpovídá na požadavky vzorových ŠVP uvedených v této práci výše, zaměřena je na 6 ročník a tematický celek v RVP Algoritmizace a programování. Inspirací se částečně stala kniha Programování ve Scratch pro 2. stupeň základní školy.

10.2. Název aplikace

Při volbě názvu jsme postupovali dle doporučení Fialy a byl zvolen název, který se skládá ze zkratky IMÁK (Informatika MÁ Kouzlo). Není vhodné do názvu domény používat diakritiku, proto byl použit název bez diakritiky IMAK. Nakonec stačilo v programu Inkscape vytvořit logo na webové stránky (viz obr. 12). Podmínkou tvorby loga bylo jeho průhledné pozadí a zvolen formát PNG.



Obr. 12 Logo aplikace Imak

(Zdroj: Vlastní zpracování)

Další použité obrázky byly staženy z internetu s licencí Creative Commons. Zde autor obrázku dává své obrázky volně k dispozici pro sdílení, používání apod.

10.3. Psaní kódu v Brackets

Pro nastavení prvotního vzhledu webových stránek jsme se rozhodli využít kódu HTML a CSS pro úpravu vzhledu stránky, který byl vytvořen v editoru Brackets. Pomocí HTML skeletonu bylo provedeno základní nastavení html kódu. Dále byly nalinkovány odkazy na CSS styly a ikonu použitou na webových stránkách. Pro lepší přehlednost byly použity komentáře. Ukázkou nastavení lze vidět na obr. 13.

```
1 <!DOCTYPE html>                                <!-- dokument typu HTML -->
2 <html lang="cs">                                <!-- dominantní jazyk stránky -->
3 <head>                                           <!-- věci které se na stránce nezobrazují -->
4   <meta charset="utf-8">                         <!-- meta = latinsky "za" znaková sada -->
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <!--Viewpont:okno prohlížeče, width=width-device:šířka zařízení, initial-scale: výchozí zoom je 100% -->
7   <link rel="icon" type="image/ico" href="images/favicon.ico">
8   <link rel="stylesheet" href="styly.css">       <!-- nalinkování css stylů -->
9   <link rel="stylesheet" href="grid.css">       <!-- nalinkování css stylů GRID -->
10  <title>IMAK</title>                           <!-- titulek pro vyhledávače -->
11 </head>
12
```

Obr. 13 Hlavička stránek IMAK

(Zdroj: Vlastní snímek obrazovky)

Jednou ze stanovených podmínek byla jednoduchost, proto kód není nijak dlouhý a je současně přehledný, tím byla splněna i druhá podmínka. Ukázkou hotového kódu vidíte na obr. 14.

```

20 <main>
21
22 <section class="informace row"> <!-- s vytvořenou třídou a pojmenováním -->
23 <div class="text">
24 <p>Ahoj, jmenuji se robot IMAK a potřebuji, abys mě pomohl dostat se do mé rakety. Cesta je
    plná asteroidů do kterých nesmím narazit, ale POZOR ve vesmíru nevydržím příliš dlouho, proto
    zkus najít nejkratší cestu.</p>
25 </div>
26 </section>
27
28 <section class="hlavni row">
29 <div class="robot">
30 <div class="col span-1-of-4">
31 
32 </div>
33 </div>
34
35 <div class="hra">
36 <div class="col span-2-of-4">
37 <p>Pravidla hry jsou jednoduchá, při cestě nesmíš narazit do překážek a dojít co nejkratší
    cestou do rakety. Máš dvě možnosti, buď si to jen zkusíš postupnými kroky nebo si můžeš
    zkusit těžší obtížnost. Tady si kroky nastavíš dopředu a pak pustíš IMAKa do akce.<br><br>
    <a href="#" class="btn">ZAČÍT</a>
38 <br><br>
39 </div>
40 </div>
41 </section>
42
43 <div class="raketa">
44 <div class="col span-1-of-4">
45 
46 </div>
47 </div>
48
49 </section>
50
51 </main>
52
53 <footer class="footer">
54 <div class="row">
55 <p>Copywrite &copy; Libor Jurča 2023</p>
56 </div>
57 </footer> <!-- patička -->
58
59 </body>
60 </html>

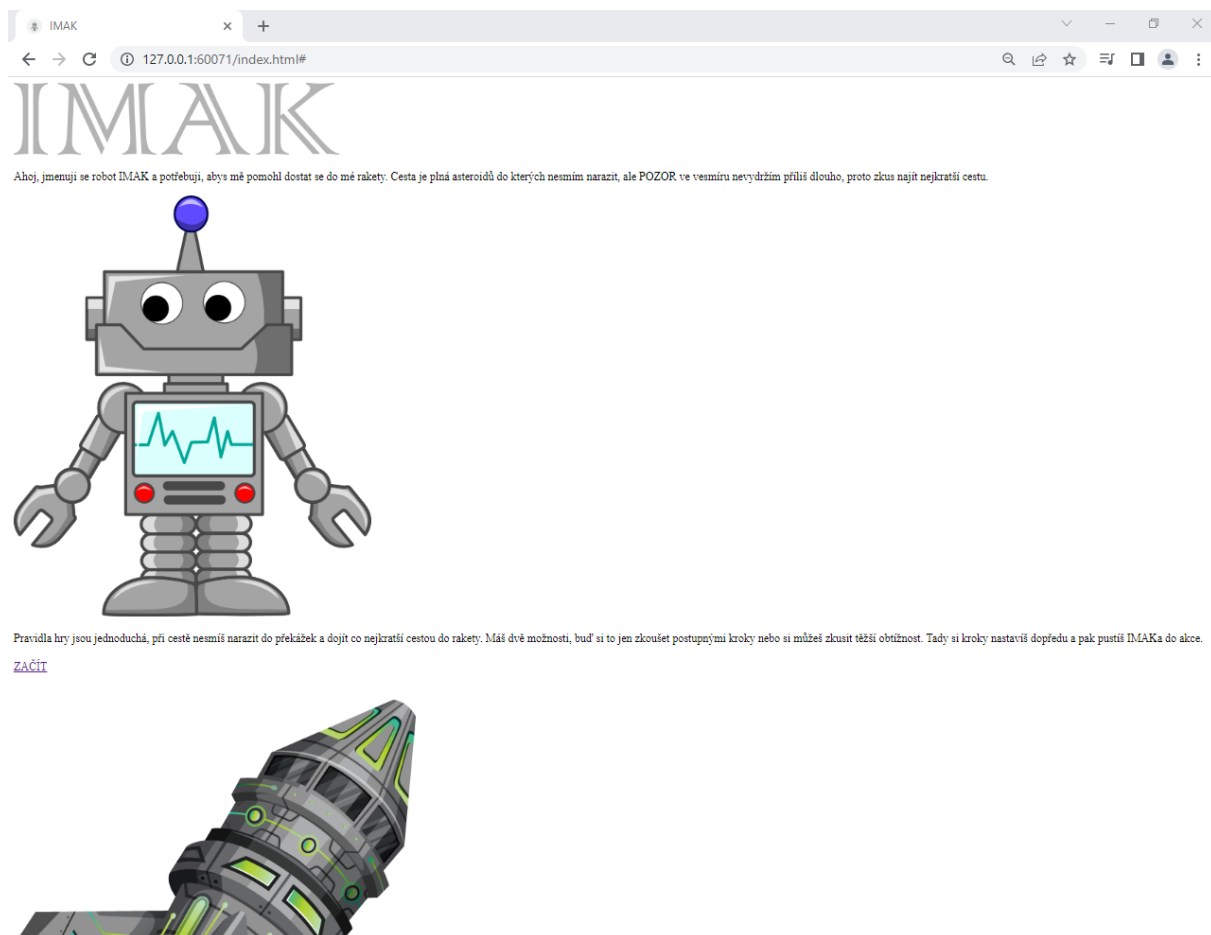
```

Obr. 14 Ukázka HTML kódu

(Zdroj: Vlastní snímek obrazovky)

Tímto vznikly stránky bez použití CSS a po prvotním spuštění vypadaly jako na obr. 15.

Po shlédnutí je jasné, že je potřeba upravit velikost obrázků, Velikost textu, umístění prvků na stránkách apod.

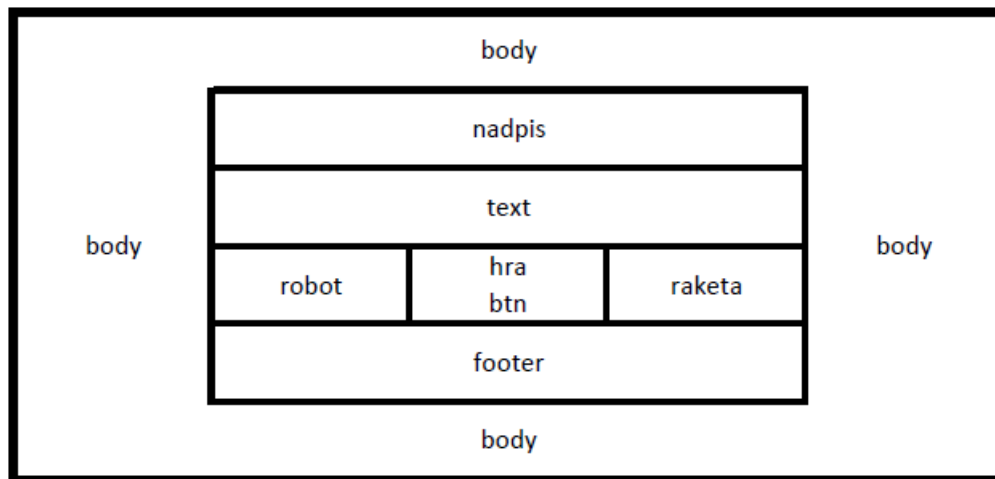


Obr. 15 Prvotní spuštění

Proto je nutné využít CSS kódu pro správný vzhled obrazovky.

Při tvorbě CSS byl využit soubor `grid.css`, který vytvořil Štefek David a dal jej volně k dispozici ve svém videonávodu na youtube. Tento soubor nám posloužil k rozdělení stránek na určité třídy, do kterých byly vloženy kódy. Dalším důvodem využití tohoto souboru byla možnost tvorby responzivity webových stránek, které jsou v současné době nezbytné. Do tohoto souboru nebylo nijak zasahováno, pouze jsme ho nalinkovali a využili.

Hlavním souborem pro vzhled stránky byl vytvořený soubor s názvem `styly.css`. Do tohoto jsme zapisovali jednotlivé vlastnosti použitých objektů. Objektům byly pro práci s nimi vytvořeny třídy (`class`). Pro stanovení tříd jsme vytvořili hrubé rozložení tříd na stránce, výsledek je na obr. 16



Obr. 16 Třídy a rozložení pro CSS

(Zdroj: Vlastní snímek obrazovky)

V průběhu úpravy vlastností prvků na stránkách probíhala kontrola pomocí tkz. živého náhledu, který nám umožňuje aplikace Brackets provádět. Při psaní kódu jsme opět využívali možnosti komentářů obr. 17.

```

1 ▾ * {                               /*výchozí nastavení stránek*/
2     margin: 0;
3     padding: 0;
4     box-sizing: border-box;
5
6 }
7
8 ▾ body {
9     background: url("images/pozadi.jpg");
10    background-size: cover;           /*pozadí se bude chovat jako pozadí za textem*/
11    height: 100vh;
12
13 }
14
15 /*****nadpis IMAK*****/
16 ▾ .nadpis img {
17     display: flex;
18     position: relative;
19     margin: 30px 0px 30px 0px;
20     left: 50%;
21     width: 40%;
22     transform: translate(-50%);
23
24 }
25
26
27 /***** sekce informace*****/
28
29 ▾ .text {
30     max-width: 1000px;
31     color: rgb(190,190,190);         /*barva v rgb,*/ /*rgba(255,255,255,0.6) - barva včetně průhlednosti*/
32     font-size: 30px;                 /*velikost písma*/
33     font-family: Garamond, serif;    /*font písma* serif = patkové písmo / sans-serif = bezpatkové písmo*/
34     font-weight: 500;                /*tlouška písma*/
35     margin: 0 0 0 0;
36     text-align: center;
37     position: relative;
38     left: 50%;
39     transform: translate(-50%);
40     display: flex;
41     flex-wrap: wrap;
42     flex: 1;
43
44 }
45
46
47 /***** hlavní*****/
48
49 ▾ .robot img { /*úprava vlastností obrázku robot*/

```

Obr. 17 Zápís CSS

(Zdroj: Vlastní snímek obrazovky)

Nastavení responzivity byly pomocí souboru grid.css jednodušší. Stačilo jen využít názvy tříd, které byly v souboru vytvořeny a dále nastavit parametry, které se mají v určité velikosti obrazovky změnit. My jsme zvolili velikost do 768px což odpovídá velikosti mobilních zařízení, či tabletů. Zde již stačilo nastavit zobrazení na maximální velikost a zarovnání doprostřed obr. 18.

```
/******responzivita*****/  
/* Mobile devices, tablets */  
@media (max-width: 768px) {  
  
    .col.span-1-of-4 {  
        width: 100%;  
        text-align: center;  
  
    }  
  
    .col.span-2-of-4 {  
        width: 100%;  
        text-align: center;  
  
    }  
  
    .text, .raketa img, .robot img, .hra p {  
        width: 100%;  
        margin: 5px;  
        padding-top: 0;  
  
    }  
  
}
```

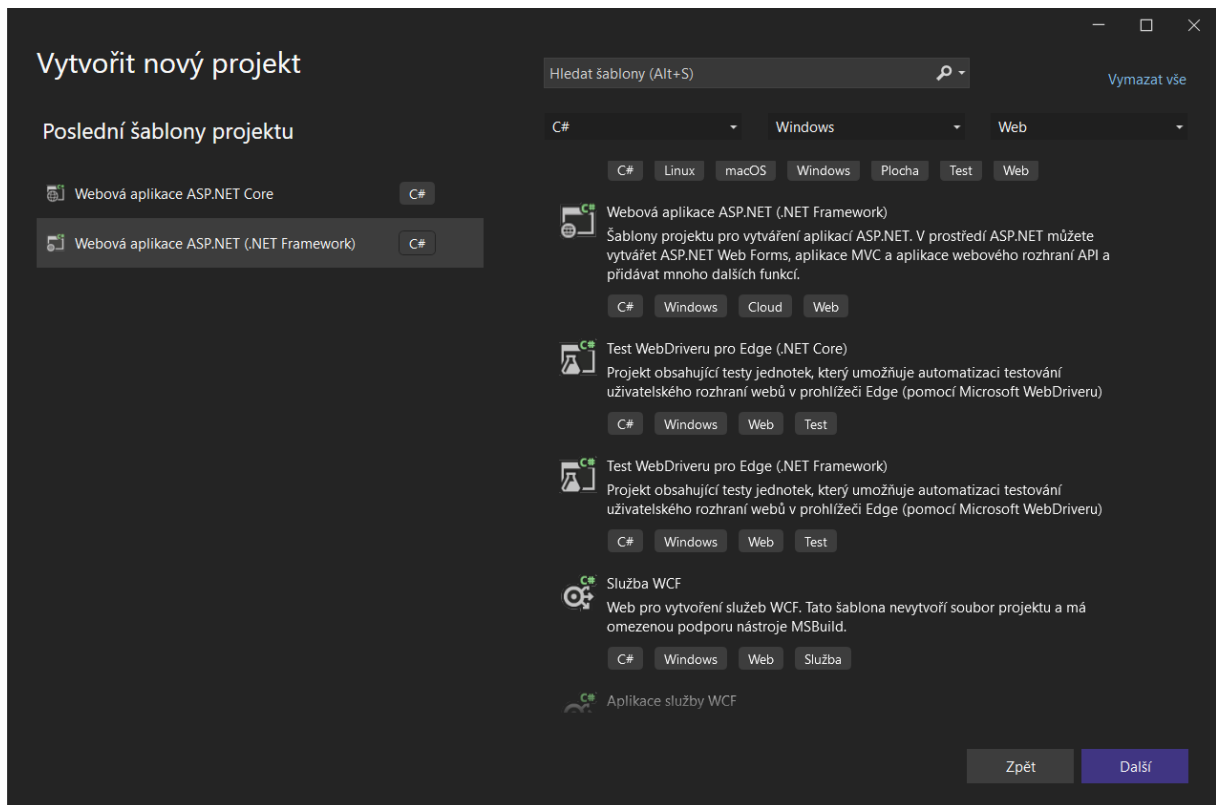
Obr. 18 Nastavení responzivity

(Zdroj: Vlastní snímek obrazovky)

10.4. Práce ve Visual studio 2022

Stažení aplikace Microsoft Visual studia proběhlo přímo na stránkách microsoftu a byla zvolena bezplatná verze Community, která je pro běžného uživatele dostačující. Průběh a doba instalace trvá v závislosti na zvolených doplňcích.

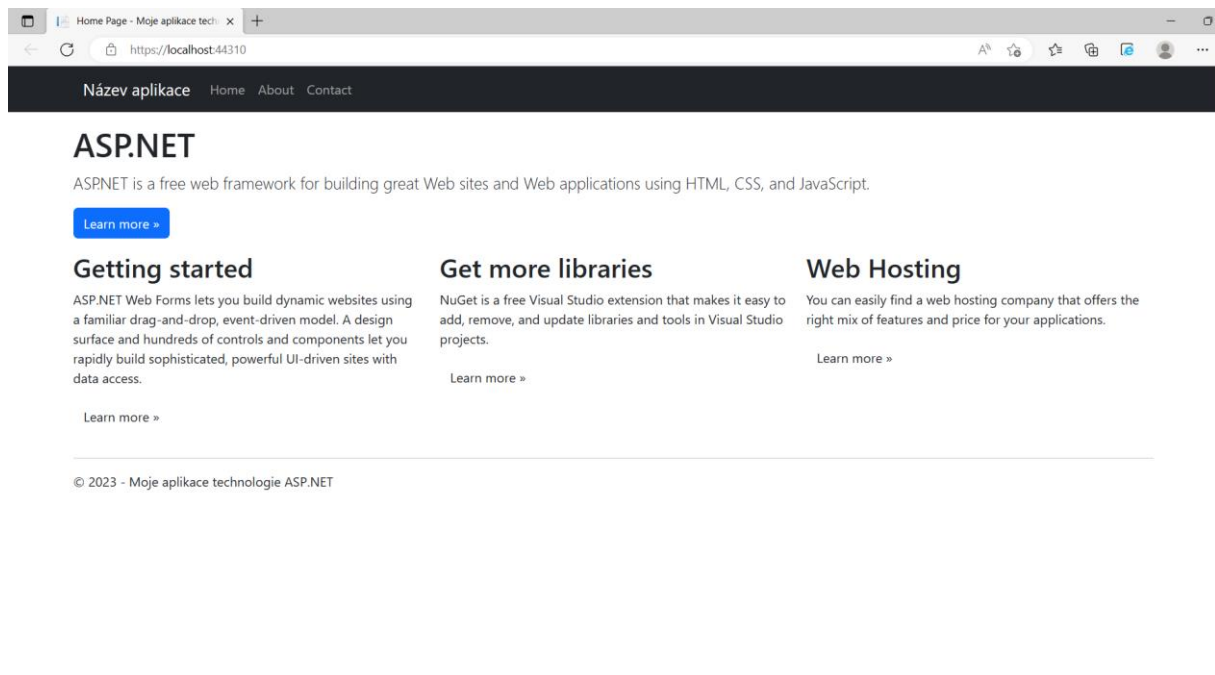
Pro náš projekt jsme po spuštění aplikace, při vytváření nového projektu, zvolili Webovou aplikaci ASP.NET (.NET Framework), která podporuje programovací jazyk C# (viz obr. 19).



Obr. 19 Vytvoření nového projektu

(Zdroj: Vlastní snímek obrazovky)

Microsoft Visual Studio se snaží svým uživatelům být co nejvíce nápomocno a po zvolení a vytvoření nového projektu za vás nachystá první webové stránky, se kterými lze dále pracovat (viz obr. 20). Takto připravené stránky ovšem pro náš projekt nebyly potřeba a využili jsme našeho HTML kódu napsaného v Brackets a kód jsme pouze zkopírovali do Visual Studia.

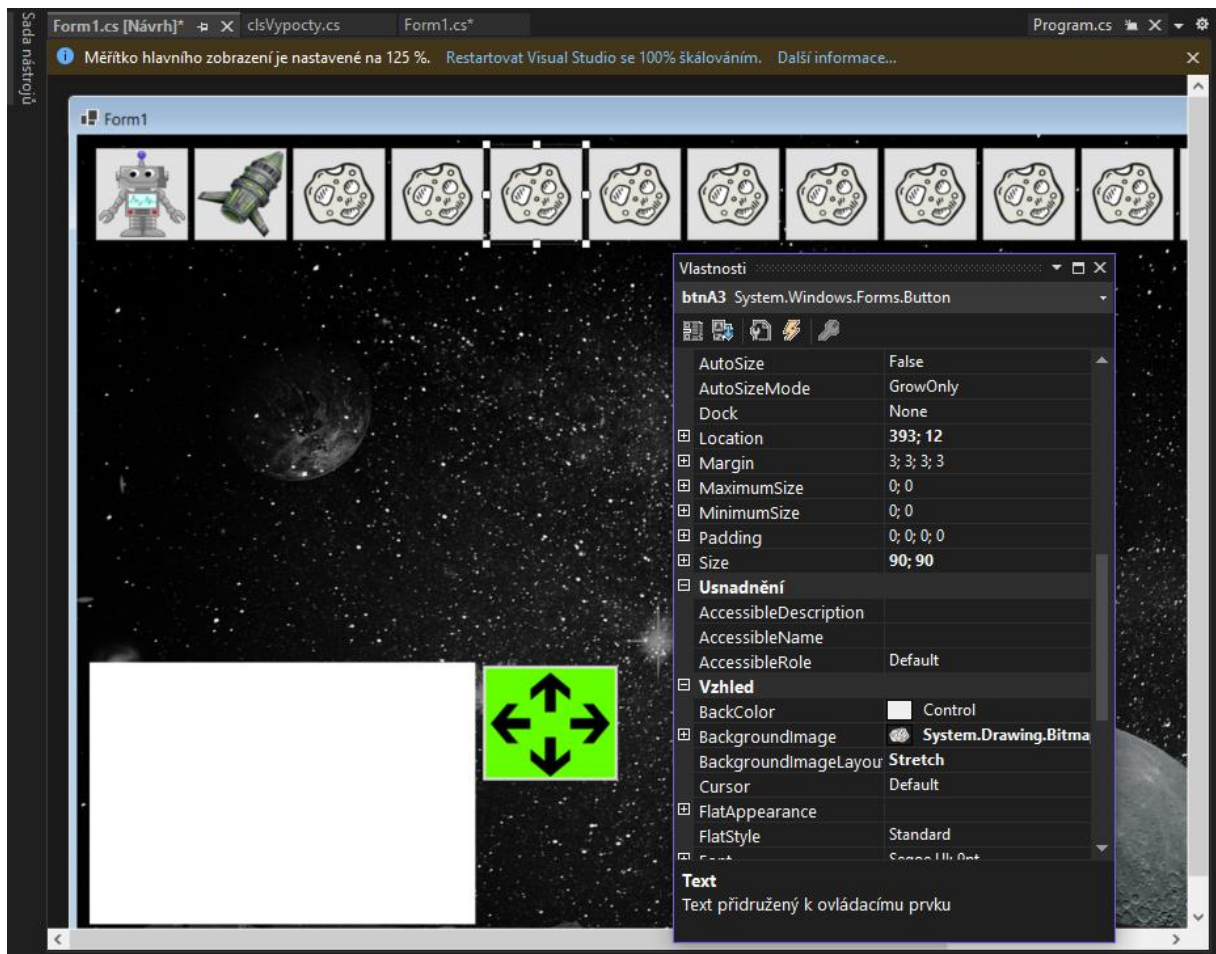


Obr. 20 Defaultní stránky projektu

(Zdroj: Vlastní snímek obrazovky)

Webové stránky byly hotové a funkční a zbývalo již jen naprogramovat pomocí webových formulářů a C# funkční aplikaci.

Soubor Form1.cs obsahuje veškeré ovládací prvky a jejich nastavení v jazyku C#. Obsah tohoto souboru je kombinací při práci v návrhu vzhledu webové aplikace a jejich doplněním. Za každý vložený prvek se v souboru objeví jeho příkaz. Dvojitým poklepáním na tento soubor se otevře soubor Form1.cs [Návrh] a je možné nastavit vzhled webové aplikace. Tento byl vytvořen pomocí formuláře a sady nástrojů, které obsahuje. Všechny prvky lze najít v nabídce sady nástrojů. Pro naše účely a pro jednoduchost jsme zvolily místo prvků tlačítka, které se pomocí kódu pohybují. Každému tlačítku lze nastavit různé vlastnosti a chování. Do vlastností byl stanoven pevný rozměr a dále pozadí, které nahradily potřebné obrázky (viz obr. 21).

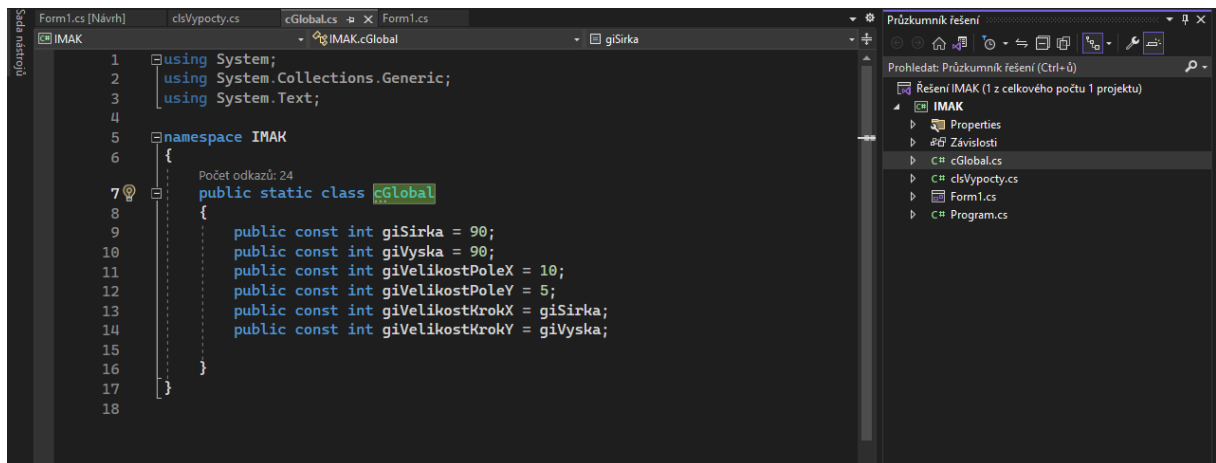


Obr. 21 Úprava vzhledu asteroidu

(Zdroj: Vlastní snímek obrazovky)

Dalšími prvky na stránkách jsou textová pole, do jednoho žák zapisuje počet kroků a druhé pole naopak vypisuje postup Imaka, který mu byl nastaven.

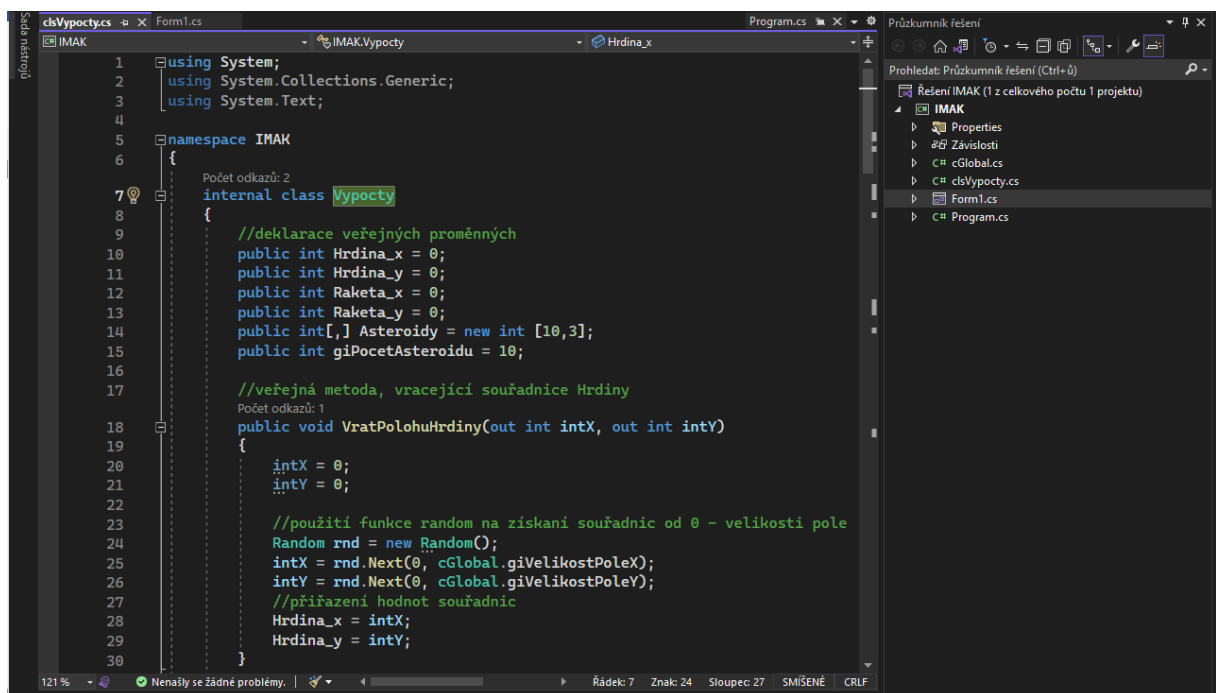
Po vzhledu aplikace přišlo na řadu samotné psaní kódů. První soubor cGlobal obsahuje nastavení veřejných konstant. Vzhledem pohybu po ose X a ose Y byly stanoveny podmínky pro tyto osy. Stanoveny byly konstanty pro šířku a výšku políček, velikost hracího pole a velikost kroků (viz obr. 22). Pomocí těchto konstant je jednoduché změnit velikost hracího pole a velikost políček.



Obr. 22 Nastavení konstant

(Zdroj: Vlastní snímek obrazovky)

Dalším nezbytným souborem pro běh aplikace je soubor clsVypocty.cs. Uvnitř tohoto souboru jsou zapsány metody a podmínky chování nastavených tlačítek. Stanovení těchto podmínek a metod je nezbytné. Probíhá hlídání umístění prvků na hracím poli, aby nedošlo při rozmístění prvků k nastavení stejné pozice. Mohlo by se totiž stát, že by se ve hře objevilo málo asteroidů, protože by se ostatní překrývaly nebo by se Imak objevil přímo v asteroidu či v raketě apod. Na začátku byly opět stanoveny proměnné, které se využívají v průběhu psaní kódu. Opět pro přehlednost byly využívány komentáře (viz obr. č 23).



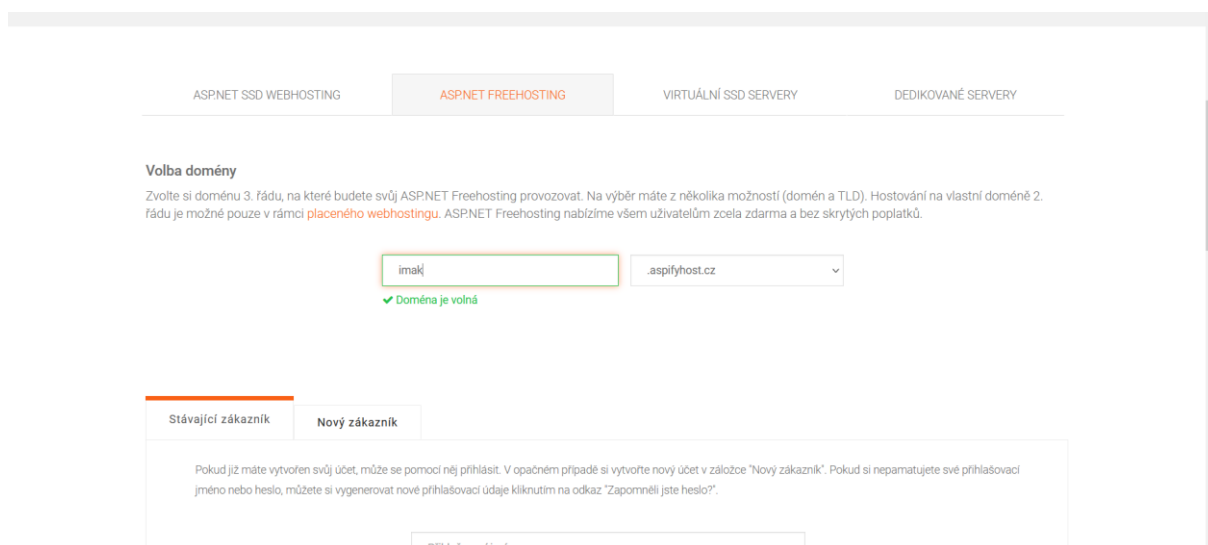
Obr. 23 clsVýpočtu

(Zdroj: Vlastní snímek obrazovky)

10.5. Volba webhostingu a publikování na internet

Při volbě webhostingu jsme museli vzít do úvahy podmínku kdy daný poskytovatel musí podporovat ASP.NET. Rozhodli jsme se pro začátek využití webhostingu zdarma. Po hledání na internetu jsme zvolili stránky od společnosti aspify. Jejich stránky jsou aspify.com, které vás po zjištění nastavení na cs jazyk přeměrují na českou verzi stránek. Tento poskytovatel webhostingu vám dá k dispozici zdarma 100 MB diskového prostoru, což pro náš projekt bohatě stačí. Dále pak 1x MS SQL databázi o velikosti 100 MB, Podporu ASP.NET 2.x-4.x i ASP.NET Core 3.x-5.x-6.x. Provoz na doméně 3.řádu, tím budou mít stránky v názvu aspifyhost.cz). Podporují nejnovější verze .NET Framework a webového serveru IIS. Mezi výhody patří běh na SSD discích od společnosti Intel, neomezené datové přenosy, v případě výpadku se společnost o všechno postará.

Pro registraci webových stránek můžete nejprve vytvořit uživatelský účet a přihlašovací údaje k administraci a publikování vašich projektů vám přijdou na uvedený email. Po přihlášení k uživatelskému účtu můžete pokračovat k vytvoření webhostingu. Tento postup lze vynechat a provést volbu domény a posléze vytvořit uživatelský účet (viz obr. 24).



ASP.NET SSD WEBHOSTING ASP.NET FREEHOSTING VIRTUÁLNÍ SSD SERVERY DEDIKOVANÉ SERVERY

Volba domény
Zvolte si doménu 3. řádu, na které budete svůj ASP.NET Freehosting provozovat. Na výběr máte z několika možností (domén a TLD). Hostování na vlastní doméně 2. řádu je možné pouze v rámci placeného webhostingu. ASP.NET Freehosting nabízíme všem uživatelům zcela zdarma a bez skrytých poplatků.

imak .aspifyhost.cz

✓ Doména je volná

Stávající zákazník Nový zákazník

Pokud již máte vytvořen svůj účet, může se pomocí něj přihlásit. V opačném případě si vytvořte nový účet v záložce "Nový zákazník". Pokud si nepamätujete své přihlašovací jméno nebo heslo, můžete si vygenerovat nové přihlašovací údaje kliknutím na odkaz "Zapomněli jste heslo?".

Přihlašovací jméno

Obr. 24 Registrace domény

(Zdroj: Vlastní snímek obrazovky)

K publikování našeho projektu jsme využili možnosti, které poskytuje Microsoft Visual studio, které umí publikovat přímo na web přes FTP protokol. Přihlašovací údaje jsme vyplnili dle pokynů, které přišly na emailovou adresu. Uživatel si musí pohlídat nastavení cesty k webu,

cesta webu musí být nastavena na www, jinak vám vaše stránky nebudou fungovat (viz obr. 25).

Publikovat

Publikovat

Připojení

Nastavení

FTPProfile

Server: free.aspify.com

Cesta k webu: www

Pasivní režim

Uživatelské jméno: imak.aspifyhost.cz

Heslo: ●●●●●●●●

Uložit heslo

Cílová adresa URL: imak.aspifyhost.cz

Ověřit připojení

< Předchozí Další > Uložit Zrušit

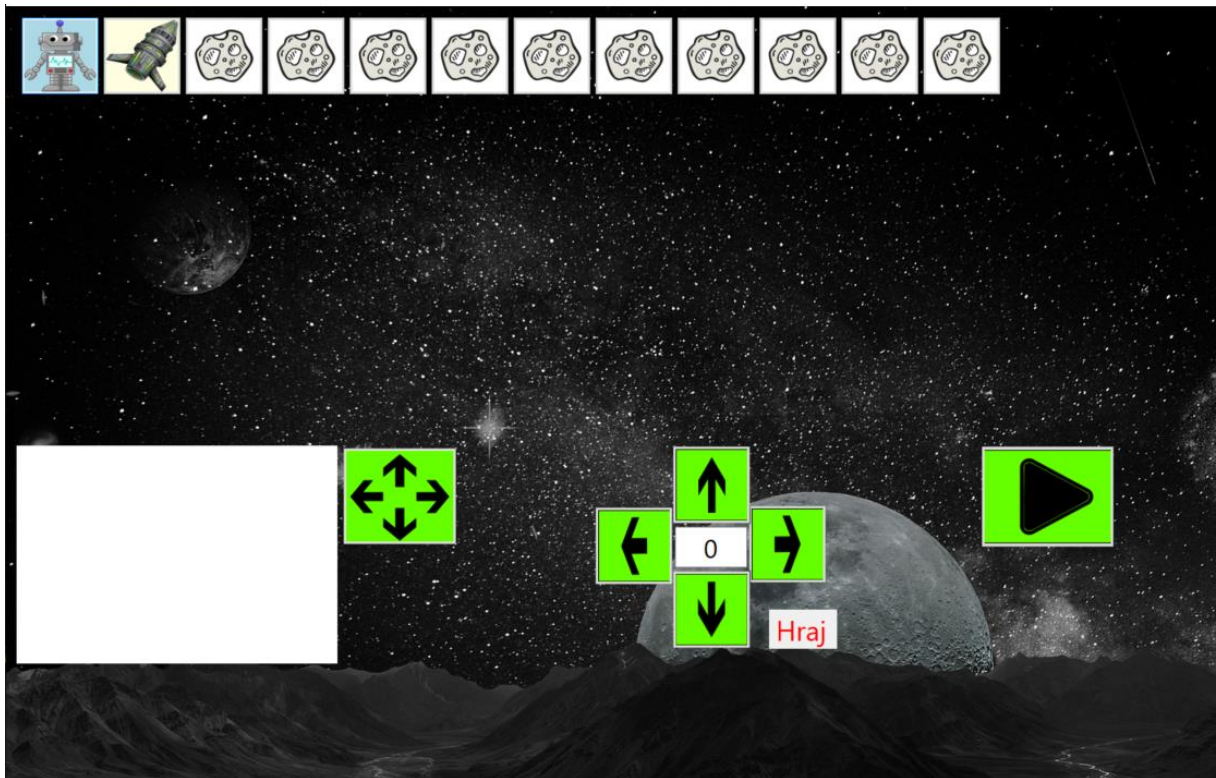
Obr. 25 Publikování projektu

(Zdroj: Vlastní snímek obrazovky)

Projekt lze publikovat i pomocí jiných nástrojů, které umí použít FTP protokol. V minulosti se nám osvědčili WINSOCP nebo také Total Commander.

10.6. Popis aplikace

Do aplikace vstupují tři hlavní prvky. Prvním z nich je Imak (Hrdina), kterému byla dána podoba robota a který se bude pohybovat dle kroků nastavených pomocí tlačítek. Druhým nezbytným prvkem je cíl, který dostal podobu vesmírné rakety a třetím prvkem jsou překážky, jež zastupují obrázky asteroidu, protože se náš hrdina pohybuje ve vesmíru (viz obr. 26).

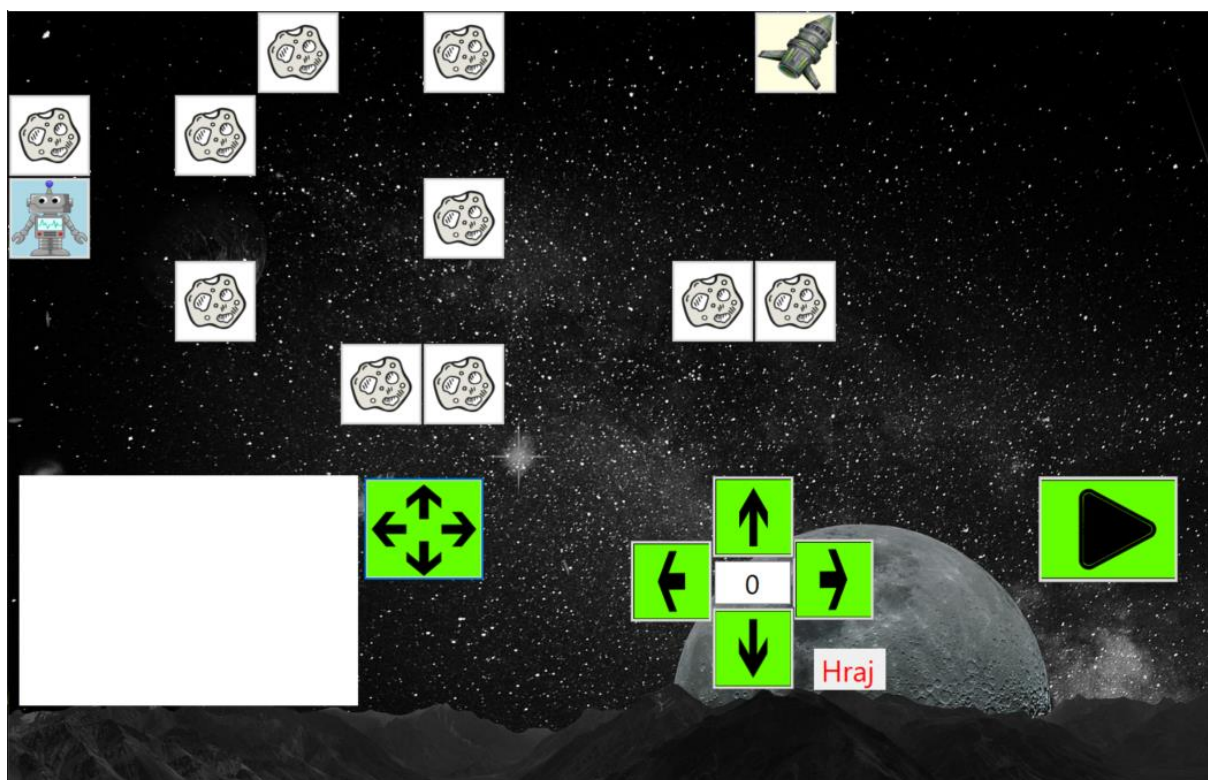


Obr. 26 Úvodní stránka aplikace

(Zdroj: Vlastní snímek obrazovky)

Na začátku musí žák provést rozmístění prvků na hracím poli aktivací tlačítka pro rozmístění prvků, která je umístěna vedle textového pole. Toto tlačítko zároveň slouží pro každé nové rozmístění prvků na obrazovce.

Překážky jsou kladeny náhodně a vždy na jiném místě. Jako optimální počet překážek (asteroidů) byl nastaven na 10. V kódu lze počet překážek lehce přenastavit, ale vzhledem k velikosti hracího pole není vhodné mít překážek méně nebo naopak více. Po rozlosování může vypadat hrací pole jako na obr. 27.



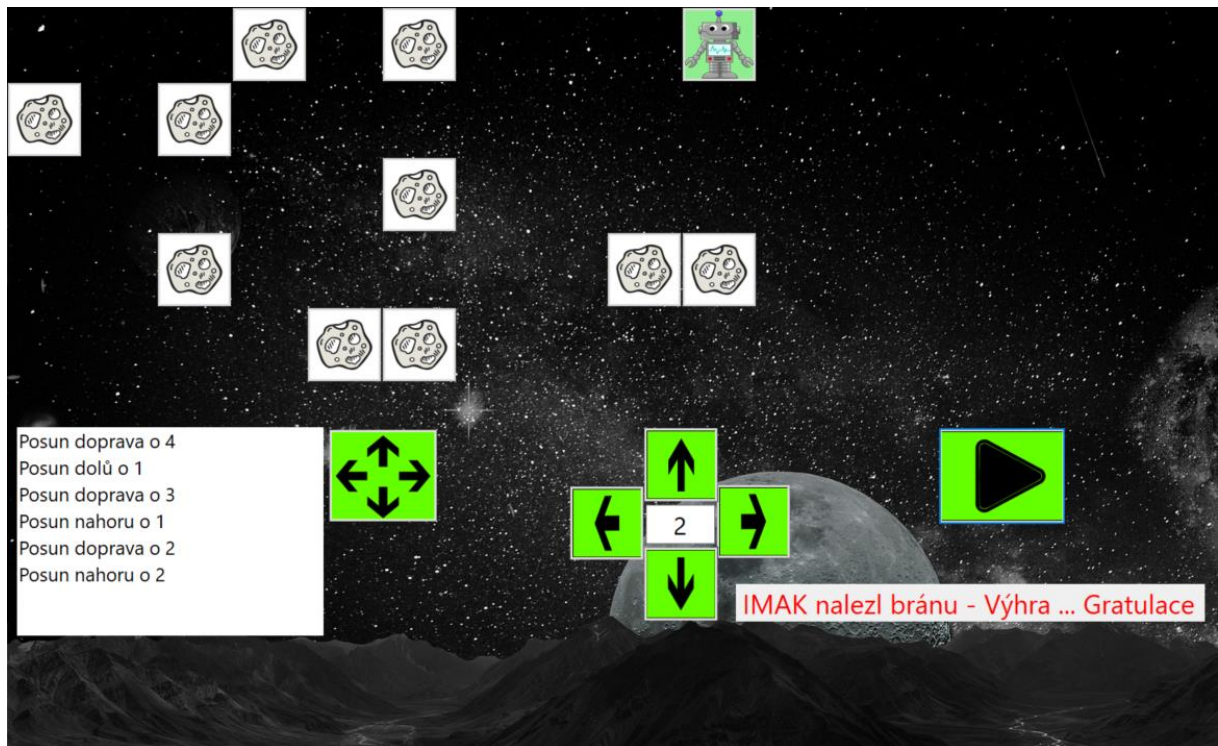
Obr. 27 Rozmístění prvků na obrazovce

(Zdroj: Vlastní snímek obrazovky)

Jak lze na obrázku vidět, žák má pro svoji cestu k cíli více možností a záleží pouze na něm jakou cestu zvolí. V podmínkách hry je napsáno, aby žák vyhledal co nejmenší počet kroků, ale pokud dorazí do cíle, aniž by narazil do asteroidu nebo do okraje hry (ztráta ve vesmíru) bude označen jako vítěz. Hra již nekontroluje počet kroků. Podmínka byla stanovena z důvodu zamyšlení se žáka nad cestou a počtem tahů potřebných pro cestu do cíle. V případě nutnosti vyhodnocení musí kroky spočítat učitel. Nám se naopak možnost více cest a kroků líbí, protože je možnost vidět, jak každý žák uvažuje jinak. Například na začátku může učitel nechat rozmístit prvky na obrazovce, kterou ukáže na dataprojektoru a děti můžou počítat, kolik kroků zbývá do cíle. Další možností je nechat žáky přemýšlet nad počtem tras, které je možné využít pro cestu. Každý jeden krok představuje čtverec o velikosti Imaka nebo asteroidů. V našem případě je nejmenší počet kroků 11, ale pro ukázkou nastavení byla nastavena cesta delší v počtu kroků 13.

Po nastavení cesty pomocí pole pohybu, kde uprostřed něj leží textové pole, které slouží k vyplnění počtů kroků a zmáčknutí šipky na kterou stranu se má Imak pohybovat dojde k zápisu pohybu do textového pole vedle šipky rozmístění prvků. Takto bude žák postupovat, až bude mít nastaven pohyb do cíle. Posledním tlačítkem play aktivuje pohyb Imaka.

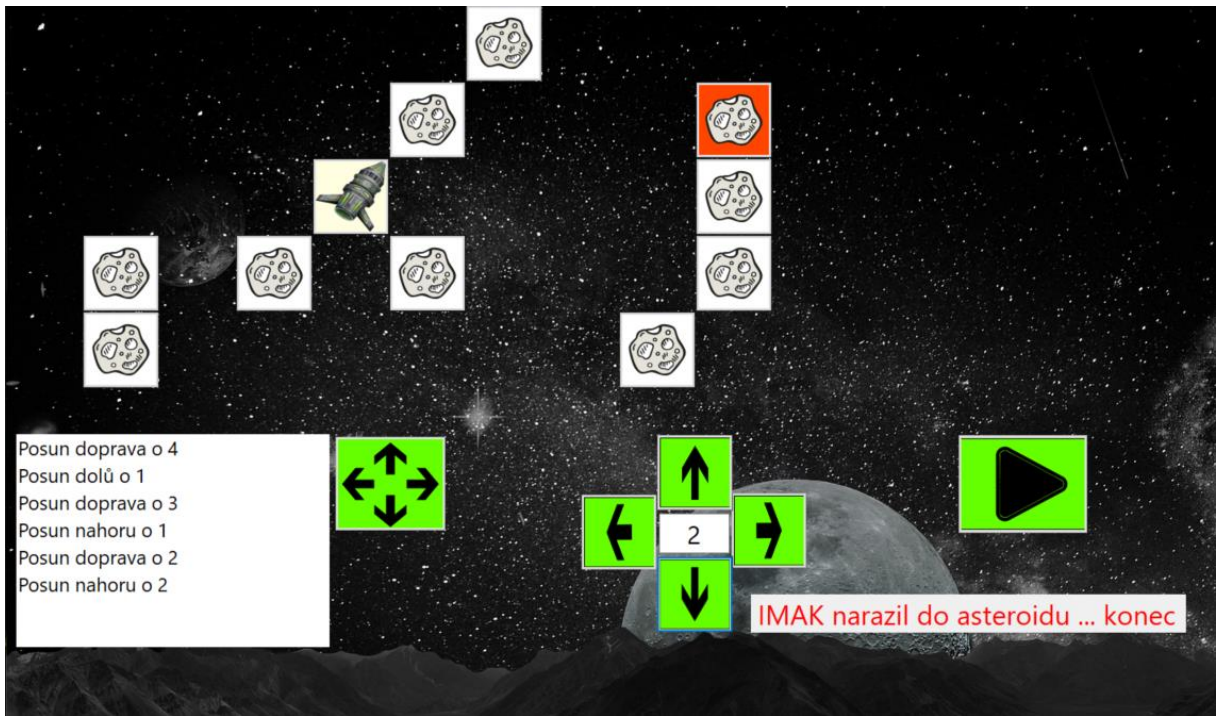
Ten se bude pohybovat po stanovené trase postupnými kroky. Jakmile dorazí do cíle, změní svoji barvu pozadí na zelenou a hra vypíše text (viz obr. 28).



Obr. 28 Nastavení cesty do cíle

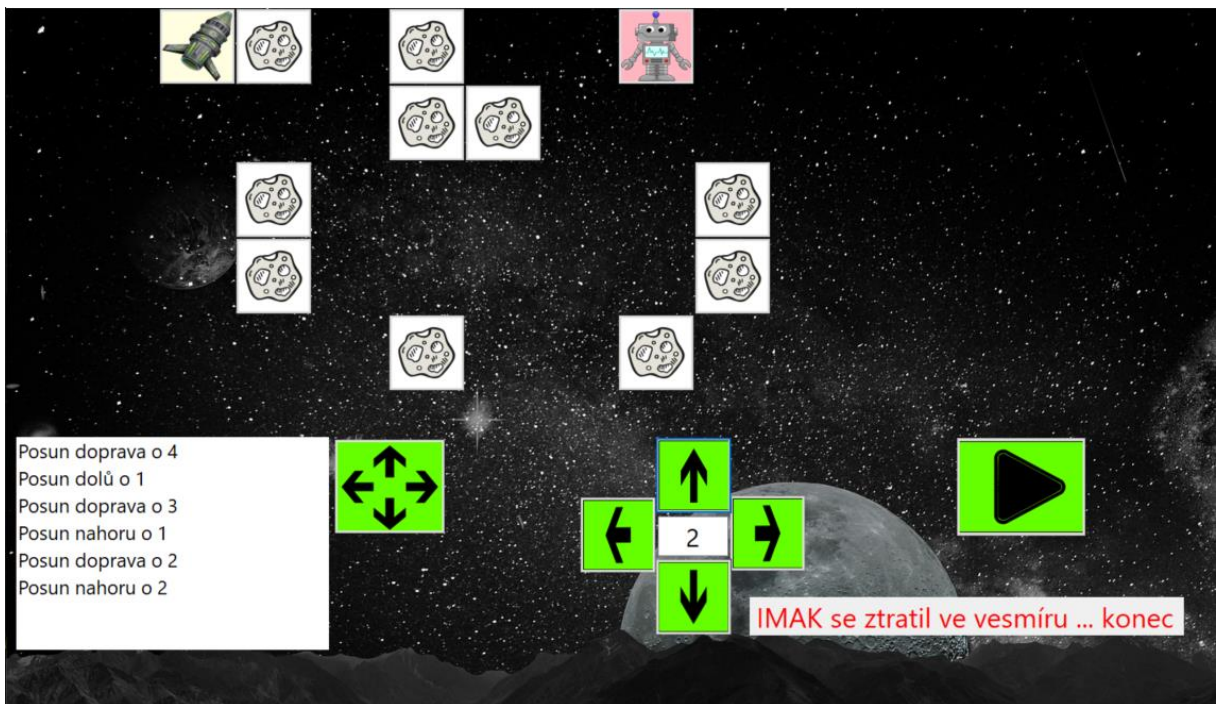
(Zdroj: Vlastní snímek obrazovky)

V případě špatného nastavení cesty může dojít k nárazu do asteroidu a hra tyto kroky vyhodnotí jako chybu. Po nárazu se asteroid zbarví do červené barvy (viz obr. 29). Další ukončení hry může nastat v případě, že Imak opustí hrací pole, pro hru bylo nastaveno pole 10 x 5 polí. Opuštění pole je znázorněno zbarvením Imaka do červena a pro žáky to představuje opuštění vesmíru (viz obr. 30).



Obr. 29 Náraz do asteroidu

(Zdroj: Vlastní snímek obrazovky)



Obr. 30 Opuštění hracího pole

(Zdroj: Vlastní snímek obrazovky)

Závěr

Cílem diplomové práce bylo vytvořit interaktivní webovou aplikaci pro podporu výuky informatických předmětů na 2. stupni ZŠ. Aplikace je v současné době funkční a plně k dispozici učitelům informatiky čímž byl stanovený cíl diplomové práce splněn.

V teoretické části byla zpracována problematika Strategie 2030+, dále informace o digitálním vzdělávání na základních školách, a také byla provedena rešerše na téma informatiky a informatického myšlení a jak učit programování. Nedílnou součástí teoretické části je popis programovacích jazyků HTML, CSS a C#, a také vývojového studia Microsoft Visual Studio 2022.

V praktické části je popsán vývoj interaktivní webové aplikace IMAK, která byla vyvinuta v Microsoft Visual Studiu 2022 a v programovacím jazyku C# s využitím HTML a CSS. V této části je popsán postup při zřizování webhostingu a domény. Dále popis vlastní interaktivní webové aplikace IMAK, která je určena pro použití jak v internetovém prohlížeči na osobním počítači, tak také přes mobilní telefon.

Tato diplomová práce je malým příspěvkem k celkové problematice vývoje interaktivních online pomůcek pro výuku, kdy během celosvětové pandemie koronaviru bylo nutné studovat online formou. S rozvojem informačních technologií jde ruku v ruce rozvoj aplikací, které lze na internetu nalézt a využít ve vzdělávání.

Seznam použitých zkratk

CSS – Cascading style sheets (kaskádové styly)

HTML – hyper text markup language (hypertextový značkovací jazyk)

MŠMT – ministerstvo školství, mládeže a tělovýchovy

NÚV – národní ústav vzdělávání

RVP PV – rámcový vzdělávací program pro předškolní vzdělávání

RVP ZV – rámcový vzdělávací program pro základní vzdělávání

RVP ZUV – rámcový vzdělávací program pro základní umělecké vzdělávání

RVP G* – rámcový vzdělávací program pro gymnázia

RVP SOV – rámcový vzdělávací programy středního odborného vzdělávání

RVP SPV – rámcový vzdělávací program pro speciální vzdělávání

RVP ICT – rámcový vzdělávací programy v oblasti informatiky a informačních a komunikačních technologií

ŠVP – školní vzdělávací program

Seznam použitých zdrojů

Literatura

BUČKOVÁ, Hana a Jiří DOSTÁL, 2020. *Kurikulum informatiky a digitálních technologií z pohledu učitelů 2. stupně základních škol*. Olomouc: Univerzita Palackého v Olomouci, Pedagogická fakulta. ISBN: 978-80-244-5904-2 (online: PDF – Bookport)

ČADA, Ondřej, 2009. *Objektové programování: naučte se pravidla objektového myšlení*. Praha: Grada. Průvodce (Grada). ISBN 978-80-247-2745-5

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi*. 3., aktualizované vydání. Praha: Grada Publishing, 2015. *Management v informační společnosti*. ISBN 978-80-247-9919-3 (ePub)

HORŇÁKOVÁ, Michaela, 2011. *Copywriting: podrobný průvodce tvorbou textů, které prodávají*. Brno: Computer Press, ISBN 978-80-251-3269-2

LAURENČÍK, Marek, 2019. *Tvorba www stránek v HTML a CSS*. Praha: Grada Publishing. Průvodce. ISBN 978-80-271-2241-7

LALÍK, Michal, 2013. *WWW pro každého*. Praha: Grada Publishing. Průvodce. ISBN 978-80-247-4528-2

PILGRIM, Mark, 2014. *Ponořme se do HTML5*. Praha: CZ.NIC, z.s.p.o. . ISBN 978-80-88168-07-2

PÍSEK, Slavoj, 2014. *HTML začínáme programovat – 4., aktualizované vydání*. Praha: Grada Publishing. ISBN 978-80-247-5059-0

ŠÍMA, František a David VILÍMEK, 2006. *Microsoft Visual Studio .NET: praktické programování krok za krokem*. Praha: Grada. Průvodce (Grada). ISBN 80-247-1418-3.

VIRIUS, Miroslav, 2021. *Programování v C#: od základů k profesionálnímu použití*. Praha: Grada Publishing. Myslíme v. ISBN 978-80-271-1216-6

ZORMANOVÁ, Lucie, 2014. *Obecná didaktika: pro studium a praxi*. Praha: Grada. Pedagogika (Grada). ISBN: 978-80-247-9132-6 (ePub)

Elektronické zdroje

Adobe.com. Informace o webových aplikacích. *Adobe.cz* [online] 19. 5. 2021 [cit. 3. 6. 2022]. Copyright © 2022 Adobe. All rights reserved. Dostupné z: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>

BARÁŠEK, Jan. Princip zapouzdření v OOP. In: *Php.baraja.cz* [online]. © 2009-2023 Jan Barášek, 16. 2. 2020, [cit. 15. 4. 2023]. Dostupné z: <https://php.baraja.cz/zapouzdeni#interni-data-ponechte-neverejna>

BĚLOHLÁVEK, R. (2016) „Informatika jako obor“, *Matematika–Fyzika–Informatika*, 25(4) s. 299–315. [cit. 1. 4. 2022]. Dostupné z: <https://www.mfi.upol.cz/index.php/mfi/article/view/287>

BuiltWith PTY Ltd. *ASP.NET USAGE Statistics*. Builtwith.com [online] [cit. 23. 3. 2023]. Dostupné z: <https://trends.builtwith.com/framework/ASP.NET>

ČÁPKA, D. 1. díl – Úvod do PHP a webových aplikací. *Itnetwork.cz* [online]. [cit. 2. 6. 2022]. Dostupné z: <https://www.itnetwork.cz/php/zaklady/php-tutorial-uvod-do-webovych-aplikaci/>

ČÁPKA, M. Lekce 1 - Úvod do ASP.NET. *Itnetwork.cz* [online]. [cit. 5. 4. 2023]. Dostupné z: https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net?gclid=CjwKCAjw_MqgBhAGEiwAnYOAejuDzOcONwNhSBA6Qz3wm7067j7G5EC2j_hXDDsFf_vGA3PQsbbNSaxoCD8wQAvD_BwE

Dotnet.microsoft.com. *ASP.NET* [online] [cit. 16. 3. 2023]. © Microsoft 2023. Dostupné z: <https://dotnet.microsoft.com/en-us/apps/aspnet>

Dotnet.microsoft.com. *What is Aspnet* [online] [cit. 16. 3. 2023]. © Microsoft 2023. Dostupné z: <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>

EDU.CZ, *Ministerstvo školství, mládeže a tělovýchovy*. 2020 [online] [cit. 1. 4. 2022]. Dostupné z: <https://www.edu.cz/rvp-ramcove-vzdelavaci-programy/>

FIALA, Tomáš, 5 pravidel, jak vybrat název domény. *Nastartujtese.cz* [online]. Komerční banka, a. s., © 2020, 18. 9. 2017 [cit. 2. 6. 2022]. Dostupné z: <https://www.nastartujtese.cz/blog/rady-a-tipy/5-pravidel-jak-vybrat-nazev-domeny>

FRYČ, Jindřich, Zuzana MATUŠKOVÁ, Pavla KATZOVÁ, et al., 2020. *Strategie vzdělávací politiky České republiky do roku 2030+*. Praha: Ministerstvo školství, mládeže a tělovýchovy. ISBN 978-80-87601-47-1. (online brožura, pdf) Dostupné z: https://www.msmt.cz/uploads/Brozura_S2030_online_CZ.pdf

FURMAN, Slavomír. Web formuláře v ASP.NET. In: *interval.cz* [online]. ©ZONER, a.s. 18. 4. 2002 [cit. 31. 3. 2023]. Dostupné z: <https://www.interval.cz/clanky/web-formulare-v-aspnet/>

GitHub. Brackets-cont/brackets. In: *GitHub.com* [online]. © 2023 GitHub, Inc., [cit. 5. 4. 2023]. Dostupné z: <https://github.com/brackets-cont/brackets/>

imysleni.cz, Co je infromatické myšlení, 2018 [online], Jihočeská univerzita v Českých Budějovicích, [cit. 1. 4. 2022]. Dostupné z: <https://imysleni.cz/informaticke-mysleni/co-je-informaticke-mysleni>

imysleni.cz, Infromatické myšlení, 2018 [online]. Jihočeská univerzita v Českých Budějovicích, [cit. 12. 4. 2022]. Dostupné z: <https://imysleni.cz/>

imysleni.cz, Modelový ŠVP: Kreativně vpřed, 2018 [online], Jihočeská univerzita v Českých Budějovicích, [cit. 1. 4. 2022]. Dostupné z: https://imysleni.cz/images/SVP/SVP4_kreativne-vpred_v20201229.pdf

imysleni.cz, Modelový ŠVP: Nebojácně vpřed, 2018 [online], Jihočeská univerzita v Českých Budějovicích, [cit. 1. 4. 2022]. Dostupné z: https://imysleni.cz/images/SVP/SVP2_nebojacne-vpred_2-4_v20201229.pdf

imysleni.cz, Modelový ŠVP: Opatrně vpřed, 2018 [online], Jihočeská univerzita v Českých Budějovicích, [cit. 1. 4. 2022]. Dostupné z: https://imysleni.cz/images/SVP/SVP1_opatrne-vpred_2-4_v20210113.pdf

imysleni.cz, Modelový ŠVP: Progresivně vpřed, 2018 [online], Jihočeská univerzita v Českých Budějovicích, [cit. 1. 4. 2022]. Dostupné z: https://imysleni.cz/images/SVP/SVP3_progresivne-vpred_3-5_v20201229.pdf

Inkscape, About. [online] [cit. 5. 4. 2023] Dostupné z: <https://inkscape.org/about/>

JANÁČEK, Ondřej. C# Internals – Method Overloading, In: *dotnetportal.cz*, [online] 9. 3. 2014. © 2023 Tomáš Herceg, Tomáš Jecha, [cit. 5. 4. 2023]. Dostupné z: <https://www.dotnetportal.cz/blogy/12/Ondrej-Janacek/5372/C-Internals-Method-Overloading>

Katedra výpočetní a didaktické techniky, 2021. Projekt PRIM, KVD [online]. [cit. 12. 4. 2022]. Dostupné z: <https://www.kvd.zcu.cz/cz/projekty/prim.php>

LESSNER, Daniel. Jak si přeložíme „computational thinking“? 2014 [online]. [13. 4. 2023]. Dostupné z: https://ksvi.mff.cuni.cz/~lessner/w/data/_uploaded/file/papers/2014_02_lessner_didactig.pdf

MŠMT.CZ, *Opatření-PM-RVP-ZV-2021-komplet-web-informatika.pdf*, 2013-2022, [cit. 1. 4. 2022]. Dostupné z: https://www.msmt.cz/file/56005_1_1/download/

Národní ústav pro vzdělávání, Podpora rozvíjení infromatického myšlení (PRIM). *Nuv.cz* [online]. ©2011-2022, [cit. 12. 4. 2022]. Dostupné z: <https://www.nuv.cz/projekty/prim>

Rámcový vzdělávací program pro základní vzdělávání [online]. Praha: MŠMT, 2021 [cit. 1. 4. 2022]. Dostupné z: https://www.nuv.cz/file/4983_1_1/download/

Ministerstvo školství, mládeže a tělovýchovy a Národní pedagogický institut České republiky., 2022 [online]. MŠMT ČR & NPI ČR [cit. 1. 4. 2022]. Dostupné z: <https://revize.edu.cz/co-se-meni>

KAŽOUSKOVÁ, B. Web, webová stránka a webová aplikace, v čem je rozdíl? *Rascasone.com* [online]. © 2022 Rascasone, s.r.o., 22. 6. 2021 [cit. 3. 6. 2022]. Dostupné z: <https://www.rascasone.com/cs/blog/web-webova-aplikace-rozdil>

KAŽOUSKOVÁ, B. Typy webových aplikací, kterou zvolit pro vlastní vývoj? *Rascasone.com* [online]. © 2022 Rascasone, s.r.o., 12. 12. 2021 [cit. 5. 6. 2022]. Dostupné z: <https://www.rascasone.com/cs/blog/typy-webovych-aplikaci>

Microsoft. Takto vytvoříme software. In: *Visualstudio.microsoft.com* [online]. © Microsoft 2023, [cit. 5. 4. 2023]. Dostupné z: <https://visualstudio.microsoft.com/cs/>

PERZINA, Radomír. *Objektové programování*. Karviná, 2021: Slezská univerzita v Opavě, Obchodně podnikatelská fakulta v Karviné. [cit. 5. 4. 2023]. Dostupné z: https://is.slu.cz/el/opf/zima2021/INMBKOBP/2972352/oop_Opora.pdf

Rámcový vzdělávací program pro základní vzdělávání [online]. Praha: MŠMT, 2021 [cit. 1. 4. 2022]. Dostupné z: https://www.nuv.cz/file/4983_1_1/download/

ROSENCRANCE, Linda. *Internet Information Services (IIS)* In: Techtarget.com [online]. TechTarget, ©2000–2023, Září 2019 [cit. 27. 3. 2023]. Dostupné z: <https://www.techtarget.com/searchwindowserver/definition/IIS>

ŠETEK, David. 81. Tvorba webových stránek (HTML a CSS) - Grid: dva sloupce vedle sebe, tři sloupce atd. In: *www.youtube.com* [online]. 17. 7. 2020 [8. 4. 2023]. Dostupné z: https://www.youtube.com/watch?v=h2e-qmuQaQc&list=PLQ8x_VWW6AkVciDzMEI5K9jW_1rsV9PTf&index=83

ŠETEK, David. 86. Tvorba webových stránek (HTML a CSS) - Sekce s referencemi od zákazníků v html a css. In: *www.youtube.com* [online]. 27. 7. 2020 [8. 4. 2023]. Dostupné z: https://www.youtube.com/watch?v=jacx_77JGQA&list=PLQ8x_VWW6AkVciDzMEI5K9jW_1rsV9PTf&index=89

ŠETEK, David. 99. Tvorba webových stránek (HTML a CSS) – Úvod do responzivity webu. In: *www.youtube.com* [online]. 4. 8. 2020 [8. 4. 2023]. Dostupné z: https://www.youtube.com/watch?v=eB6mhRg4w1g&list=PLQ8x_VWW6AkVciDzMEI5K9jW_1rsV9PTf&index=101

ŠTRÁFELDA, Jan. Kaskádové styly (CSS). *Starfelda.cz* [online]. [cit. 7. 4. 2023]. Dostupné z: <https://www.strafelda.cz/kaskadove-styly>

TRIANGLE717. le717/Brackets-html-skeleton. In: *Github.com* [online]. © 2023 GitHub, Inc., 10. 8. 2015 [cit. 7. 4. 2023]. Dostupné z: <https://github.com/le717/brackets-html-skeleton>

VANÍČEK, Jiří, NAGYOVÁ, Ingrid, TOMCSÁNYIOVÁ, Monika. *Programování ve Scratch pro 2. stupeň základní školy* [online]. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2020. Dostupné z: <https://imysleni.cz/ucebnice/programovani-ve-scratchi-pro-2-stupen-zakladni-skoly>

Visual studio. Produktová řada Visual Studio 2022 – požadavky na systém. In: *Learn.Microsoft.com* [online]. © Microsoft 2023, 8. 3. 2023 [cit. 31. 3. 2023]. Dostupné z: <https://learn.microsoft.com/cs-cz/visualstudio/releases/2022/system-requirements>

Visual studio. Visual Studio 2022. [online]. © Microsoft 2023, [cit. 31. 3. 2023]. Dostupné z: <https://visualstudio.microsoft.com/cs/>

WING, Jeannette M. Computational Thinking: What and Why? [online]. 2010 [cit. 13. 4. 2023]. Dostupné z: <https://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf>

w3schools.com *ASP.NET Web Forms – Tutorial* [online]. [cit. 31. 3. 2023]. Dostupné z: https://www.quanzhanketang.com/aspnet/aspnet_intro.html

w3School.com. *C#Introduction*. W3School.com [online] Copyright 1999-2023 by Refsnes Data. All Rights Reserved. W3Schools is Powered by W3.CSS [cit. 15. 3. 2023]. Dostupné z: https://www.w3schools.com/cs/cs_intro.php

ŽŮREK, D. Lekce 1 – Visual Studio – Úvod do vývojového prostředí. *Itnetwork.cz* [online]. [cit. 5. 4. 2023]. Dostupné z: <https://www.itnetwork.cz/csharp/visual-studio/tutorial-visual-studio-uvod>

Kvalifikační práce

NOVÁK, Vlastislav. Tvorba webové aplikace pro podporu rozvoje inženýrského myšlení. Olomouc, 2021. Diplomová práce. Univerzita Palackého, Pedagogická fakulta, Katedra techniky a informatiky. Vedoucí práce Mgr. Tomáš Dragon.

Seznam obrázků

Obr. 1 Rozcestník učebnic.....	25
Obr. 2 Vývojový diagram přihlášení k webové aplikaci	27
Obr. 3 Strukturovaný diagram	28
Obr. 4 Zpracování statické webové stránky	34
Obr. 5 Zpracování dynamických stránek.....	35
Obr. 6 Vnořování elementů	40
Obr. 7 Responzivita	44
Obr. 8 Vložený skeleton HTML.....	46
Obr. 9 Ukázka základních tagů.....	46
Obr. 10 Dynamický web ASP.NET.....	50
Obr. 11 Sada nástrojů	51
Obr. 12 Logo aplikace Imak	54
Obr. 13 Hlavička stránek IMAK	55
Obr. 14 Ukázka HTML kódu	56
Obr. 15 Prvotní spuštění	57
Obr. 16 Třídy a rozložení pro CSS	58
Obr. 17 Zápis CSS	59
Obr. 18 Nastavení responzivity	60
Obr. 19 Vytvoření nového projektu.....	61
Obr. 20 Defaultní stránky projektu.....	62
Obr. 21 Úprava vzhledu asteroidu.....	63
Obr. 22 Nastavení konstant	64
Obr. 23 clsVýpočtu.....	64
Obr. 24 Registrace domény	65
Obr. 25 Publikování projektu	66
Obr. 26 Úvodní stránka aplikace	67
Obr. 27 Rozmístění prvků na obrazovce	68
Obr. 28 Nastavení cesty do cíle.....	69
Obr. 29 Náráz do asteroidu.....	70
Obr. 30 Opuštění hracího pole.....	70

Seznam grafů

Graf 1 Technologie pro tvorbu webových stránek	49
--	----

Seznam tabulek

Tab. 1: Učební plán – Opatrně vpřed	18
Tab. 2: Učební plán – Nebojácně vpřed	19
Tab. 3: Učební plán – Progresivně vpřed	20
Tab. 4: Učební plán – Kreativně vpřed, 1. stupeň	22
Tab. 5: Učební plán – Kreativně vpřed, 2. stupeň	23

ANOTACE

Jméno a příjmení:	Bc. Libor Jurča
Katedra:	Katedra technické a informační výchovy
Vedoucí práce:	Mgr. Tomáš Dragon
Rok obhajoby:	2023

Název práce:	Vývoj interaktivní webové aplikace pro podporu výuky informatických předmětů na 2. stupni ZŠ
Název v angličtině:	The Development of an interactive web application to support the teaching of computer science subjects at lower secondary school.
Anotace práce:	Práce se zabývá tvorbou výukové webové aplikace pro podporu výuky informatických předmětů na 2. stupni ZŠ. Úkolem této aplikace je rozvoj informatického myšlení a osvojení si základních kroků při výuce algoritmizace a programování. Teoretická část se zabývá odůvodněním změn výuky a ukazuje možnost využití připravených materiálů k novému pojetí výuky informatiky na základních školách. V praktické části je soustředěna na samotný vývoj webové aplikace pomocí Visual studio a dalších webových technologií.
Klíčová slova:	Strategie 2030+, Informatické myšlení, Algoritmizace, Programování, Webová aplikace, Výuka, C#, Rámcový vzdělávací program.
Anotace v angličtině:	The diploma thesis deals with the creation of an educational web application for supporting a teaching IT subjects at the second grade of the basic school. The aim of an application is to develop the IT thinking and to master basic steps in a process of the teaching an algorithmization and a programming. The theoretical part is focused on the justification of changes in a teaching and it proves the possibilities of using prepared materials to the new concept of the IT teaching methods at the basic schools. The practical part is concentrated on the development of the web application itself via the usage of Visual studion and the other web technologies.
Klíčová slova v angličtině:	Strategy 2030+, Informatics thinking, Algorithmization, Programming, Web application, Teaching, C#, Framework educational program.
Přílohy vázané v práci:	žádné
Rozsah práce:	82
Jazyk práce:	čeština