UNIVERZITA PALACKÉHO V OLOMOUCI PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

Karolína Šmatlová

II. ročník – kombinované studium

Obor: Učitelství českého jazyka pro 2. stupeň základních škol a učitelství technické a informační výchovy pro střední školy a 2. stupeň základních škol

Tvorba multimediálního výukového softwaru pro výuku českého jazyka a literární výchovy Diplomová práce

Vedoucí práce: PhDr. Jan Lavrinčík, DiS., Ph.D.

OLOMOUC 2017

Čestné prohlášení

Prohlašuji, že jsem práci vypracovala samostatně a použila jen uvedených pramenů.

V Olomouci dne 21. 6. 2017

.....

Bc. Karolína Šmatlová

Poděkování

Poděkovat bych chtěla hlavně PhDr. Jan Lavrinčík, DiS., Ph.D. za odborné rady a trpělivý přístup, hlavně při dokončování diplomové práce.

Obsah

Ú	vod		7
	Te	oretická část	3
1	Vý	vukový software)
	1.1	Pojem výukový software)
	1.2	Didaktické funkce výukového softwaru10)
	1.3	Členění výukového softwaru11	ł
	1.4	Výhody a nevýhody výukového software13	3
	1.5	Dílčí závěr kapitoly výukový software14	1
2	Vý	vukový software pro výuku českého jazyka a literární výchovy15	5
	2.1	Výukový software od firmy SILCOM15	5
	2.2	Výukový software od firmy TERASOFT 19)
	2.3	Výukový software od nakladatelství Tobiáš22	2
	2.4	Diktátor	3
	2.5	Výukový software určený pro tablety a mobilní telefony23	3
	2.6	Dílčí závěr kapitoly výukový software pro výuku českého jazyka a literárn	í
	vých	ovy25	5
3	Pro	ogramovací jazyky	7
	3.1	C/C++	7
	3.2	Python	3
	3.3	Pascal	3
	3.4	Delphi)
	3.5	Java29)
	3.6	JavaScript)
	3.7	РНР)
	3.8	Dílčí závěr kapitoly programovací jazyky	l
4	Vi	sual Basic	2
	4.1	Historie Visual Basic	2

	4.2	Ovládací prvky Visual Basic 2010	34
	4.3	Procedury Visual Basic 2010	34
	4.4	Proměnné Visual Basic 2010	36
	4.5	Příkazy Visual Basic 2010	38
	Pra	aktická část	44
5	Uk	tázka číslo 1 (Hádanka)	45
	5.1	Výukový software – Úvod	45
	5.2	Popis cvičení ve výukovém softwaru	45
	5.3	Využití ve výuce českého jazyka a literární výchovy (literární výchova)	46
	5.4	Visual Basic – grafická úprava	47
	5.5	Visual Basic – zdrojový kód	49
6	Uk	tázka číslo 2 (Skládání textů)	52
	6.1	Popis cvičení ve výukovém softwaru	52
	6.2	Využití ve výuce českého jazyka a literární výchovy (literární výchova)	52
	6.3	Visual Basic – grafická úprava	53
	6.4	Visual Basic – zdrojový kód	54
7	Uk	tázka číslo 3 (Poslech textu)	58
	7.1	Popis cvičení ve výukovém softwaru	58
	7.2	Využití ve výuce českého jazyka a literární výchovy (literární výchova)	58
	7.3	Visual Basic – grafická úprava	58
	7.4	Visual Basic – zdrojový kód	59
8	Uk	tázka číslo 4	61
	8.1	Popis cvičení ve výukovém softwaru	61
	8.2	Využití ve výuce českého jazyka a literární výchovy (mluvnice)	61
	8.3	Visual Basic – grafická úprava	62
	8.4	Visual Basic – zdrojový kód	62
9	Hl	avní nabídka výukového softwaru	65

9.1	Program – grafická úprava	. 65		
9.2	Program – zdrojový kód	. 66		
Závěr		. 68		
Literatura				
nternetové zdroje				

Úvod

Diplomová práce pojednává o možnostech výukového softwaru primárně zaměřených na výuku českého jazyka a literární výchovy a dále o možnostech vlastní tvorby výukového softwaru.

Informační a komunikační technologie se během let staly nedílnou součástí lidských životů. Informační a komunikační technologie se postupem času zařadili do školního vzdělávacího systému. Nejprve se využívali převážně ve výuce informační výchovy, ale postupem času pronikly i do výuky ostatních předmětů (např. český jazyk a literární výchova, matematika, fyzika, dějepis,...). Jednou z oblastí informačních a komunikačních technologií, která se začala využívat při výuce ve školách, jsou výukové softwary.

Na českém trhu dnes existuje spousta výukových softwaru, které můžeme využívat při výuce ve školách. Položme si, ale otázku: Co dělat pokud nám žádný výukový software při naší výuce nevyhovuje, ale chceme jej přesto využívat? V tomto případě si můžeme výukový software vytvořit sami. Zaměřme se tedy v diplomové práci na tvorbu výukového softwaru, který může splňovat kritéria, která při výuce preferujeme.

Hlavním cílem diplomové práce je vytvořit vlastní ukázky výukového software a popsat možnosti přístupu k tvorbě výukového softwaru v aplikaci Visual Basic, který bude určený pro výuku českého jazyka a literární výchovy. Dílčím cílem práce je analyzovat dostupný výukový software pro český jazyk a literární výchovu. Zaměřit se na využití ve výuce českého jazyka a literární výchovy. Představit si možnosti programovacích jazyků a analyzovat strukturu programovacího jazyka Visual Basic.

V teoretické části diplomové práce se budeme zabývat nejprve výukovým softwarem. Vymezíme si pojem výukový software, jeho funkce a členění. Zaměříme se na výhody a nevýhody, které výukový software s sebou přináší. Dále se zaměříme na konkrétní výukový software určený pro výuku českého jazyka a literární výchovy, který nalezneme na českém trhu. Představíme si programovací jazyky, s jejichž pomocí můžeme vytvořit výukový software. Konkrétně se budeme zabývat programovacím jazykem Visual Basic a jeho strukturou.

V praktické části diplomové práce si představíme možnosti vytvoření výukového softwaru do výuky českého jazyka a literární výchovy. Popíšeme si konkrétní příklady tvorby výukového softwaru.

7

Teoretická část

1 Výukový software

V kapitole výukový software se zaměříme na vymezení pojmu výukový software, dále funkce a členění výukového softwaru. Projdeme si výhody a nevýhody, které s sebou přináší výukový software.

Využití počítače pro vzdělávání si můžeme rozdělit do dvou kategorií výuka o počítači a výuka s počítači. Výuka o počítači se zabývá hardwarovým a softwarovým vybavením počítače a následnými principy práce s tímto vybavením. Výuka o počítači se převážně vyučuje v informační výchově. Princip výuky s počítači spočívá ve využití počítače ve vzdělávání. Počítač tedy slouží jako pomůcka pro učitele a žáky a lze jej zapojit do různých předmětů (český jazyk, fyzika, matematika,...).

Výukovým softwarem se budeme zabývat téměř ve všech částech diplomové práce.

1.1 Pojem výukový software

Vymezení pojmu výukový software není jednoznačné. Řada autorů (např. Eva Burianová,...) v čele s Jiřím Dostálem uvádí svůj vlastní náhled na pojem výukový software. V diplomové práci si představíme tři definice výukového softwaru.

"Výukový software je jakékoliv programové vybavení počítače, které je určeno k výukovým účelům a dokáže plnit alespoň některou z didaktických funkcí."¹

"Vzdělávací software je program, jehož struktura a obsah vedou uživatele k osvojení nebo procvičení si nových znalostí z určité oblasti, přičemž při tom využívá moderních didaktických metod."²

"Počítačový program, který umožňuje, aby systém člověk-počítač plnil didaktické funkce. Výukový software řídí práci počítače a podle své kvality se rozdílně adaptuje na způsob žákova učení, případně na žákovy osobnostní vlastnosti, a řídí žákovo učení. "³

¹ Definice výukového software z knihy: DOSTÁL, Jiří. Výukový software a počítačové hry - nástroje moderního vzdělávání. Journal of Technology and Information Education. 2009, Olomouc - EU, Univerzita Palackého, Ročník 1, Číslo 1, s. 24 - 28. ISSN 1803-537X (print). ISSN 1803-6805.

² Definice výukového software z knihy: BURIANOVÁ, Eva. *Matematický a výukový software*. Vyd. 1. Ostrava: Ostravská univerzita, 2003, 74 s. Systém celoživotního vzdělávání Moravskoslezska. ISBN 80-704-2867-8.1

³ Definice výukového software z knihy: PRUCHA, Jan, Eliška WALTEROVÁ a Jiří MAREŠ. *Pedagogický slovník*. 6., rozš. a aktualiz. vyd. Praha: Portál, 2009, 395 s. ISBN 978-807-3676-476.

Ze všech předložených vymezení pojmu výukového softwaru vyplívá, že se jedná o počítačový program určený k výuce, který musí splňovat určitá didaktická kritéria.

V některých publikacích autoři uvádějí ještě termín edukační software. Edukační software je takové programové vybavení počítače, které umožňuje rozvoj osobnosti žáka.

1.2 Didaktické funkce výukového softwaru

Výukový software musí splňovat alespoň jednu z požadovaných didaktických funkcí. Mezi didaktické funkce řadíme motivaci žáků, expozici učiva, upevnění a osvojení poznatků a dovedností, kontrolu získané úrovně vědomostí a dovedností, reakce podle výsledků zpětnovazební informace.

Motivace žáka

Platí stejný princip jako při výuce bez výukového softwaru učitel by měl žáka správně motivovat. Motivace by měla probíhat takovým způsobem, aby splňovala určitá didaktická kritéria. Nestačí, aby výukový software obsahoval jen spoustu barevných obrázků, které žáky zaujmou, ale jeho obsah by měl žákovi rozvíjet dovednosti.

Expozice učiva

Výukový software slouží k rozšiřování dovedností a znalostí žáka. Cvičení, která nalezneme ve výukovém softwaru, by měli být přizpůsobeny rozvoji dovedností a znalostí. Vhodným doplněním cvičení jsou poučky vysvětlující jednotlivé témata, na která jsou cvičení zaměřená. Poučky ve výukovém softwaru by měli být srozumitelné a volně dostupné pro žáka (např. možnost tisku).

Upevnění a osvojení poznatků a dovedností

Ve výukovém software by se měli nacházet různorodá cvičení, které žák může od nejjednoduššího po nejtěžší absolvovat. Vždy by měl žák začínat těmi nejjednoduššími cvičeními a postupně přecházet na těžší cvičení v případě, ale pouze v případě, že zvládá předcházející učivo.

Kontrola získané úrovně vědomostí a dovedností

Podstatnou funkcí výukových softwaru je okamžitá odezva. Při splnění cvičení žákovi výukový software poskytuje hodnocení. Žák získává obratem informaci, jestli danému učivu rozumí nebo jestli je zapotřebí znovu si dané učivo projít. Okamžité vyhodnocení cvičení žáka dokáže pozitivně, ale i negativně motivovat.

Reakce podle výsledků zpětná vazba na informace

Pokud získává informaci, zda v daném tématu obstál nebo ne. Případně je mu nabídnuto shrnutí učiva, nebo části učiva, v kterých zaostává a měl by se doučit. Pro učitele jsou výsledky z výukového softwaru důležité a snadno dosažitelné.

1.3 Členění výukového softwaru

Výukový software můžeme členit hned z několika hledisek. V této podkapitole se zaměříme na některá z mnoha členění výukového softwaru.

Podle úrovně vzdělávání

Výukový software podle úrovně vzdělávání členíme na mateřské, základní, střední a vysoké vzdělávání. Výukové softwary jsou upraveny pro každou úroveň vzdělávání tak, aby splňovali očekávaná didaktická kritéria. Výukový software se přizpůsobuje úrovni vzdělání od nejjednoduššího po nejsložitější.

Podle interaktivity

Podle interaktivity dělíme výukový software na dvě úrovně interaktivní a bez interaktivních prvků. Pro výuku s využitím výukového softwaru je vhodnější používat interaktivní výukový software. Žák interaktivní výukový software řídí sám a může ovlivňovat jeho práci. Zatím co u výukového softwaru bez interaktivních prvků práci programu neřídí.

Podle organizovanosti vzdělávání

Výukový software dělíme na software pro školní výuku a software pro samostudium. Neznamená to, že by výukový software byl navržen striktně pro školu nebo samostudium. Většina těchto výukových software se liší pouze v licenci.

Podle on-line a off-line funkce

Výukový software nainstalovaný na počítači, který lze spustit pouze na příslušném počítači členíme do kategorie off-line. Výukový software, který se nachází na vzdáleném serveru a je nutné jej sdílet prostřednictvím internetu, patří mezi off-line softwary s on-line podporou. Poslední kategorií jsou on-line výukové softwary, které instalujeme do počítače a prostřednictvím internetu získáváme on-line podpory.

Podle tematického rozsahu

Podle tematického rozsahu rozlišujeme výukový software monotematický a polytematický. Monotematický výukový software obsahuje pouze jedno dané téma (např. procvičování vyjmenovaných slov). V polytematickém výukovém softwaru nalezneme víc témat (např. vyjmenovaná slova, předložky s/z, předpony s/z).

Podle funkcí výukového software

Software k výkladu učiva: Takový software, který převážně obsahuje jen výklad učiva. Software k výkladu učiva nahrazuje učebnici v psané podobě. Skládá se z textů. Software k výkladu učiva bývá vzájemně propojen pomocí odkazů. Což je výhodou takových softwarů, protože žáci jednoduše přechází na navazující učivo. Nevýhodu v sobě skrývají zdlouhavé texty, které žáci nejsou schopní přečíst do konce. Tato nevýhoda se ovšem projevuje i u učebnic v psané podobě.

Software k procvičování učiva: Software primárně poskytuje různá cvičení k probranému učivu. V takovém software se většinou nachází velké množství cvičení. Žák tak naráží na různé problémy spojené s probraným učivem. Software k procvičování učiva slouží k rozvoji žákových dovedností. Velkou výhodou je okamžitá zpětná vazba (kontrola). Žák může v tomto případě hned reagovat na svou chybu.

Software ke zkoušení a testování: Pracuje na podobném principu jako software k procvičování učiva. Rozdíl je v tom, že software se zaměřuje na kontrolu zvládnutí učiva. Většinou se výsledek zobrazí okamžitě i s navrženou známkou, ale software určený ke zkoušení a testování už nenabízí možnost opravy. Takový software může žákovi podat zpětnou vazbu tak, že mu vypíše, z jakého důvodu udělal chybu.

Demonstrační a motivační software: Software, který umožňuje promítat filmy, animace, schémata, grafy. Demonstrační a motivační software dokáže upoutat

a motivovat žáka. Velká výhoda je využití například v hodině fyziky, kde je možné takovým způsobem ukázat co se děje při zchlazení kapaliny.

Software řízení experimentů a procesů: U software řízení experimentů a procesů počítač snímá určité hodnoty, které jsou následně programem statisticky a graficky vyhodnoceny. Slouží pro větší představivost žáků v probíraném učivu (např. napětí proudu).

Počítačové hry: Pod tím to pojmem je nutné si představit počítačové hry, které plní určitou didaktickou funkci. Slouží k doplnění probíraného učiva. Počítačové hry slouží k pobavení a motivaci žáka. Mohou být určeny k domácí výuce. Samozřejmě je nutné zvolit vhodnou počítačovou hru a nenechat dítě u počítačové hry sedět dlouho i v takovém přídě, že je hra zaměřená na vzdělávání.

1.4 Výhody a nevýhody výukového software

Využití výukového softwaru ve výuce s sebou přináší určité výhody a nevýhody. V následujícím textu se zaměříme na některé výhody a nevýhody využití výukového software při výuce.

Výhody výuky s využitím výukového softwaru

- Výuka při, které bývá využíván výukový software, je pro žáky zajímavější a zábavnější formou výuky.
- Žáci si líp pamatují probírané učivo, jelikož jej vnímají zrakem i sluchem najednou.
- ✓ Při využití výukového softwaru rozvíjíme žákům kreativní myšlení.
- ✓ Na probírané učivo navazují cvičení, kde si žáci v daný moment zopakují probírané učivo.
- ✓ Po splnění úkolů dochází k zpětné vazbě (okamžitá kontrola). Žák po splnění úkolu zjišťuje, kde udělal chybu. Případně jakým způsobem vzniklou chybu má opravit.
- ✓ Výukový software slouží k nahrazení různých didaktických pomůcek (např. učebnic,...).
- Výukový software si mohou žáci pořídit domů. Poté je možné, aby plnili zadané úkoly doma a procvičovali probrané učivo stejnou formou doma.

Nevýhody výuky s využitím výukového softwaru

- ✓ Velkou nevýhodou výuky za pomocí výukového softwaru se stává neustálé vysedávání u počítače. Nesprávným sezením u počítače, dlouhodobým vysedáváním u počítače dochází k řadě zdravotním problémů (např. zhoršování zraku, trpí celý pohybový aparát,...).
- Mezi žáky dochází k změně komunikace. Nejsou zvyklí pracovat v kolektivu.
 Může docházet i k úplnému odcizení jedince od kolektivu.
- ✓ V případě dlouhodobého užívání i u výukového softwaru vzniká závislost.

Nevýhody při využití výukového softwaru pro školu

- Zakoupení výukového softwaru, který splňuje požadované kritéria školy, může být velmi nákladné.
- Nedostatečná kapacita učeben, které by byly přizpůsobeny pro výuku za pomocí výukového softwaru.
- Vysoké náklady spojené s vytvořením prostor pro výuku s výukovým softwarem.

1.5 Dílčí závěr kapitoly výukový software

V kapitole výukový software jsme si vymezili pojem výukový software a představili základní podstatu výukových softwaru. Vyplívá z toho, že podstatou pro výukový software je plnění didaktických funkcí.

V závěru kapitoly jsme se zaměřili na výhody a nevýhody výukových softwaru. Z předložených argumentů vyplývá, že převládají negativní vlivy výuky tvořené za pomocí výukového softwaru. Žáci tráví velké množství hodin u počítače. Nejen ve škole, ale i doma. Ideálním řešením tohoto negativního problému může být klasická hodina kombinovaná s výukou za pomocí výukového softwaru. V takovém případě si myslím, že by měl výukový software hodinu doplnit například na posledních 10-15 minut. Výukový software by měl spíš plnit funkci procvičování probíraného učiva.

V úvodu jsme zmínili, že se budeme zabývat konkrétními výukovými softwary určenými pro výuku českého jazyka a literární výchovy. Podívejme se tedy v následující kapitole na konkrétní příklady výukových softwaru využívaných ve výuce českého jazyka a literární výchovy.

2 Výukový software pro výuku českého jazyka a literární výchovy

Pro výuku českého jazyka a literární výchovy nalezneme na českém trhu velké množství titulů výukového softwaru. Zaměřme se na výukové softwary, které jsou u nás nejdostupnější, nejrozšířenější a nejzajímavější. V následující kapitole si projdeme, co nabízí jednotlivé výukové softwary určené pro výuky českého jazyka. Firmy, které si v diplomové práci představíme, nevytváří primárně výukový software pro český jazyk a literaturu, ale zabývají se velkým množstvím předmětů.

2.1 Výukový software od firmy SILCOM

<u>Didakta</u>

Výukový software Didakta slouží k procvičování učiva. Nejedná se o výkladový software. Ve výukovém softwaru Didakta nalezneme jen skryté poučky. Všechny verze Didakty od roku 2014 mají podporu i pro všechny typy interaktivních tabulí. Některá cvičení z výukového softwaru lze vytisknout v podobě pracovních listů. Do softwaru se ukládají veškeré výsledky žáků. Výsledky zobrazují typ cvičení, datum a čas práce s cvičením, počet špatných a správných odpovědí a výslednou známku žáků.

Výukový software Didakta vychází v několika verzích (Didakta - Český jazyk 1, Didakta - Český jazyk 2, Didakta – Diktáty 1, Didakta – Diktáty 2). Každá verze se zaměřuje na jinou problematiku výuky Českého jazyka. V následujících odstavcích se zaměřím na jednotlivé verze výukového softwaru.

Po vybrání cvičení se nám spustí úvodní tabulka. Před spuštění cvičení si volíme počet úloh, který se pohybuje vždy po pěti cvičeních (5-30). U cvičení Didakta – Diktáty 1 a 2 volíme ještě typ cvičení. Existují čtyři možnosti výběr jevů, diktát, doplňování jevů, opravy chyb. U výběrů jevů výukový software nabízí nabídku, z které je možné si vybrat jeden jev. Diktování diktátů spouštíme stisknutím amplionu. Jednou z možností je počet diktování diktátu, který určujeme v nastavení. Máme na výběr ze čtyř možností (1x, 2x, 3x, neomezeně). Při opravě chyb se objeví věta, kterou musíme opravit.

Po vyplnění cvičení klikneme na tlačítko kontrola. Po stisknutí tlačítka kontrola se za jednotlivými částmi cvičení objeví žlutá fajfka nebo červený křížek. Kliknutím na křížek zjistíme, kde jsme udělali chybu. Stisknutím tlačítka pokračovat se zobrazí tabulka s výsledky, kde se zobrazuje i výsledná známka.

Didakta – Český jazyk 1

Výukový software zaměřený na procvičování větné skladby. Najdeme zde pět tematických okruhů (základní větné členy, rozvíjející větné členy, věta jednoduchá, věty hlavní a vedlejší, souvětí), které se dále člení na jednotlivá cvičení.

Typy cvičení:

- ✓ základní větné členy základní skladební dvojice, druhy podmětu, podmět jako slovní druh, druhy přísudku
- ✓ rozvíjející větné členy určování větných členů, přívlastek shodný a neshodný, přístavek
- ✓ věta jednoduchá druhy vět podle postoje mluvčího, jednočlenné a dvojčlenné věty, čárka v jednoduché větě
- ✓ věty hlavní a vedlejší poměry mezi hlavními větami, věta hlavní a vedlejší, druhy vedlejších vět, vsuvka a věta vložená
- ✓ souvětí druhy souvětí, čárka v souvětí, doplňování spojovacích výrazů.

Didakta – Český jazyk 2

Obsah výukového softwaru tvoří cvičení zaměřená na pravopis a tvarosloví českého jazyka. Opět se skládá z pěti tematických částí (pravopis hlásek, doplňování i/y, ohebné slovní druhy, neohebné slovní druhy, tvorba a význam slov).

Typy cvičení:

- ✓ 1. okruh: pravopis hlásek a hláskových skupin spodoba znělosti a zdvojené hlásky, velká písmena, u/ú/ů, ĕ/je/ně
- ✓ 2. okruh: pravopis i/y podle významu slov, po tvrdých a měkkých souhláskách, po obojetných souhláskách, koncovky podstatných a přídavných jmen, shoda podmětu s přísudkem
- ✓ 3. okruh: ohebné slovní druhy podstatná jména (určování pádu, čísla, rodu a vzoru; přiřazování konkrétních vzorů; třídění na konkrétní a abstraktní podstatná jména; určování látkových; hromadných a pomnožných podstatných jmen), přídavná jména, zájmena (skloňování a druhy), číslovky (skloňování a druhy), slovesa (určování osoby, čísla, způsobu a času; přiřazování rodů, určování vidu a třídy)
- ✓ 4. okruh: ostatní slovní druhy stupňování příslovcí a přídavných jmen, druhy předložek, druhy spojek, určování všech slovních druhů
- ✓ 5. okruh: tvorba a význam slov kořeny slov, sousloví a složená slova, význam přejatých slov, slabiky.

Didakta – Diktáty 1 a 2

Diktáty 1 a 2 se nedělí na tematické okruhy, ale dělí se podle ročníků na základní škole. Diktáty 1 jsou určeny pro 2. až 5. ročník základní školy a diktáty 2 pro 6. - 9. ročník.

Diktáty 1:

- ✓ 2. ročník znaménka za větou, souhlásky (měkké, tvrdé a párové), krátké a dlouhé samohlásky, obecná a vlastní jména
- ✓ 3. ročník vyjmenovaná slova, i/y po obojetných souhláskách, obtížnější slova s i/y, slovesa v přítomném čase
- ✓ 4. ročník předpony a předložky, slova příbuzná, koncovky jmen, shoda přísudku s podmětem
- ✓ 5. ročník hláskové skupiny a přípony, bě/bje, pě/pje, vě/vje, mě/mně, zájmeno já a ona, číslovky, přejatá slova, podmět, souvětí a přímá řeč
- ✓ souhrnné diktáty vždy u každého ročníku, všechny probrané pravopisné jevy najednou

Diktáty 2:

- ✓ 6. ročník čárky v souvětí, jména (osob, zvířat, věcí), zeměpisná jména a hvězdářské názvy, přídavná jména, zájmena, i/y v různých částech slova, souhrnné diktáty
- ✓ 7. ročník čárka v souvětí a znaménka za větou, čárka mezi větnými členy, názvy (institucí, událostí, děl), příslovce, spřežky a neohebná slova, odborné názvy, souhrnné diktáty
- ✓ 8. ročník čárka v podřadném souvětí, čárka v souřadném souvětí, vložené věty, přejatá slova obecná a vlastní, i/y v různých částech slova, souhrnné diktáty
- ✓ 9. ročník spodoba znělosti a hláskové skupiny, přívlastek volný a těsný, čárka ve složitém souvětí, přímá řeč, čárka v jednoduché větě, u/ů/ú, číslovky a zájmena v dopisech, souhrnné diktáty

Slovní druhy

Slovní druhy patří k výukovým softwarům od firmy SILCOM. Jak sám název napovídá, výukový software se zabývá tvaroslovím. V softwaru Slovní druhy pracujeme s výukou nebo cvičeními.

Ve výukové části se postupně představí všech deset slovních druhů. U každého slovního druhu software vždy uvádí definici, latinský název, typický příklad a názornou ilustraci.

Část s cvičeními je tvořená devíti typy her. Žáci si procvičí určování slovních druhů samostatně i ve větách, přiřazování pojmů a latinských názvů, rozpoznávání slovních druhů. V nastavení uvádíme, zda chceme procvičovat všechny slovní druhy nebo jen jeden z nich.

Výukový software Slovní druhy lze využít na interaktivních tabulích. Zajímavou možností u cvičení je tisk diplomu pro nejlepšího žáka. Vytisknout se dají i definice slovních druhů ve výukové části.

Vyjmenovaná slova

Výukový software na procvičování vyjmenovaných slov, který se dělí na dvě části výukovou a procvičovací.

Ve výukové části se postupně probírají veškerá vyjmenovaná slova. Vyjmenovaná slova jsou přehledně rozdělená podle souhlásek. Každé vyjmenované slovo reprezentuje text, obrázek a mluvená ukázka.

Okruh s procvičováním v sobě ukrývá devět her. V hrách žáci doplňují i/y, hledají chyby nebo píší diktát slov. Další cvičení pracuje s rozlišováním vyjmenovaných slov a slov, které nejsou vyjmenované. Hledání příbuzných a odvozených slov.

Samozřejmě i vyjmenovaná slova pracují na interaktivních tabulích a opět zde najdeme možnost tisku diplomu pro nejlepšího žáka.

<u>Ohebné slovní druhy 1</u>

Nejnovějším titulem od firmy SILCOM vydaným v roce 2016 jsou Ohebné slovní druhy 1. Výukový software se zabývá problematikou podstatných jmen a sloves. Hlavně mluvnickými kategoriemi. U podstatných jmen rodem, číslem, pádem, životností a vzory. U sloves osobou, číslem, způsobem a časem. V Ohebných slovních druzích 1 najdeme cvičení spojené s podmětem a přísudkem. Výukový software rozdělujeme do dvou částí procvičování a hra.

Část s procvičováním nám může pomoci při výkladu a opakování učiva. Žáci získávají okamžitou zpětnou vazbu na cvičení, které vyplní.

V části hra žák pomáhá zahraničnímu studentovi s českým jazykem. Žák se zahraničním studentem cestuje po území České republiky a společně získávají odznaky z míst, kde splnili úkol. Ve výukovém softwaru přesně vybíráme jaké kategorie a typy cvičení žáci vyřeší v dané hře.

2.2 Výukový software od firmy TERASOFT

Nejznámější a nejvydávanější výukové softwary pocházejí z dílny firmy TERASOFT. Výukové softwary od TERASOFTU se nachází téměř na každé základní škole v České republice.

<u>TS Český jazyk 1 – Pravopis</u>

Ve výukovém softwaru se objevují všechny pravopisné jevy, které se učí na základní škole. Výukový software Pravopis lze využívat již od 2. ročníku. Využití Pravopisu je vhodné i pro střední školy a gymnázia.

Ke každému pravopisnému jevu v programu najdeme motivační obrázky. Výukový software se skládá z velkého množství cvičení zaměřených na určitá témata. Při vyplňování cvičení se okamžitě vykreslí chybná odpověď, která odkrývá nápovědu. Cvičení doprovází hlasový záznam, který předčítá jednotlivá cvičení.

Cvičení ve výukovém softwaru můžeme ukládat do počítače a upravovat v textovém editoru, nebo si můžeme vytisknout pracovní listy.

<u>TS Český jazyk 2 – Jazykové rozbory</u>

Jazykové rozbory lze využít ve výuce od 3. ročníku. Výukový software Jazykové rozbory se podobá výukovému softwaru Pravopis. Rozdíl je v probíraném učivu. Cvičení jsou zaměřená na procvičování větných rozborů, mluvnických kategorií, druhů vedlejších vět a psaní čárek ve větě jednoduché a souvětí.

<u>TS Český jazyk 3 – Diktáty</u>

Ve výukovém softwaru Diktáty se objevují veškeré pravopisné jevy. Vhodný pro procvičování na základní škole nebo k přípravě na příjímací řízení na střední školy a gymnázia. Diktáty jsou rozčleněny do několika kategorií (např. první kategorie obsahuje diktáty pro první stupeň základních škol).

Ve výukovém softwaru najedeme dva režimy (procvičovací a testovací) a více než 300 cvičení. V procvičovacím režimu doplňujeme různé pravopisné jevy, které software okamžitě vyhodnotí a poskytne nám vysvětlení pravopisného jevu. V testovacím režimu nejprve vyplňujeme cvičení s pravopisnými jevy. Vyhodnocení cvičení se objeví až po úplném vyplnění. Po vyhodnocení teprve žáci získávají odůvodnění chyb, které v diktátě udělali. Po vyplnění a vyhodnocení cvičení se ukáže obrázek, který se vykreslí podle procentuálního počtu chyb.

<u>TS Český jazyk 4 – Zábavná mluvnice brouka Koumese</u>

Výukový software určený pro výuku od 2. ročníku základní školy, žáky se specifickými poruchami a přípravu na příjímací řízení na střední školy a gymnázia. Výukový software je opět zaměřený na pravopisné jevy.

V softwaru nejprve nalezneme procvičovací a testovací režim, který je rozdělený do 17 kategorií. Kategorie se skládají s podkategorií (např. kategorie vyjmenovaná slova a podkategorie vyjmenovaná slov po B). U každého cvičení je navíc uvedený ročník, pro který jsou uvedená cvičení vhodná. Chyby výukový software Zábavná mluvnice brouka Koumese vyhodnocuje stejně jako výukový software Diktáty.

V další části žáci pracují s dobrodružnou hrou, v které provází detektiva brouka Koumese (namluvil Jiří Lábus). Cílem hry je pomoci detektivu Koumesovi vyřešit záhadu z hmyzí louky. Detektivu Koumesovi pomáhají řešit stopy pomocí znalosti pravopisu. Dobrodružné hře lze nastavit několik úrovní obtížnosti.

<u>TS Český jazyk 5 – "Přijímačky"</u>

Výukový software Přijímačky je určen pro žáky 7. a 8. ročníku základních škol. Přijímačky se zaměřují na všechny typy středních škol a gymnázií. K sestavení výukového softwaru Přijímačky byly použity záznamy ze starých přijímacích řízení.

Výukový software se skládá z několika kategorií, kde najdeme různé typy úloh a různé okruhy učiva. Žáci si mohou vyzkoušet 120 zkoušek, které jsou zaměřené na pravopis (20 zkoušek), tvoření a význam slov (10 zkoušek), tvarosloví (30 zkoušek), skladba (20 zkoušek), literatura (10 zkoušek), souhrnná cvičení (30 zkoušek). Souhrnná cvičení v sobě neobsahují pravopis. Nejlepším řešením je kombinace souhrnných cvičení a pravopisu. Každá zkouška se skládá z 15 cvičení. V Přijímačkách tak najdeme více než 1800 cvičení. Každé cvičení je doprovázeno správnou odpovědí a vysvětlením správné odpovědi.

Software Přijímačky nabízí i 120 pracovních listů, které se dají vytisknout. Součásti těchto pracovních listů jsou i správné odpovědi.

<u>TS Český jazyk 6 – Procvičujeme pravopis</u>

Výukový software Procvičujeme pravopis navazuje na TS Český jazyk 4 -Zábavná mluvnice brouka Koumese. Skládá se celkem z 300 cvičení, které slouží k procvičování pravopisných jevů a je určený od 2. ročníku základních škol. Při opakování vhodný i pro střední školy a gymnázia. Z výukového softwaru lze tisknout pracovní listy.

Software se skládá ze 17 kategorií (např. pravopisná cvičení specifikovaná pro 2. ročník, vyjmenovaná slova, koncovky podstatných jmen, koncovky u sloves, koncovky přídavných jmen, předložky s a z, předpony s-, z-, vz-, psaní velkých a malých písmen, věta jednoduchá a souvětí, psaní slov přejatých).

Opět výukový software pracuje s procvičovacím a testovacím režimem, kde se nám chyby opravují okamžitě, software podává vysvětlení těchto chyb.

Nová dobrodružná hra detektiva Koumese žáky tentokrát zavede do Egypta, kde byly odcizeny vzácné koberce. Žáci pátrají po kobercích a klíči k pokladu. Jednodušší verze hry je určena mladším dětem, kde děti řeší úkoly ve správném pořadí.

<u>TS Český jazyk hrou – Vyjmenovaná slova</u>

Vyjmenovaná slova v sobě ukrývají šest napínavých her určených pro mladší ročníky.

- ✓ 1. hra skládání slov Při zpuštění hry žáci volí buď jedno vyjmenované slovo, nebo všechny vyjmenovaná slova. Z jednotlivých písmenek žák vytváří vyjmenované slovo.
- ✓ 2. hra slova příbuzná Žáci rozlišují, zda se jedná o příbuzné slovo a zařadí příbuzné slovo k správnému vyjmenovanému slovu.
- ✓ 3. hra řady slov Z ilustrovaných kartiček seřazují správné pořadí jednotlivých vyjmenovaných slov.
- ✓ 4. hra souboj pirátů V souboji pirátských lodí se žáci snaží pomocí vyjmenovaných slov zachránit loď před potopením
- ✓ 5. hra Člověče, nezlob se! Hra určená pro jednoho nebo dva hráče, kteří doplňují i/y ve vyjmenovaných slovech nebo slovech příbuzných.
- ✓ 6. hra Zachraňte princeznu! Žáci zachraňují princeznu ze zajetí zlého čaroděje za pomocí vyjmenovaných slov.

2.3 Výukový software od nakladatelství Tobiáš

Pavučinka

Výukový software Pavučinka nakladatelství Tobiáš přizpůsobilo pro výuku dysortografických žáků. Nabízí výklad učiva s propojením interaktivních jevů. Probírané učivo v Pavučince se vždy dělí na postupné kroky. Výukový software Pavučinka vychází v několika verzích Pavučinka Podstatná jména, Pavučinka Vyjmenovaná slova a Pavučinka Pravopis.

Pavučinka je rozdělená do různých typů cvičení. Cvičení nabízí možnost výuky nebo možnost testování. Při výuce žák dochází k řešení cvičení postupně (viz. Schéma číslo 1). V testovací možnosti vyplňuje cvičení dle svých vědomostí. Výukový software reaguje na každé vyplnění cvičení okamžitě.



Kliknutím na slovo dav_ se zobrazí postup určení vzoru podstatného jména. Nejprve si žák vybírá z jednotlivých rodů. V případě podstatného jména dav dále žák volí životnost. Pak změny koncovek v prvním a druhém pádě a nakonec vzor podstatného jména. Kliknutím na vzor podstatného jména se vrátí k větě, v které měl žák určit daný pravopisný jev. Nad podstatným jménem dav se žákovi zobrazí daný vzor v našem případě hrad. Vzor hrad se přizpůsobí pádu určovaného podstatného jména dav.

2.4 Diktátor

Výukový software Diktátor obsahuje více než 1000 různých diktátů zaměřených na různé pravopisné jevy. Diktátor slouží k samostudiu doma. Správce počítače (rodič) v nastavení softwaru Diktátor zvolí heslo k počítači, nastaví obtížnost dle ročníků pro dítě a taky na jak dlouho se počítač spustí podle počtu chyb. Diktátor se vždy spustí při načtení počítače a dítě musí vyplnit daný diktát, který se po úplném vyplnění opraví a ukáže chyby, které dítě udělalo. Počítač se dítěti spustí dle počtu chyb (např. nastavíme 1 chyba = 60 minut). Po uplynutí lhůty Diktátor spustí nový diktát. Dítě tak musí vyplnit další diktát a opět mu bude po vyhodnocení přidělena lhůta, kterou může strávit na počítači.

Pokud rodič do programu zapíše svůj e-mail automaticky bude dostávat zprávy o chybách dítěte. Existuje i množnost zasílání upozornění, že dítě dokázalo obejít heslo a dlouhou dobu nepsalo diktát.

2.5 Výukový software určený pro tablety a mobilní telefony

Abeceda pro děti

Abecedu pro děti vytvořila PMQ software. Aplikace přizpůsobená na androidy, apply a windows. Určená pro 1. ročník základních škol nebo předškolní děti v mateřských školách. Aplikaci si můžeme stáhnout ve freeware verzi (písmena A-H) nebo plné verzi (cena plné verze 89 Kč). Celou aplikaci Abeceda pro děti provází ženský hlas.

Výuková aplikace Abeceda pro děti se skládá z devíti her (abeceda, velká písmena, malá písmena, najdi kartičku, tvoření slov, písmena ve slově, pexeso, první písmeno ve slově, poslední písmeno ve slově).

Hry:

- ✓ 1. hra: abeceda V první hře listuje dítě abecedou, kde poznává vzhled písmene a ženský hlas mu vždy přečte název písmene.
- ✓ 2. hra: velká písmena V náhodném pořadí se zobrazují velká písmena, která jsou doplněná obrázkem (např. H jako HASIČ).
- ✓ 3. hra: malá písmena Hra malá písmena pracuje stejným způsobem jako hra velká písmena.
- ✓ 4. hra: najdi kartičku Dítě má kliknout na kartičku s obrázkem, který začíná na zobrazené písmeno (např. D, d kartičky představují dárek, housku a Ferdu mravence). Ženský hlas vždy přečte, co je na kartičkách s obrázkem zobrazeno.

- ✓ 5. hra: tvoření slov Otáčením políček dítě zobrazuje písmena a vytváří tak slovo. Písmena jsou dítěti předčítána nejprve postupně. Po složení celého slova ženský hlas přečte dané slovo.
- ✓ 6. hra: písmena ve slově Dítě hledá v daném slově písmeno, které mu zadá ženský hlas (např. Ananas najdi písmeno A, dítě musí najít všechny písmena A). Za správné splnění úkolu dítě získává sluníčko. Po nasbírání osmi sluníček dítě čeká odměna.
- ✓ 7. hra: pexeso Po zobrazení dvaceti kartiček dítě hledá kartičky, které patří k sobě. Písmeno vždy spojuje s obrázkem. Obrázek začíná daným písmenem (A Auto, C Cibule).
- ✓ 8. hra: první písmeno ve slově Pod obrázkem se objeví tři písmena. Dítě klikne na správné písmeno po přečtení názvu na obrázku. V tomto případě stiskne písmeno, které slyší na začátku slova (např. Eliška – dítě stiskne E).
- ✓ 9. hra: poslední písmeno ve slově Poslední hra Abecedy pro děti pracuje na stejném principu jako devátá hra. Tentokrát ovšem dítě hledá poslední písmeno (např. datel – dítě stiskne L).

<u>Český jazyk – Pravopis</u>

Aplikaci Český jazyk – Pravopis vytvořila firma PMQ software ve verzi pro android, apple a windows. Freeware licence umožňuje přístup jen k některým částem pravopisných cvičení. Plná licence zajistí úplné pracování aplikace (cena 89 Kč).

V aplikaci procvičují žáci pravopisné jevy (např. psaní ú/ů, měkké a tvrdé souhlásky, vyjmenovaná slova, psaní i/y v koncovkách podstatných jmen, shodu přísudku s podmětem,...). U každého cvičení si mohou zvolit jednu ze tří oblastí (jak na to, procvičování a zkoušení na čas). V oblasti jak na to žáci najdou poučky k danému pravopisnému jevu. Procvičování vždy nabízí dvanáct slov nebo spojení slov, u kterých žáci určují daný pravopisný jev. Poslední oblast zkoušení na čas obsahuje dvanáct slov nebo slovních spojení, které musí žáci stihnout vyplnit na daný čas.

V aplikaci Český jazyk – Pravopis se automaticky ukládají výsledky testů a chyby, které žák udělal. Kliknutím na tlačítko procvičení chyb se zobrazí na procvičování jen slova a slovní spojení, v kterých žák udělal chybu.

Nauč se pravopis

Aplikace určená k procvičování pravopisných jevů. Ve freeware verzi aplikace nabízí procvičování pravopisných jevů (vyjmenovaná slova, předpony s/z, bě/bje, mě/mně, vě/vje, koncovky i/y, ú/ů) a taháky (vyjmenovaná slova a vzory). Do aplikace je možné si zakoupit velká písmena, lektora a statistiky.

Žák si může vybrat jeden nebo všechny pravopisné jevy. Počet příkladů v cvičení se dá nastavit od 4 do 40. Každé cvičení ukazuje po vyřešení příkladu špatnou nebo správnou odpověď a vysvětlení procvičovaného jevu. Na další příklad se žák přesouvá sám.

Psaní s motýlkem

Aplikace psaní s motýlkem vychází z písma comenia script a je určená pro děti v prvním ročníku. Ve freeware verzi najdeme jen písmena od A po Č.

Dítě se učí psát velká a malá písmena a číslovky. Průvodcem aplikace je malý motýlek, který u každého písmene ukazuje dítěti cestu, kterou má udělat čáru. V aplikaci můžeme nastavit, zda je dítě levák nebo pravák.

2.6 Dílčí závěr kapitoly výukový software pro výuku českého jazyka a literární výchovy

Podíváme-li se na představené výukové softwary, je velmi náročné přesně určit, který z těchto výukových softwarů je nejideálnějším pomocníkem při výuce. Chceme-li zvolit správný výukový software pro výuku českého jazyka, musíme být seznámeni s konkrétním obsahem daného výukového softwaru. V kapitole jsme se seznámili s různými druhy výukových softwarů a podle popisu jsme schopni určit na jaké téma a pro koho je daný výukový software zaměřen. Podstatnou záležitostí je samotný výukový software si vyzkoušet, alespoň v demo verzi, abychom si sami udělali představu o principu cvičení. Často si vybereme výukový software, ale od jeho koupě nás odradí pořizovací cena.

Představené výukové softwary nejsou všechny primárně určeny jen do škol, ale taky si je můžeme pořizovat do domácností pro mimoškolní rozvoj dítěte.

S předchozího textu vyplívá, že výukové softwary jsou určené převážně pro výuku mluvnice. Výuka literární výchovy je hodně v pozadí a nalezneme jen malé množství výukových softwaru. V uvedených výukových softwarech najdeme cvičení do literární výchovy jen ve výukovém softwaru TS Český jazyk 5 – "Přijímačky".

Výukové softwary do českého jazyka se buď zaměřují konkrétně na nějaké téma mluvnice (např. vyjmenované slova) nebo shrnují veškerou mluvnici.

Pokud si žádný výukový software nevybereme, jelikož nám nevyhovuje z nějakých osobních důvodů, nebo nás odradí pořizovací cena, můžeme přistoupit na jinou variantu. Buď upustíme úplně od výuky za pomocí výukového softwaru, nebo si takový výukový software můžeme vytvořit sami.

Podívejme se nyní na možnosti tvorby výukového software. Nejprve se musíme, ale zaměřit na programovací jazyky, které nám pomohou vytvořit takový výukový software.

3 Programovací jazyky

V předešlých kapitolách diplomové práce jsme si přiblížili výukový software a některé výukové softwary pro výuku českého jazyka. Podívejme se na možnost co dělat v případě, že nabídka výukových softwarů nesplňuje naše očekávání ani kritéria. V případě, že výukový software nesplňuje naše požadavky, je možné si výukový software navrhnou a vytvořit podle našich kritérií. Pro vytvoření požadovaného výukového software využijeme programovacích jazyků. V následující části si přiblížíme podle mého názoru nejvyužívanější programovací jazyky.

Než přejdu k samotným programovacím jazykům, je dobré si definovat pojem programovací jazyk. Programovací jazyk můžeme chápat jako jazyk, pomocí kterého vytváříme programy nebo aplikace. Programování je vlastně vytváření postupů, jimiž řešíme dané úlohy (tzv. algoritmizace).

3.1 C/C++

Mezi nejstarší programovací jazyky patří jazyk C, který v 70. letech 20. století navrhl Dennis Ritchiev. Vývoj programovacího jazyka C byl spjatý s Unixem. Patřil mezi první programovací jazyky, které využívaly strukturované programování. Velká většina dnešních programovacích jazyků vychází převážně z jazyka C.

Programovací jazyk C++ má základ v jazyce C. Za jeho zrodem stál Bjarne Stroustrup. Programovací jazyk C je procedurální jazyk, který se skládá z dat a algoritmů. Algoritmy v případě jazyka C vytvářejí informace k datům. Pro programovací jazyk C++ platí, že pracuje s daty. Tedy je objektově orientovaným jazykem. C++ patří mezi nejpoužívanější a zároveň nesložitější programovací jazyk.

Zdrojový kód číslo 1: Ukázka zdrojového kódu C (Hello World):

#include <stdio.h>
int main(void)
{ printf("Hello World\n");
 return 0; }

Zdrojový kód číslo 2: Ukázka zdrojového kódu C++ (Hello World):

#include <iostream> int main(void) {std::cout << "HelloWorld\n"; return 0;}

3.2 Python

Mezi dynamicky interpretované jazyky patří Python, u kterého se chyby projevují až po spuštění vytvořeného programu. Tento jazyk je vyvíjen jako opensource. Open-source znamená, že si každý může přečíst zdrojový kód.

Syntaxe programovacího jazyka Python spadá mezi jednoduché. Jazyk Python má velké množství datových typů (např. počítají i s komplexními čísly). Samotné jádro a funkce Pythonu jsou vytvořeny v jazyku C.

Zdrojový kód číslo 3: Ukázka zdrojového kódu Python (Hello World):

```
# -*- coding: utf-8 -*-
from Tkinter import *
root = Tk()
w = Label(root, text="Hello World!")
w.pack()
root.mainloop()
print "Konec"
```

3.3 Pascal

Spolu s jazykem C je Pascal nejstarším programovacím jazykem, vytvořeným v 70. letech profesorem Niklausem Wirthem (Curych). Wirth chtěl vytvořit programovací jazyk vhodný pro výuku programování, který by měl srozumitelnou konstrukci. Pascal byl vytvořen tak, aby byl schopný pracovat na většině tehdejších počítačů.

Velký úspěch měla verze Turbo Pascal od firmy Borland. Turbo Pascal vyniká svým rychlým kompilátorem. Nástupcem Turbo Pascalu se v 90. letech stává Delphi.

```
Zdrojový kód číslo 4: Ukázka zdrojového kódu Pascal (Hello World):

program Hello World;

uses crt;

begin

clrScr;

writeln('Hello World!');

repeat

until keyPressed;

end.
```

3.4 Delphi

Jak už jsem se zmínila v podkapitole Pascal, Delphi se vyvinulo z Turbo Pascalu. Delphi patří mezi objektově orientované programovací jazyky. Mezi významné rysy Delphi patří:

- ✓ transparentní manipulace s objekty stejně jako s ukazateli;
- ✓ vlastnosti (properties) jako část jazyka, které transparentně zapouzdřují přístup k položkám objektu, včetně pohodlného indexování umožňující přímý přístup k objektům pole a jejich vlastností;
- ✓ vlastnosti (properties) jako část jazyka, které transparentně zapouzdřují přístup k položkám objektu (Položka může představovat i pole prvků, ke kterým se lze dostat pomocí celočíselného indexu. V objektu pak musí být definovány funkce pro čtení/zápis prvku v závislosti na indexu.);
- ✓ bezpečná práce s typováním tříd a metod;
- ✓ přenesení implementace rozhraní do položky nebo vlastnosti třídy;
- ✓ snadné zpracování systémových zpráv (metoda třídy, která má být volána v okamžiku, kdy třída obdrží danou zprávu, se jednoduše označí jménem této zprávy);
- ✓ komfortnější práce s integrovaným debuggerem (např. možnost nastavování breakpointů i za běhu aplikace včetně podmínkového omezení).⁴

Zdrojový kód číslo 5: Ukázka zdrojového kódu Delphi (Hello World):

lblNapis.Caption := 'Hello World!'

3.5 Java

V roce 1991 vzniká projekt s názvem "Zelený projekt", který se zabývá myšlenkou programovacího jazyka, který by dokázal bez kompilace běžet na zařízeních spotřební elektroniky. Společnost SunMicrosystems přichází s programem Oak. Jelikož autor programu Oak zjišťuje, že název Oak byl již použit u jiného programu, přejmenovává svůj program na Javu. Platforma Java vychází poprvé v roce 1995.

V 90. letech 20. století přichází velký bum internetových stránek. Internetové stránky mají svůj obsah statický. Java přichází s možností dynamických internetových stránek, které dovolují každému uživateli individuální zobrazení obsahu. Java

⁴ Rysy Delphi převzané z internetových stránek: Fakulta informatiky Masarykovy univerzity [online]. Dostupné z: https://www.fi.muni.cz/usr/jkucera/pv109/2003p/xskriva1.historie.htm

tedy přizpůsobuje obsah uživatelům a navíc se dokáže spustit na jakémkoliv druhu počítače bez kompilace.

Zdrojový kód programovacího jazyka Java obsahuje jednu nebo více tříd, které v sobě ukrývají instrukce a data programu.

Zdrojový kód číslo 6: Ukázka zdrojového kódu Java (Hello World): public class HelloWorld { public static void main(String[] args) { System.out.println("Hello World!!!");}}

3.6 JavaScript

JavaScript patří mezi skriptovací programovací jazyky. Umožňuje nám vytvářet dynamické internetové stránky. Vlastní chod JavaScriptu zajišťuje internetový prohlížeč, který posuzuje programovací jazyk pomocí modulu DOM.

JavaScript dokáže pracovat s okny internetových prohlížečů. Umožňuje změnit obrázek podle polohy kurzoru myši (efekt rollover). V JavaScriptu vytváříme hierarchické (folder) nebo roletové nabídky (pull-down).

Kontroluje a opravuje správnost údajů, které zadá uživatel internetových stránek ještě před odesláním na server.

```
Zdrojový kód číslo 7: Ukázka zdrojového kódu JavaScript (Hello World):
<script type="text/javascript">
alert('Hello World');
</script>
```

3.7 PHP

PHP (Hyperttext Preprocessor) patří mezi skriptovací programovací jazyky. Vytváříme jeho pomocí dynamické internetové stránky a internetové aplikace. PHP se vkládá přímo do zdrojového kódu internetové stránky napsané ve verzích jazyka HTML. Syntaxi jazyka PHP inspirovalo hned několik programovacích jazyků (C, Pascal, Java).

```
Zdrojový kód číslo 8: Ukázka zdrojového kódu PHP (Hello World):
<?php
echo "Hello World!";
?>
```

3.8 Dílčí závěr kapitoly programovací jazyky

Po přečtení kapitoly programovací jazyky je patrné, že se nám nabízí spousta programovacích jazyků pro tvorbu výukového softwaru. Představili jsme si hned několik nejpoužívanějších programovacích jazyků, kterými můžeme vytvářet výukový software.

Z ukázek zdrojového kódu je hned na první pohled patrné, že každý programovací jazyk se liší svým zápisem. Zdrojový kód je podstatou celého vytvářeného softwaru. Správný zápis programovacího jazyka je základem pro správné pracování výukového softwaru.

Uvedené programovací jazyky se mohou svou strukturou zdát mnohdy komplikované a nepřehledné. Mezi programovacími jazyky najdeme jeden, který je svou strukturou mnohem jednodušší a přehlednější. Jedná se o programovací jazyk Visual Basic. V následujících kapitolách se zaměříme na strukturu programovacího jazyka Visual Basic.

4 Visual Basic

Po zvážení náročnosti některých programovacích jazyků jsme se rozhodli využít k tvorbě výukového softwaru programovací jazyk Visual Basic. V následujících podkapitolách si představíme programovací jazyk Visual Basic a pak se zaměříme na tvorbu výukového softwaru v programovacím jazyce Visual Basic. Pro tvorbu software je podstatné znát strukturu zdrojového kódu, abychom mohli pracovat s Visual Basicem představíme si ovládací prvky, procedury, proměnné a příkazy, s kterými budeme následně pracovat v praktické části diplomové práce, kde si vytvoříme výukový software.

4.1 Historie Visual Basic

Následující podkapitola pojednává o historii Visual Basic. Od první verze Visual Basic 1.0 až po poslední verzi Visual Basic 2015. V dalších podkapitolách se budeme zabývat už jen verzí Visual Basic 2010.

Visual Basic 1.0 - 6.0

Předchůdcem programovacího jazyka Visual Basic byl programovací jazyk Basic. Basic v roce 1963 navrhnul John Kemeny a Thomas Kurz. Basic byla zkratka pro Beginner's All-purpose Symbolic Instruction Code. Právě z Basicu se vyvíjely další programovací jazyky.

Visual Basic pro Windows vychází poprvé v roce 1991 pod názvem Visual Basic 1.0. O rok později vychází Visual Basic 1.0 pro DOS. U Visual Basic 1.0 pro DOS bylo rozhraní textové a využívalo high ASCII znaky pro simulování vzhledu GUI.

Ve stejném roce vychází Visual Basic 2.0, kde bylo snadnější využití programového prostředí, a zvýšila se rychlost operací.

Poté vychází v roce 1993 verze Visual Basic 3.0, která obsahovala verzi Microsoft Jet Database Engine. Tato verze umožňovala první práce s databázemi.

Roku 1995 vychází Visual Basic 4.0, který umožnil vytvářet 32bitové programy a psaní programu v non-GUI třídách.

Visual Basic 5.0 vytvořený v roce 1997 už umožňuje uživateli pracovat s vlastními uživatelskými prvky a kompilovat kód do nativních Windows kódu.

Visual Basic 6.0 z roku 1998 přichází s možností tvorby internetových aplikací.

Visual Basic 2002 - 2015

Po první části vývoje Visual Basic (verze 1.0 – 6.0) přichází delší pauza teprve v roce 2002 vydává společnost Microsoft program pod názvem Visual Basic .NET 2002. Visual Basic .Net 2002 přechází na rozhraní .NET Framework. Jedná se o knihovnu řešení, která spolupracuje s platformou Windows.

Verze Visual Basicu vytvořené od roku 2003 až do roku 2015 se staly nedílnou součástí balíku Visual Studio. Například balík Visual Sudia 2010 obsahuje následující programovací jazyky Visual C++, Visual C#, Visual Basic, Visual Web Developer.

V roce 2003 vychází Visual Studio .NET 2003, které obsahuje aktualizace .NET Frameworku. Součástí Visual Studio .NET 2003 je první podpora pro vývoj aplikací pro mobilní zařízení (využití ASP.NET nebo .NET Compact Framework).

Visual Studio 2005 nazývané Whidbey vychází v roce 2005. Z názvu je odstraněna přípona .NET, ale Visual Studio 2005 nadále pracuje s technologií .NET Framework, která je pro tuto verzi aktualizována .NET Framework 2.0 (prvek ASP.NET 2.0).

Po menší odmlce vychází v roce 2007 Visual Studio 2008. Visual Studio 2008 pracuje s verzí .NET Framework 3.5. Ve verzi 2008 se objevuje nový vizuální designer Windows Presentation Foundation a nový editor HTML/CSS. Visual Studio 2008 přichází s novým nástrojem pro analýzu kódu.

Visual Studio 2010 vychází v roce 2010, jak už sám název napovídá. Ve verzi 2010 došlo k přepracování uživatelského rozhraní s použitím Vindows Presentation Foundation a Managed Extensibility Framework. Pro verzi 2010 vznikají nové nástroje například Call Hierarchy (nástroj pro zobrazení metod volání), Quick Search (inkrementální vyhledávání mezi symboly v projektech).

Visual Studio 2012 přináší lepší vývoj aplikací pro NET. Framework 4.5 a WindowsStore (pro verzi Windows 2008). Vylepšila se navigace v řízených i neřízených projektech. Vznikl také nástroj Expression Blend pro tvorbu uživatelského rozhraní.

S vylepšeným IDE (pro běžné i pokročilé vývojáře) přichází verze Visual Studio 2013. V IDI se objevuje více barev, nový scrolbar, lepší ladění, vylepšený debugging, peek view atd. Největší změny konstrukce vyvinuty pro C++, JavaStript.

Nejnovější verzí je Visual Studio 2015, která vytváří univerzální aplikace pro všechna zařízení Windows 10.

4.2 Ovládací prvky Visual Basic 2010

Ovládací prvky slouží jako grafické nástroje, které nám vytváří uživatelské rozhraní programu. Ovládací prvky patří mezi objekty. Ovládacím prvkům ve Visual Basci přiřadíme vlastnosti, procedury, příkazy a proměnné. Ve Visual Basicu nalezneme ovládací prvky v okně s názvem Toolbox (např. Label (popis), Button (tlačítko), ComboBox (rozbalovací seznam), GroupBox (skupina boxů), CheckBox (zaškrtávací políčko), PictureBox (kreslící plátno), ListBox (posuvný seznam), TextBox (textové pole).

Ovládací prvky vkládáme do tak zvaného Form (formuláře), který se zobrazí při spuštění Visual Basicu jako jedna z možností.

Každý ovládací prvek má své vlastnosti, které si můžeme definovat podle funkce, kterou má daný ovládací prvek plnit.

4.3 Procedury Visual Basic 2010

Programové příkazy seskupujeme do tzv. procedur. Objektům formuláře přiřazujeme události, které vyvolají proceduru události. Procedura události je kód, který se spustí v programu při manipulaci s objektem. Běžným typem procedury je příkaz Sub (EndSub), které bývají označovány jako podprogram. Procedury události vyhodnocují a nastavují vlastnosti programu. Dále využívají programové příkazy k provedení potřebných činností programu.

Rozeznáváme dva typy procedur (funkce a procedury). Funkce i procedury jsou volány z procedur názvem. Funkce používáme pro výpočty, protože přijímají argumenty a vrací hodnotu v identifikátoru funkce. Procedury přijímají argumenty a předávají zpět upravené hodnoty v seznamu argumentů. Rozdíl spočívá v tom, že procedury nevracejí hodnoty spojené s jejich konkrétními názvy. Procedury se obvykle používají pro nastavení vlastností, k získání nebo zpracování vstupů a následné zobrazení výstupů.

Výhody procedur s obecným účelem:

- Umožňují vám přiřadit název často používané skupině programových příkazů;
- Eliminují opakování řádků. Proceduru stačí deklarovat jednou a program ji může spouštět, kolikrát chceme;
- Programy jsou díky nim přehlednější. Program rozdělený na malé části je srozumitelnější než program tvořený jednou velkou částí;

- Zjednodušují vývoj programů. Programy rozdělené do logických jednotek se snadněji navrhují, píší a ladí. Pokud navíc píšete program v rámci týmu vývojářů, můžete si místo celých programů vyměňovat procedury a moduly;
- ✓ Lze je opakovaně používat v dalších projektech a řešeních. Procedury standardních modulů můžete snadno začlenit do dalších projektů;
- ✓ Rozšiřují jazyk Visual Basic. Procedury mohou často provádět úkoly, které nelze provést pomocí jednotlivých klíčových slov jazyka Visual Basic nebo metod rozhraní Microsoft .NET Framework.⁵

Umístění procedury

Proceduru můžeme umístit buď do kódu formuláře, nebo do tzv. modulu. Pokud umístíme proceduru do kódu formuláře, zajistíme, že se procedura bude vztahovat na daný formulář.

Chceme-li proceduru zajistit pro celý program, nadefinujme tzv. modul. Modul se vytváří jako specifický soubor s příponou .vb. Obsahuje deklaraci proměnných nebo procedur, které využíváme v celém programu. V modulu vytváříme pouze kód nikoliv uživatelské rozhraní. Modul nepracuje v objektově orientovaném rozhraní, proto nedefinujeme strukturu, ani vlastnosti objektů. Modul nepodporuje dědičnost.

Vytváření procedur

Jak už jsem se zmínila v úvodu nejběžnější procedurou je příkaz Sub (EndSub). V ukázce zdrojového kódu číslo 9 vidíme základní syntax procedury. Podrobněji si rozebereme každý řádek základní syntaxe.

V prvním řádku procedury definujeme název procedury. Tedy poklepeme-li ve formuláři na tlačítko Button1 spustí se procedura s název Button1_Click, jak je patrné ze zdrojového kódu číslo 10.

Za názvem procedury se uvádějí tzv. argumenty, které vždy umísťujeme do kulatých závorek. Kulaté závorky musí být za názvem procedury i v případě, že bychom argument vůbec neuváděli. Argumenty jsou deklarovány jako určité typy.

Mezi proceduru uvádíme příkazy, které provádějí činnost procedury.

Visual Basic dokáže sám vygenerovat proceduru. Klikneme-li na tlačítko

⁵ Výhody procedur s obecným účelem přepsané z knihy: HALVORSON, Michael. Microsoft Visual Basic 2010: Krok za krokem. 1. vydání. Brno: Computer Press, 2012. ISBN 978-80-251-3146-6.

Button1 automaticky se do kódu vygeneruje název procedury a argumenty připadající k této proceduře.

Zdrojový kód číslo 9: Základní syntaxe procedury: Private Sub Název Procedury (argumenty) příkaz procedury End Sub

Zdrojový kód číslo 10: Ukázka procedury:

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click End End Sub

4.4 Proměnné Visual Basic 2010

Proměnou můžeme definovat jako schránku, která dočasně uchovává data v programu. Proměnné pracují s hodnotami a daty. Slouží k ukládání, čtení informací.

Zdrojový kód proměnné se skládá z příkazu *Dim* (viz. Kapitola Příkazy Visual Basic 2010) a datového typu.

Proměnné

V prvním řádku zdrojového kódu č. 1 vytváříme proměnou s pomocí příkazu *Dim* s určitým názvem a datovým typem. Proměnné zapisujeme pomocí písmen abecedy, čísel, podtržítka. Samotná proměnná nesmí nikdy začínat číslem. Pro správnou funkci proměnné volíme raději proměnnou bez diakritiky.

V druhém řádku zdrojového kódu č. 2 přiřazujeme dané proměnné hodnotu. Pomocí třetího řádku zobrazíme hodnotu proměnné.

```
Zdrojový kód číslo 11: Proměnné:
Dim ukazka As Integer
ukazka = 1
Label1=ukazka
```

Za povšimnutí stojí, že proměnnou ukazka nevkládám do uvozovek. Pokud umístím proměnou ukazka (třetí řádek zdrojového kódu) do uvozovek nenačte se po spuštění programu proměnná, ale jen text ukazka. Uvozovky využíváme
jen při přiřazování hodnoty proměnné, v které definujeme text (většinou datový typ String).

Zdrojový kód číslo 12: Přiřazování hodnoty (uvozovky):
Dim ukazka As String
ukazka = "Marek"
Label1=ukazka

Datové typy

Datové typy určují, jakým způsobem se data do paměti uloží. Datový typ vymezuje, kolik místa v paměti zabere samotná proměnná (viz. Tabulka Datové typy). Mezi nejpoužívanější datové typy patří Integer a String.

Datový typ	Použití	Velikost	Ukázka využití
Boolean	logické hodnoty (True = pravda nebo	16bitové	Dim ukazka As Boolean
	False = nepravda)		ukazka = True
Byte	malá celá čísla	8bitový	Dim ukazka As Byte
			ukazka = 1
Date	datum	16bitový	Dim ukazka As Date
			ukazka = #1/1/2016#
Decimal	hodnoty	128bitový	Dim ukazka As Decimal
			ukazka = 10000000
Double	desetinná čísla	64bitový	Dim Pi As Double
		s plovoucí	<i>Pi</i> = <i>3</i> , <i>1415926535</i>
		řádovou	
		čárkou	
Char	znak Unicodu	16bitový	Dim ukazka As Char
	u proměnné datového typu Char na		$ukazka = "\check{C}"c$
	konec hodnoty přidáme znak "c"		
Integer	celá čísla	32bitový	Dim ukazka As Integer
			ukazka = 10000000
Long	-9 233 372 036 854 775 808	64bitový	Dim ukazka As Long
	až 9 233 372 036 854 775 807		ukazka = 1000000000
Object	do proměnné Object lez uložit	32bitový	Dim MyApp As Object
	jakýkoliv typ		MyApp = CreateObject
			(,, Word.Application)

Short	-32 768 až 32 767	16bitový	Dim ukazka As Short
			ukazka = 10000
Single	3,4028235 x 1038	32bitový	Dim ukazka As Single
		s plovoucí	ukazka = 1,2
		řádovou	
		čárkou	
String	textové hodnoty	obvykle	Dim ukazka As String
		16bitů pro	ukazka = "Marek"
		znak	

Tabulka 1: Datové typy

4.5 Příkazy Visual Basic 2010

Řádek kódu v programu Visual Basic nazýváme programový příkaz. Programový příkaz můžeme vysvětlit jako samostatnou část kódu spouštěného překladačem Visual Basic. Příkazu přidělujeme práci nad určitou částí vyvíjeného programu. Příkazy tvoří platnou instrukci, kterou rozpoznává překladač.

Všechny programové příkazy musí splňovat určitá syntaktická pravidla, které definuje a uplatňuje překladač Visual Basicu. Příkazy se od sebe mohou lišit délkou (např. jedno klíčové slovo End, nebo kombinace příkazů Label1.Text = TimeString). Visual Basic poskytuje informace o syntaktických chybách a nabízí jejich alternativní řešení.

<u>Příkazy</u>

Visual Basic obsahuje velké množství příkazů, které se dají využít pro tvorbu programu. V kapitole vysvětlím funkci některých příkazů, které se dají využívat ve Visual Basicu 2010. Každý příkaz je doplněn ukázkou zdrojového kódu.

AddHandler, RemaveHandler

Příkaz AddHandler spustí, zastaví a zpracuje událost v případě, že je program zapnutý. Klíčovým slovem příkazu je slovo Handles, které definuje zpracování určitých událostí.

V uvedené ukázce zdrojového kódu číslo 13 zajištuje příkaz AddHandle volání procedury událostí tisk obrázku.

Zdrojový kód číslo 13: Ukázka použití příkazu AddHandler:

AddHandler PrinctDocument1.PrintPage, AddressOf Me. Tisk Obrázku PrintDocument1.Print()

Call

Příkaz Call přivolává nebo řídí procedury a funkce. Při použití příkazu Call stačí uvádět název volané procedury.

Zdrojový kód číslo 14: Ukázka použití příkazu Call:

Call printToDebugWindow("Hello World")

Class

Příkazem Class deklarujeme název třídy. Příkaz Class definuje vlastnosti, proměnné, události, které jsou zahrnuté do tříd.

Zdrojový kód číslo 15: Ukázka použití příkazu Class:

Public Class From1

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click End End Sub

End Class

Const

Příkaz Const slouží k definici konstanty. Konstanty se definují v případě hodnoty, která se v programu nemění. Konstanta se do zdrojového kódu programu zadává vždy jen jednou.

V uvedené ukázce zdrojového kódu číslo 16 vytváří příkaz Const konstantu s názvem Pi.

Zdrojový kód číslo	16:	Ukázka	použití	příkazu Const:
Const Pi As Double =	3,14	159		

Dim

Příkaz Dim, jinak zvaný dimenze, deklaruje ve zdrojovém kódu programu proměnné. Příkaz Dim požadované proměnné zajistí prostor v paměti a označí argumenty dat za běhu programu.

Zdrojový kód číslo 17: Ukázka použití příkazu Dim:

Dim ukazka As Single ukazka = 1,2

End, End<keyword>

Příkaz End zaručuje úplné uzavření všech souborů a vymazání všech proměnných programu. K ukončení celku příkazů nebo procedur slouží příkaz End<keyword>.

V případě ukázky zdrojového kódu číslo 18 příkaz End po kliknutí na tlačítko (Button) ukončí program. Kdežto v ukázce zdrojového kódu číslo 19 příkaz End uzavírá proceduru událostí společně s příkazem Sub.

Zdrojový kód číslo 18: Ukázka použití příkazu End:

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click End End Sub

Zdrojový kód číslo 19: Ukázka použití příkazu End<keyword>:

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click End Sub

If, Then, Else

Příkazem If, Then, Else vyvoláváme podmínku, která v programu musí plnit určité náležitosti. Klíčová slova If, Then, Else se překládají jako jestliže, potom, jinak. Příkazem vytváříme tak zvanou logickou podmínku, kterou tvoří logické výrazy. Logickými výrazy jsou výrazy, které nabývají pravdivých (true) nebo nepravdivých (falsh) hodnot.

Ukázka zdrojového kódu číslo 20 pro příkaz If, Then, Else vyvolává podmínku. Jestliže If bude rovno nule potom Then vypíše zprávu "ahoj kamaráde" jinak ElseIf se zpráva změní na "nejsi vítán".

```
Zdrojový kód číslo 20: Ukázka použití příkazu If, Then, Else:

Dim count As Integer = 0

Dim zprava As String

If count = 0 Then

zprava = "Ahoj kamaráde."

ElseIf count = 1 Then

zprava = "Nejsi vítán."

End If
```

Sub

Příkaz Sub vytváří definici názvu, parametrů a seskupuje veškeré příkazy do procedur. Ve zdrojovém kódu programu jej vždy najdeme na začátku a na konci procedury.

Zdrojový kód číslo 21: Ukázka použití příkazu Sub:

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click End Sub

Do, Loop

Příkazy Do, Loop slouží k vytváření cyklů. V cyklu se opakuje neustále blok příkazů, dokud podmínka nenabyde hodnoty True ve spojení s While nebo honoty True ve spojení s Until.

Zdrojový kód 22 zobrazuje použití příkazu Do, Loop v cyklu. Uživatel žádá o vstup do programu, dokud nenapíše své jméno nebo "Konec" program ho nikam nepustí.

```
Zdrojový kód číslo 22: Ukázka použití příkazu Do,Loop:

Dim vstupnijmeno As String

Do While vstupnijmeno <> "Konec"

vstupnijmeno= _

InputBox (,,Zadejte jméno nebo napište konec pro ukončení programu.")

If vstupnijmeno <> "Konec" Then TextBox1.Text = vstupnijmeno

Loop
```

Function

Mezi příkazy Function a End Function umísťujeme skupinu příkazů, která vytváří funkce. Příkaz Function v programu využíváme k zpracování textu, vstupu nebo výpočtu číselných hodnot.

Ve zdrojovém kódu číslo 24 vytváří příkaz funkci pro obsah kruhu, která následně přiřadí výsledek do identifikátoru obsahu kruhu.

Zdrojový kód číslo 23: Ukázka použití příkazu Function:
Function obsahkruhu(ByVal polomer as Single) As Single
$obsahkruhu = 3.14 * (polomer ^ 2)$
End Function

Imports

Příkaz Imports slouží k jednoduššímu odkazování na třídy tisku a pro další potřebné hodnoty v oboru názvů. Příkaz Imports se vkládá na začátek zdrojového kódu formuláře.

V ukázce zdrojového kódu číslo 25 příkaz Imports slouží k odkazování na třídu tisku.

```
Zdrojový kód číslo 24: Ukázka použití příkazu Imports:
Imports System. Drawing. Printing
```

Return

Příkaz Return umožní vrátit volající rutině jakoukoliv hodnotu. Ukázka zdrojového kódu číslo 26 vytváří příkaz Return.

```
Zdrojový kód číslo 25: Ukázka použití příkazu Return:
Function obsahkruhu(ByVal polomer as Single) As Single
Return 3.14 * (polomer ^ 2)
End Function
```

Structure

Příkazem Structure vytváříme vlastní datové typy, které nazýváme uživatelské datové typy. Příkaz Structure využijeme v případě, že pracujeme se skupinou datových položek, které vzájemně patří k sobě, ale každá z nich má jinou kategorii dat. Příkaz Structure je zakončený End Structure. Mezi začátek a konec příkazu Structure vkládáme

deklaraci příkazem Dim. Příkaz Structure se uvádí buď na začátku formuláře, nebo v modulu spolu s dalšími deklaracemi proměnných. Nikdy příkaz Structure nesmí být uveden do procedury.

V uvedené ukázce zdrojového kódu číslo 27 příkaz Structure vytváří datový typ s názvem žák.

Zdrojový kód číslo 26:	Ukázka použití příkazu Structure:
Structure žák	
Dim Jmeno As String	
Dim DatumPrijeti As Date	
End Structure	

Praktická část

5 Ukázka číslo 1 (Hádanka)

5.1 Výukový software – Úvod

V praktické části diplomové práce si vysvětlíme, jak pomocí Visual Basic lze vytvořit výukový software. Pro tvorbu výukového softwaru používáme programovací jazyk Visual Basic 2010 Expres, který je součástí verze Visual Studio 2010.

Pro diplomovou práci jsme si vytvořili výukový software, který se skládá ze čtyř ukázek. V jednotlivých částech si rozebereme postupně jednotlivé ukázky cvičení. Všechny ukázku obsahují čtyři rozbory, a to popis cvičení ve výukovém softwaru, využití ve výuce českého jazyka, Visual Basic – grafická úprava a Visual Basic – zdrojový kód.

V části popis cvičení ve výukovém softwaru se zaměříme na základní principy a funkce cvičení. Využití ve výuce českého jazyka a literární výchovy si uvedeme konkrétní příklady zaměřené na výuku českého jazyka a literární výchovy. V první části Visual Basic – grafická úprava si vysvětlíme, jak vytvořit základní formulář pro výukový software. V poslední části Visual Basic – zdrojový kód si popíšeme princip vytvoření výukového softwaru.

5.2 Popis cvičení ve výukovém softwaru

Ukázka číslo 1 (Hádanka) se skládá ze tří částí (viz obrázek číslo 1: Části cvičení). V první části se nachází rozložená hádanka, která je sestavena z devíti karet. Úkolem žáka je jednotlivé karty sestavit od jedničky po devítku. Karty se posunují kliknutím na danou kartu. Směry, kterými se jednotlivými kartami pohybuje, jsou naznačeny na obrázku číslo 2: Směry pohybu. Vždy jedna karta zůstává volná. Volná karta je vždy označena modrou barvou. Po sestavení jednotlivých karet od jedničky po devítku vznikne obrázek, na kterém se ukrývá hádanka.



Obrázek číslo 1: Části cvičení

↓ [→]	+ _ →	÷.
$\stackrel{ \uparrow }{}$	$^{\dagger}_{\downarrow} \rightarrow$	← † ↓
† →	← [†] →	↓

Obrázek číslo 2: Směry pohybu

Poté přechází žák k části dva a tři, kde se nachází otázky, které se vztahují k dané hádance na obrázku. Žák musí hádanku vykreslenou na sestaveném obrázku vyluštit, aby mohl správně odpověď na druhou část úkolu. Pro ověření chyby se za položenými otázkami zobrazí špatná odpověď (viz obrázek číslo 3: Špatná odpověď) nebo správná odpověď (viz obrázek číslo 4: Správná odpověď).





Obrázek číslo 3: Špatná odpověď

Obrázek číslo 4:Správná odpověď

5.3 Využití ve výuce českého jazyka a literární výchovy (literární výchova)

Na obrázku číslo 5: Rozložená hádanka lze vidět rozložený příklad hádanky zaměřené na výuku literární výchovy. Položené otázky v druhé a třetí části nyní

napovídají, že hádanka skrývá název literárního díla. Složením obrázku získává žák nápovědu pro název díla (viz obrázek číslo 6: Složená hádanka). V tomto konkrétním případě se objeví v hádance název literárního díla Krakatit (správná odpověď pro otázku číslo jedna) a autor literárního díla Karel Čapek (správná odpověď pro otázku číslo dvě).



 $\mathbf{K}_{4}^{1} \mathbf{k}_{8}^{2} \mathbf{k}_{9}^{3}$

Obrázek číslo 5: Rozložená hádanka

Obrázek číslo 6: Složená hádanka

Hlavní účelem hádanky je přiblížení určitých pojmů žákovi pomocí grafického znázornění. Samotná hádanka by měla zapůsobit na logickou stránku žákova učení.

5.4 Visual Basic – grafická úprava

Základním prvkem Visual Basicu je formulář, na který se postupně vkládají další ovládací prvky. Po vytvoření formuláře ve vlastnostech (properties) si zadáme název formuláře Hádanka. Dále nastavíme velikost (size) formuláře na 774, 435 pixelů. Dobrou variantou je předem si připravit několik pozadí, které nám budou utvářet grafiku daného softwaru. Pozadí do formuláře se vkládá pomocí vlastnosti backgroundimage. Chceme-li zabránit deformaci výukového softwaru v případě maximalizace, nastavíme si u vlastnosti maximizebox nepravdu (false). Posledním krokem při nastavování vlastností bude nastavení pozice při startu výukového softwaru. Pozici nastavíme přes vlastnost startpozition, kde vybereme možnost centerscreen. Při zapnutí se vždy výukový software zobrazí uprostřed plochy monitoru. Seznam Ovládacích prvků vložených do formuláře:

- > ToolStrip (1x);
- ➤ ImageList (1x);
- ➢ GroupBox (3x);
- PictureBox (11x);
- > Button (4x);

- > TextBox (2x);
- \succ Label (2x).

Ovládacímu prvku ToolStrip přiřadíme vlastnost ToolStripButton, kterou nastavíme kliknutím levého tlačítka myši přes políčko DispleyStyle na text. Objeví se tlačítko, kterému v nastavení přidělím text Hádanka1.

Formulář obsahuje tři GroupBoxy, do kterých postupně vkládáme jednotlivé ovládací prvky. GroupBoxy v ukázce č. 1 slouží k rozlišení jednotlivých částí cvičení. GroupBox1 s názvem "Poskládej obrázek" bude obsahovat ovládací prvek PictureBox 1-9. GroupBox 2 s textem "Název díla ukrytého v obrázku" se bude skládat z TextBox 1, Button 1 a PictureBox 10. Poslední GroupBox3 s názvem "Jméno autora (Jméno, Příjmení)" se bude skládat z TextBox 2, Button 2 a PictureBox 11.

Poté vytvoříme ovládací prvek PictureBox1, kterému nastavím vlastnosti velikost (size) 101,99 a barvu pozadí (backcolor) YellowGreen. PictureBox1 zkopírujeme tak, aby vzniklo devět samostatných PictureBoxů. U PicureBox 9 změníme barvu pozadí (BackColor) LightBlue. Poslední barvu měníme z důvodu, aby bylo vidět po naprogramování, že se obrázek přesouvá.

Pokud chceme následně PictureBoxu (1-9) přiřadit obrázek vytvoříme si ImageList. Do ImageList nahrajeme přes volbu ChooseImages předem připravené obrázky (k1 – k9). Velikost obrázku přizpůsobíme PictureBoxu.

Do formuláře vložíme TextBox, kterému nastavíme vlastnost písma (font) Times New Romen velikost písma 14 a poté TextBox zkopírujeme. Takto vytvoříme dva TextBoxy.

Tlačítka (Button 1-4) naformátujeme stejně jako . Tlačítka v sobě budou ukrývat jen jiný text (Button 1,2 = Vyhodnotit, Button 3 = Jak na to, Button 4 = Hlavní nabídka).

Vytvoříme PictureBox 10 - 11, kde se po provedení kontroly bude zobrazovat obrázek sp.jpg nebo šp.jpg. PictureBox 10 a 11 má pozadí (backcolor) transparent. Což znamená, že budou mít v programu průhledné pozadí.

Jako poslední ovládací prvky vložíme prvky Label 1 a 2, které se po startu programu nebudou zobrazovat. Přidělíme jim vlastnost visible false.



Obrázek číslo 7: Grafická úprava Hádanka

5.5 Visual Basic – zdrojový kód

ToolStrip jsme zvolili jako tlačítko (Button 1). Po kliknutí na tlačítko vyvoláme proceduru, která má za úkol zobrazit v PictureBox 1-9 určitý obrázek. Dále zobrazí prvky Label 1-2 s určitým textem (v našem případě Krakatit a Karel Čapek). Prvek Label 1 a 2 zůstává pro uživatele programu neviditelný. Pro zobrazení obrázku v PictureBoxu, po kliknutí na ToolStrip, přidáme do ImageListu požadované obrázky. V ImaginListu se zobrazí seznam zvolených obrázků pod čísly 0-8. Teprve teď přiřadíme k jednotlivým PictureBoxům jejich obrázek. Příkaz bude vypadat následovně PictureBox1.Image = ImageList1.Images(1). Příkazem definujeme, že v PictureBoxu se zobrazí obrázek, který je uložený v ImageListu na pozici 1. Pro prvky Label stačí nadefinovat jen text.

Zdrojový kód číslo 27: Vložení obrázků do PictureBoxu:

Private Sub ToolStripButton1_Click(sender As System.Object, e As System.EventArgs) Handles ToolStripButton1.Click

PictureBox1.Image = ImageList1.Images(1) PictureBox2.Image = ImageList1.Images(0) PictureBox3.Image = ImageList1.Images(2) PictureBox4.Image = ImageList1.Images(6) PictureBox5.Image = ImageList1.Images(3) PictureBox6.Image = ImageList1.Images(7) PictureBox7.Image = ImageList1.Images(4) PictureBox8.Image = ImageList1.Images(5) PictureBox9.Image = ImageList1.Images(8) End Sub Zdrojový kód číslo 28: Přiřazení hodnot Label 1 a 2: Label1.Text = "Krakatit" Label2.Text = "Karel Čapek"

Teď se zaměříme na část "Poskládej obrázek". V tomto případě si přejeme, aby se některý z PictureBoxů přesunul, když se na něj klikne. Schéma přesunu PictureBoxů uvádíme na obrázku číslo 2 v části popis výukového softwaru. Z obrázku je zjevné, že stačí obrázek přesunout jen na pozice, které určují šipky. Tedy pokud klikneme na PictureBox 1 žák obrázek může přesouvat na pozici PictureBox 2 a PictureBox 3. PictureBoxu 1 definujme cestu přes příkaz If. Program tvoří, před kliknutím na ToolStriptButton1 (Hádanka číslo 1), barevné pozadí PictureBoxů 1-9, vytvořila jsme pohyby i pro barevné pozadí. Jak postupovat při psaní programovacího jazyka pro přesouvání jednotlivých PictureBoxů 1-9 je ukázáno ve zdrojovém kódu číslo 30.

Zdrojový kód číslo 29: Ukázka programování pro přesouvání jednotlivých PictureBoxů:
If PictureBox2.BackColor = Color.LightBlue Then
PictureBox2.Image = PictureBox1.Image
PictureBox2.BackColor = Color.YellowGreen
PictureBox1.Image = ImageList1.Images(8)
PictureBox1.BackColor = Color.LightBlue
End If

Pro tlačítka (Button 1 a 2) zvolíme příkaz If, který bude sloužit ke kontrole správnosti textového pole (TextBox 1 a 2). Pro zjištění správného zadání textového pole (TextBox 1 a 2) přimějeme příkaz, aby zkontroloval text v textovém poli (TextBox 1 a 2), jestli se shoduje s textem u prvku Label1 a 2. Pokud se textové pole (TextBox 1 a 2) a Label1 a 2 shodují, objeví se v PictureBoxu10 a 11 obrázek sp.jpg. Pokud se textové pole a Label1 a 2 neshodují, stačí si jen zvolit, že po kliknutí na tlačítko (Button1 a 2) se zobrazí v PictureBoxu (1 a 2) obrázek šp.jpg. Obrázky sp a šp nalezneme v Resources. Důležité je nadefinovat cestu k obrázkům. Cesta k obrázku ve Visual Basic vypadá následovně: My.Resources.šp. Cestu k obrázku umístíme do závorek.

Zdrojový kód číslo 30: Pro vyhodnocení cvičení: Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click PictureBox10.Image = (My.Resources.šp) If Label1.Text = "Krakatit" And TextBox1.Text = "Krakatit" Then PictureBox10.Image = (My.Resources.sp) End Sub

Zdrojový kód číslo 31: Pro vyhodnocení cvičení:

Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click

PictureBox11.Image = (My.Resources.šp)

If Label2.Text = "Karel Čapek" And TextBox2.Text = "Karel Čapek" Then PictureBox11.Image = (My.Resources.sp) End Sub

6 Ukázka číslo 2 (Skládání textů)

6.1 Popis cvičení ve výukovém softwaru

Cvičení v ukázce číslo 2 se skládá ze tří částí (viz obrázek číslo 9). V první části se nachází zadání cvičení, které se zobrazuje na čtyřech samostatných kartách. Principem cvičení je přemístit čtyři dané karty z první části cvičení do druhé části cvičení tak, aby po přemístění dávaly smysl. Přemístění jednotlivých karet probíhá tak, že žák pod danou kartu, do volného políčky uvede číslo místa z druhé části cvičení, kam chce kartu přenést. Po stisknutí tlačítka kontrola se na přenesených kartách zobrazí správná nebo špatná odpověď. Kartu lze vrátit zpět do první části pomocí tlačítka zpět, které se nachází pod původní pozicí karty.

Poslední třetí část cvičení může sloužit pro pokládání otázky, která se vztahuje k druhé části cvičení.



Obrázek číslo 8: Ukázka cvičení skládání textu

6.2 Využití ve výuce českého jazyka a literární výchovy (literární výchova)

V první části cvičení se zobrazí přeházený text z literárního díla (viz obrázek číslo 10). V tomto konkrétním případě text z literárního díla Karla Čapka. Správným seřazením karet na pozici 1 - 4, žák vytvoří souvislý text ukázky z literárního díla Karla Čapka. Po přečtení správně seřazeného textu, by měl být žák schopný poznat název literárního díla. V třetí části cvičení žák odpovídá na název literárního díla, které se skrývá v ukázce.



Obrázek číslo 9: Vložený text

6.3 Visual Basic – grafická úprava

Po vložení formuláře nastavíme jeho velikost (size) na 1122, 772 pixelů. Formulář přejmenujeme na Skládání textů. Do vlastnosti start position, maximizebox a background zadáme stejné parametry jako u ukázky číslo 1.

Seznam ovládacích prvků vložených do formuláře:

- ToolStrip (1x);
- ➤ ImageList (1x);
- PictureBox (4x);
- > Panel (4x);
- > TextBox (5x);
- > Button (12x);
- ➤ Label (10x).

Stejným způsobem jako u ukázky číslo 1 nastavíme ToolStrip tak, aby fungoval jako tlačítko. Tlačítku dáme název Karel Čapek. Ovládací prvek ToolStrip využíváme k tomu, abychom mohli do výukového softwaru vložit více cvičení založených na stejném principu s jiným zadáním.

Dále si vytvoříme ImageList do kterého vložíme čtyři obrázky. Na těchto obrázcích se skrývá úryvek z literárního díla. Každý obrázek představuje jednu část úryvku z literárního díla. Po stisknutí tlačítka s názvem Karel Čapek na ToolStripu z ImaginListu zobrazí obrázky 0-3 do PictureBoxu 1-4.

Pro vytvoření PictureBoxů 1-4 stačí nejprve vložit jeden PictureBox, který si zkopírujeme až po jeho naformátování. Velikost PictureBoxů bude 256, 256 pixelů. Na pozadí vložíme předem vytvořený obrázek ve PhotoFiltru přes vlastnost BackgroundImage.

Naproti PictureBoxů 1-4 vložíme čtyři Panely 1-4. Panel 1-4 bude sloužit k označení místa přesunu obrázku z PictureBoxů 1-4. Panel 1-4 má stejnou velikost jako PictureBox 1-4. Barva pozadí (BackColor) je tak zvaně transparentní, což znamená, že je průhledná. BorderStyle (rámeček panelu) má vlastnost Fixed3D.

Do formuláře dále vložíme TextBox. TextBox 1-4 bude sloužit pro čísla panelů, do kterých bude obrázek následně přenesen. Styl písma pro TextBox 1-4 zvolíme Times New Roman, řez písma tučný a velikost písma 20. Barvu písma nastavíme ve vlastnostech ForeColor jako barva DeepSkyBlue. TextBox 5 bude sloužit pro odpověď na otázku, která je v něm zobrazená. V tomto konkrétním případě název díla.

Tlačítka (Button 1-12) jsou rozdělena podle jejich funkce. Button 1-4 slouží k přesunu obrázku a nese název Přesunout. Button 5-8 s názvem zpět slouží k vrácení obrázku na původní pozici. Tlačítko Button 9 a 10 slouží ke kontrole správnosti cvičení. Button 11 v sobě ukrývá nápovědu, jak pracovat s cvičením a Button 12 má funkci pro vrácení do hlavní nabídky. Tlačítka jsou všechna naformátována stejně, liší se jen velikostí. Tlačítka mají nastavenou ve vlastnostech barvu Yellow Green.

Posledním ovládacím prvkem ve výukovém softwaru jsou Label 1-10. Label 1 ponese ve zdrojovém kódu název Válka s mloky a bude mít vlastnost Visible false. Což znamená, že po spuštění cvičení jej žák neuvidí. Label 2-5 budou sloužit jenom pro zobrazení čísel v panelech. Tato čísla slouží pro určení pozice přenosu, která se zapisuje do TextBoxu 1-4. Ostatní Label 6-10 slouží pro ověření správnosti cvičení. Label 6-10 se zobrazí až po kontrole splněného cvičení. Skrývají v sobě odpověď správně nebo špatně.

6.4 Visual Basic – zdrojový kód

ToolStrip je opět zvolený jako tlačítko (Button1), u kterého vyvoláme proceduru ToolStripButton1_Click. Procedura bude obsahovat příkaz, který do PictureBoxu 1-4 umístí z ImageListu obrázky (viz zdrojový kód čísla 33). Procedura bude následně obsahovat také definici Label1, ve kterém se bude zobrazovat text zapsaný v uvozovkách válka s mloky.



Obrázek číslo 10: Ukázka přesouvání textu



Pro nastavení tlačítek přesun (Button 1-4) vytvoříme čtyři stejné procedury, které budou mít totožné příkazy. Proceduru vyvoláme po kliknutí na jednotlivá tlačítka (Button 1-4). Pro příkaz If bude platit, pokud v TextBox 1-4 bude uvedeno číslo 1-4 přesune se PictureBox 1-4 na jinou pozici. Přesun pozic je určený jednotlivými Panely 1-4. Tedy je-li TextBox roven číslu jedna PictureBox se přesouvá pomocí příkazu Location na určenou pozici. Dále v proceduře uvedeme možnost prázdného TextBox 1-4. V takovém případě zůstává daný obrázek na svém místě.

Zdrojový kód číslo 33: Určení pozice obrázku:

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click If TextBox1.Text = "1" Then PictureBox1.Location = New Point(563, 54) If TextBox1.Text = "2" Then PictureBox1.Location = New Point(844, 54) If TextBox1.Text = "3" Then PictureBox1.Location = New Point(563, 342) If TextBox1.Text = "4" Then PictureBox1.Location = New Point(844, 342) TextBox1.Text = """ End Sub

U tlačítka zpět (Button 5-8) ve zdrojovém kódu zadáme u jednotlivých obrázků už jen jednu pozici. Tedy po kliknutí na tlačítko zpět se obrázek vrátí do původní pozice. Opět vytvoříme příkaz Location.

Zdrojový kód číslo 34: Vrácení pozice:

Private Sub Button5_Click(sender As System.Object, e As System.EventArgs) Handles Button5.Click PictureBox1.Location = New Point(12, 28) End Sub

U prvního tlačítka zkontrolovat je nutné brát v úvahu pozice obrázku a zadaný text v Label 1. V příkazu If bude zaznamenáno, pokud se správná pozice obrázku (PictureBox.Location) rovná Label 1, tak se Label 6-9 vykreslí jako správná odpověď. V případě že se nebude Label 6-9 rovnat správné odpovědi vypíše se Label 6-9 jako špatná odpověď. U druhého tlačítka zkontrolovat bude platit, jestliže je TextBox shodný s Label 1 zobrazí se v Label 10 správná odpověď. Pokud Label 10 nebude mít správnou odpověď opět se v Label 10 zobrazí špatná odpověď.

Zdrojový kód číslo 35: Určení správnosti pozice obrázku:

Private Sub Button9_Click(sender As System.Object, e As System.EventArgs) Handles Button9.Click If PictureBox1.Location = New Point(844, 342) And Label1.Text = "Válka s mloky" Then Label12.Text = "Správně" If Label9.Text <> "Správně" Then Label9.Text = "Špatně" End Sub

Zdrojový kód číslo 36: Kontrola správnosti odpovědi na otázku:

Private Sub Button10_Click(sender As System.Object, e As System.EventArgs) Handles Button10.Click

If TextBox5.Text = "Válka s mloky" And Label1.Text = "Válka s mloky" Then Label2.Text = "Správně"

If Label2.Text <> "Správně" Then Label2.Text = "Špatně"

End Sub

7 Ukázka číslo 3 (Poslech textu)

7.1 Popis cvičení ve výukovém softwaru

Cvičení se dělí na dvě části, které na sebe navazují (viz obrázek číslo 12). V první části cvičení se nachází zvukový záznam. Zvukový záznam se přehraje po kliknutí na tlačítko s obrázkem přehrávání. V druhé části se nachází možnosti, které navazují na přehrávaný zvukový záznam. Úkolem je zatrhnout správnou odpověď v druhé části cvičení. Výsledek odpovědi se zobrazí po vyhodnocení pod zatrhnutou odpovědí.



Obrázek číslo 11: Ukázka cvičení poslech textu

7.2 Využití ve výuce českého jazyka a literární výchovy (literární výchova)

Zvukové záznamy jsou zaměřeny na výuku literární výchovy. Konkrétně se zvukové záznamy v cvičení zaměřují na Karla Jaromíra Erbena a jeho sbírku Kytice. Pro zvukové záznamy jsme si vybrali šest balad, které se nacházejí ve sbírce Kytice (Štědrý den, Svatební košile, Vodník, Polednice, Zlatý kolovrat, Kytice). Úkolem žáka je si vybrané balady poslechnout a poté správně určit o jakou baladu se jedná. Správnou baladu žák zatrhne pod daným zvukovým záznamem.

7.3 Visual Basic – grafická úprava

Zvolený formulář bude mít velikost (size) 800, 600 pixelů a název poslech textu. Do vlastnosti start position, maximizebox a background zadáme stejné parametry jako u ukázky číslo 1.

Seznam ovládacích prvků vložených do formuláře:

 \succ Label (1x);

- ➢ GroupBox (6x);
- > Button (8x);
- RadioButton (36x);
- PictureBox (6x).

Vložený Label 1 obsahuje ve vlastnostech text přesné zadání cvičení (podle zvukového záznamu určete baladu z knihy Karla Jaromíra Erbena Kytice). Pro Label 1 nastavíme zelenou barvu písma (ForeColor).

Dále do formuláře vložíme GroupBox, který bude sloužit jako rámeček jednotlivým částem cvičení. Každému GoupBoxu 1-6 přidělíme ve vlastnostech text ukázka č. 1-6.

Vložená tlačítka (Button 1-6) nebudou obsahovat text, ale obrázek, který bude znázorňovat spuštění zvukového záznamu. Obrázek vložíme do tlačítek pomocí vlastnosti BackgroundImage. Button 7 slouží k vyhodnocení cvičení. Ve vlastnostech nastavíme pro tlačítko vyhodnocení černé pozadí (BackColor) a bílé písmo (ForeColor). Button 8 nazvu ve vlastnostech text hlavní nabídka.

Pod tlačítka vložíme zaškrtávací tlačítko (RadioButton 1-36). Vždy šest zaškrtávacích tlačítek ponese stejný název, který upravíme ve vlastnostech text.

Pod RadioButton ještě přidáme PictureBox 1-6. V PictureBoxu 1-6 se bude po vyhodnocení zobrazovat obrázek pro správnou nebo špatnou odpověď.

7.4 Visual Basic – zdrojový kód

Pro každé tlačítko (Button 1-6) zadáme do zdrojového kódu proceduru Button_Click. Do procedury Button_Click zaznamenáme příkaz pro přehrání a načtení zvukového záznamu. Zvukový záznam spustíme pomocí příkazu My.Computer.Audio.Play. U příkazu My.Computer.Audio.Play musíme do závorky zaznamenat cestu k daném zvukovém záznamu. Jinak nebude výukový software schopný najít daný zvukový záznam.

Zdrojový kód číslo 37: Vložení zvukové ukázky:

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

My.Computer.Audio.Play(My.Resources.vodnik, AudioPlayMode.Background) End Sub Po přiřazení jednotlivých zvukových záznamů potřebujeme u každého zvukového záznamu vědět, pod jakým zaškrtávacím tlačítkem se skrývá správná odpověď. Poté můžeme napsat zdrojový kód pro zobrazení správné a špatné odpovědi. V případě, že odpověď bude správná, zobrazí se v PictureBoxu obrázek fajfky. Pokud bude odpověď špatná, zobrazí se v PictureBoxu obrázek křížku. Pro zobrazení správné odpovědi využijeme příkaz If. Příkaz If definuje zatrhnutou správnou odpověď jako obrázek fajfky a zatrhnutou špatnou odpověď jako obrázek křížku.

Zdrojový kód číslo 38: Určení správnosti zaškrtávacího políčka:

PictureBox1.Visible = True

PictureBox1.Image = (My.Resources.šp)

If RadioButton3.Checked = True Then PictureBox1.Image = (My.Resources.sp) If RadioButton11.Checked = True Then PictureBox2.Image = (My.Resources.sp) If RadioButton13.Checked = True Then PictureBox3.Image = (My.Resources.sp) If RadioButton20.Checked = True Then PictureBox4.Image = (My.Resources.sp) If RadioButton30.Checked = True Then PictureBox5.Image = (My.Resources.sp) If RadioButton34.Checked = True Then PictureBox6.Image = (My.Resources.sp) 'RadioButton1.Checked = False 'RadioButton2.Checked = False 'RadioButton4.Checked = False 'RadioButton5.Checked = False 'RadioButton6.Checked = False

8 Ukázka číslo 4

8.1 Popis cvičení ve výukovém softwaru

Cvičení se dělí na pět části, které jsou vzájemně propojené (viz obrázek číslo 14). První a druhá část vytváří zadání cvičení. V poslední třetí části cvičení žák odpovídá na zadání s první a druhé části. První část tvoří otázka, na kterou žák hledá odpověď. V druhé části se nachází obrázek, který slouží jako nápověda k zadané otázce. Třetí část tvoří seznam možností na danou odpověď.

Žák musí rozbalit nabídku, která je zadána v třetí části a podle zadané otázky odpovědět. Ve čtvrté části se zobrazí správná nebo špatná odpověď. Na čtvrtou část navazuje pátá část, která vyhodnotí počet správných a špatných odpovědí v cvičení.



Správně 5. část Špatně:

Obrázek číslo 12: Ukázka cvičení

8.2 Využití ve výuce českého jazyka a literární výchovy (mluvnice)

Cvičení slouží k procvičování párových vyjmenovaných slov, které podle významu mohou obsahovat měkké i nebo tvrdé y (výr x vír). V první části se zobrazuje věta s vynechaným slovem. Ve druhé části se vykreslí obrázek, který představuje činnost nebo věc, která patří na vynechané místo ve větě. Ve třetí části žák rozklikne nabídku, kde si může vybrat konkrétní slovo, které patří do zadané věty.

Po vyhodnocení správných a špatných odpovědí se žákovi zobrazí počet správných a špatných odpovědí. V případě správné odpovědi se pod nabídkou objeví nápis správně. V případě špatné odpovědi žákovi výukový software napoví, jakou udělal chybu. V tomto konkrétním případě se objeví nápis dané slovo je vyjmenovaným slovem nebo není vyjmenovaným slovem.

8.3 Visual Basic – grafická úprava

Zvolený formulář bude mít velikost (size) 1261, 576 pixelů a název vyjmenovaná slova (týká se konkrétního příkladu). Do vlastnosti start position, maximizebox a background zadáme stejné parametry jako u ukázky číslo 1. Seznam ovládacích prvků vložených do formuláře:

- ➢ GroupBox (10x);
- \blacktriangleright PictureBox (10x);
- \succ Button (2x);
- ➢ ComboBox (10x);
- ➤ Label (25x).

GroupBox 1-10, který jsme vložili do formuláře, slouží pro vytvoření rámečku pro jednotlivá zadání cvičení. V názvu GroupBoxu 1-10 bude konkrétní zadání cvičení (př. Na hladině se objevil _____.).

Do GroupBoxu přidáme PictureBox, kterému ve vlastnostech přiřadíme pořadí daného obrázku, který patří k zadání cvičení. U příkladu "Na hladině se objevil ." bude na obrázku zobrazení vodní vír.

Pod PictureBox vložíme ComboBox, který bude sloužit pro zadání odpovědí. Nabídku odpovědí do ComboBoxu vytvoříme až ve zdrojovém kódu. Každému ComboBoxu ve vlastnostech zadáme zelenou barvu pozadí a bílé písmo textu v nabídce.

Ve formuláři ukázky číslo 4 se nachází jen dvě tlačítka (Button). První tlačítko slouží ke kontrole cvičení a druhé tlačítko pro návrat do hlavní nabídky. Tlačítka (Button 1-2) mají stejné vlastnosti. Barva pozadí (BackColor) tlačítka je zelená a barva písma (ForeColor) bílá. Na vzhled písma (Font) jsme využili Comic Sans MS s tučným řezem písma.

Do formuláře je vloženo velké množství ovládacího prvku Label. Label 1 slouží jako nápověda jak pracovat s cvičením. Label 3 a 4 slouží pro vytyčení místa počtu správných a špatných odpovědí. Label 5 a 6 vykreslí po kontrole počet správných a špatných odpovědí. Zbylých dvacet ovládacích prvků Label se zobrazuje pod nabídkou (ComboBox 1-10), kde zobrazí buď nápis správné nebo jestli je slovo vyjmenované nebo není vyjmenované.

8.4 Visual Basic – zdrojový kód

Ve zdrojovém kódu nejprve zadáme obsah ComboBoxu 1-10, který se načte po zobrazení formuláře. Aby se zadání ComboBoxu zobrazilo při otevření formuláře ukázky číslo 4, vyvoláme proceduru Form4_Load. Do procedury Form4_Load následovně vložíme příkaz pro vytvoření nabídky do ComboBoxu 1-10. Slova, která se vkládají do nabídky, se vždy vpisuji do kulatých závorek a uvozovek (viz. Zdrojový kód číslo 39: Definice ComboBoxu).



Obrázek číslo 13: Ukázka hotového cvičení

Zdrojový kód číslo 39: Definice ComboBoxu:
Private Sub Form4_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
ComboBox1.Items.Add("MY")
ComboBox1.Items.Add("MI")
ComboBox1.Items.Add("VÍR")
ComboBox1.Items.Add("VÝR")
ComboBox1.Items.Add("MÍT")
ComboBox1.Items.Add("MÝT")
ComboBox1.Items.Add("SLEPÍŠ")
ComboBox1.Items.Add("SLEPÝŠ")
ComboBox1.Items.Add("LÍSKA")
ComboBox1.Items.Add("LYSKA")
End Sub

Pro výpočet počtu správných a špatných chyb použijeme proměnnou Dim s datovým typem Integer. Zvolíme hned několik proměnných Dim a to proměnnou vysledek a písmeno a – j. Dále nadefinujeme příkaz If. U příkazu If bude platit pokud ComboBox obsahuje správnou odpověď pak se proměnná a - j rovná jedné. V případě, že obsahuje ComboBox špatnou odpověď bude se proměnná a - j rovnat nule. Proměnnou vysledek definujeme jako součet proměnných a – j. Potom tedy můžeme definovat, že Label 6 se rovná vysledek (součet správných odpovědí) a Label 5 se rovná 10 - vysledek (10 položených otázek mínus počet správných odpovědí).

Zdrojový kód číslo 40: Výpočet správných a špatných odpovědí: Dim vysledek As Integer Dim a As Integer Dim b As Integer Dim c As Integer Dim d As Integer Dim ee As Integer Dim f As Integer Dim g As Integer Dim h As Integer Dim i As Integer Dim j As Integer If ComboBox1. Text = "VIR" Then a = 1 Else a = 0If ComboBox2.Text = "SLEPÝŠ" Then b = 1 Else b = 0If ComboBox3.Text = "LÍSKA" Then c = 1 Else c = 0If ComboBox4.Text = "MI" Then d = 1 Else d = 0If ComboBox5.Text = "SLEPIŠ" Then ee = 1 Else ee = 0If ComboBox6.Text = "LYSKA" Then f = 1 Else f = 0If ComboBox7.Text = "MÝT" Then g = 1 Else g = 0If ComboBox8. Text = "MÍT" Then h = 1 Else h = 0*If ComboBox9.Text* = "VÝR" *Then i* = 1 *Else i* = 0 If ComboBox10.Text = "MY" Then j = 1 Else j = 0vysledek = a + b + c + d + ee + f + g + h + i + j*Label6.Text* = *vysledek* Label5.Text = 10 - výsledek

Pro ComboBox 1-10 platí ještě jeden příkaz If, který se vztahuje k zobrazení správné odpovědi pod daný ComboBox 1-10. U tohoto příkazu If platí, že pokud je v ComboBoxu 1-10 vybrána správná odpověď zobrazí se v Label text správně. V případě, že je odpověď chybná zobrazí se v Label možnost, že slovo je vyjmenované nebo není vyjmenované.

```
Zdrojový kód číslo 41: Definice Label podle správnosti vyplnění úkolu:

If ComboBox1.Text = "VÍR" Then Label2.Text = "Správně"

If ComboBox1.Text <> "VÍR" Then Label7.Text = "Vír není vyjmenované slovo"
```

9 Hlavní nabídka výukového softwaru

V kapitole se zaměříme na úvodní formulář výukového softwaru, který nese název Program. Program slouží jako rozcestník mezi jednotlivými cvičeními. Každé cvičení má tlačítko s názvem hlavní nabídka, které odkazuje na formulář s názvem Program.

9.1 Program – grafická úprava

Nejprve si vytvoříme formulář Form1, kterému ve vlastnostech (properties) přidělíme název Program. Dále nastavíme velikost (size) formuláře na 300, 300. Velikost se uvádí v pixelech. U svého programu chceme zakázat maximalizaci programu. Maximalizaci zakážu ve vlastnostech MaximizeBox, kde zadáme vlastnost false.

Posledním krokem při nastavování vlastností formuláře s názvem Program bude nastavení pozice při startu programu. Pozici nastavíme přes Startpozition a vybereme vlastnost CenterScreen. Při zapnutí programu se vždy program zobrazí uprostřed monitoru. Barvu pozadí nastavíme v Backcolor, kde pozadí formuláře přidělíme bílou barvu (White).

Do formuláře (Program) vložíme pět tlačítek (Button). Button 1, 3, 5, 6 přidělíme ve vlastnostech text hádanka - literatura (Button 1), skládání textu – literatura (Button3), poslech textu - literatura (Button 5), Vyjmenovaná slova (Button 6). Tlačítkům nastavíme velikost (Size) 200, 40. Poslední dvě tlačítka (Button 2,4) ponesou název O programu a Konec. Tlačítka budou odpovídat velikosti (Size) 125,30. Následně označíme všechny tlačítka a upravím barvu pozadí (Backcolor) na LightBlue a písmo (Font) na Times New Roman velikost písma 12.



Obrázek číslo 14: Program

9.2 Program – zdrojový kód

Nejprve si pro všechny tlačítka (Button 1-6) nadefinuji proceduru Button_Click, kterou vyvoláme klepnutím na jednotlivá tlačítka. Tlačítka Button 1,3,5,6 budou mít za úkol přejít na ostatní formuláře (Form2, Form3, Form4) v programu. Tlačítko O programu ponese informace, které se týkají programu a poslední tlačítko (Konec) ukončí program.

Pro přechod na ostatní formuláře programu přidělíme proceduře Button_Click příkaz Show, před který zadáme popis formuláře, na který se chceme pomocí tlačítka přesunout. Po provedení operací příkaz vypadá takto Form2.Show (). Pro zobrazení vždy jen jednoho formuláře, přidáme do zdrojového kódu procedury Button_Click příkaz Me.Hide(). Pokud nepoužijeme příkaz Me.Hide() zůstane zobrazený i původní formulář.

Zdrajavý kód 12. Dro tlažítka (Dutton 1.2.5).
Zurojovy kou 42: Pro tiacitka (Button 1,5,5):
Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
Form2.Show()
Me.Hide()
End Sub
Private Sub Button3_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click
Form3.Show()
Me.Hide()
End Sub
Private Sub Button5_Click(sender As System.Object, e As System.EventArgs) Handles Button5.Click
Form5.Show()
Me.Hide()
End Sub
Private Sub Button6_Click(sender As System.Object, e As System.EventArgs) Handles Button5.Click
Form4.Show()
Me.Hide()
End Sub

Po stisknutí tlačítka O programu se otevře tzv. dialogové okno, které se zobrazí současně s formulářem Program. Pokud budeme chtít začít pracovat opět s formulářem, musíme dialogové okno zavřít.

Pro zobrazení dialogového okna používáme funkci MsgBox(), kde do závorek nadefinuji co se má po otevření dialogového okna zobrazit. Na obrázku č. vidíme, že text se zobrazuje pod sebou. Za text, který chci přesunout na další řádek, zadáme & vbCrLf &. Mezeru mezi texty vytvoříme vsunutím & vbTab &.

Zdrojový kód 43: Dialogové okno:

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click MsgBox("Autor:" & vbTab & vbTab & "Bc. Karolína Šmatlová" & vbCrLf _ & "Rok:" & vbTab & vbTab & "2015/2016" & vbCrLf _ & "Účel programu:" & vbTab & "Diplomová práce" & vbCrLf,) End Sub

Pro ukončení programu kliknutím na tlačítko, přidělíme do procedury Button_Clic příkaz End.

Zdrojový kód 44: Konec programu: Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click End End Sub

Závěr

Informační a komunikační technologie se staly nedílnou součástí našich životů, a tak vstoupili i do vzdělávání dětí. Využívání informačních a komunikačních technologií při výuce ve školách je v dnešní době zcela běžná záležitost. Zajímavou možností se stalo využití výukových softwarů. V diplomové práci jsme se zaměřili na možnosti výukových software a možnost vytvoření vlastního výukového software.

Hlavním cílem diplomové práce je vytvořit vlastní ukázky výukového software a popsat možnosti přístupu k tvorbě výukového softwaru v aplikaci Visual Basic, který bude určený pro výuku českého jazyka a literární výchovy. Vytvořili jsme výukový software, který se skládá z ukázek cvičení. Cvičení jsou zaměřena na vizuální vnímání žáků. Především jsme se zaměřili na výuku literární výchovy. Chtěli jsme vytvořit ukázky cvičení, které by pomáhali při výuce, a tak doplňovali výkladovou část hodiny. Cvičení jsme vytvářeli za pomocí programovacího jazyka Visual Basic 2010. Představili jsme si využití cvičení v hodinách a poté jsme zpracovali podrobný popis jak cvičení vytvořit.

Dílčím cílem práce je analyzovat dostupný výukový software pro český jazyk a literární výchovu. Dříve než jsme se začali zabývat výukovými softwary pro český jazyk a literární výchovu definovali jsme si výukový software a představili jsme si jeho funkce. Rozebrali jsme si dostupné výukové softwary do výuky českého jazyka a literární výchovy od předních českých firem (např. Silcom, Terasoft, nakladatelství Tobiáš,...). Zjistili jsme, že převážná většina výukových software se zabývá českým jazykem především mluvnicí.

Dalším dílčím cílem bylo představit si možnosti programovacích jazyků a analyzovat strukturu programovacího jazyka Visul Basic. Na trhu existuje velký výběr programovacích jazyků a každý přináší jiné možnosti programovaní. Především se liší v struktuře zdrojového kódu. Záleží jen na uživateli, který programovací jazyk ovládá. Do diplomové práce jsme si nakonec vybrali programovací jazyk Visual Basic 2010. Nejprve jsme si programovací jazyk představili na teoretické úrovni a na základě toho jsme zpracovali hlavní cíl diplomové práce. Kdy jsme za jeho pomocí vytvořili zmíněné ukázky cvičení.

Zapojení výukového softwaru do výuky českého jazyka může přispět k motivaci žáka a lepšímu pochopení probíraného učiva. Výukový software by měl být jen doplněním výkladu učiva a neměl by se stát základní podstatou hodiny.

Literatura

- BURIANOVÁ, Eva. Matematický a výukový software. Vyd. 1. Ostrava: Ostravská univerzita, 2003, 74 s. Systém celoživotního vzdělávání Moravskoslezska. ISBN 80-704-2867-8.1.
- BURIANOVÁ, Eva. Využití aplikačních programu ve výuce. Vyd. 1. Ostrava: Ostravská univerzita, 2003, 146 s. Systém celoživotního vzdělávání Moravskoslezska. ISBN 80-704-2858-9.
- ČAPEK, K. Válka s mloky. 2. vydání. Praha: Leda: 2009. 274 s. ISBN 978-80-7335-399-5.
- ČERNOCHOVÁ, Miroslava a Tomáš KOMRSKA. Využití počítače při vyučování: náměty pro práci dětí s počítačem. Praha: Portál, 1998. ISBN 80-7178-272-6.
- DOSTÁL, Jiří. *Počítač ve vzdělávání*. Vyd. 1. Olomouc: Votobia Olomouc, 2007, 2 sv. ISBN 978-80-7220-295-912.
- DOSTÁL, Jiří. Výukové programy. 1. vyd. Olomouc: Univerzita Palackého v Olomouci, 2011, 67 s. ISBN 978-80-244-2782-9.
- DOSTÁL, Jiří. Výukový software a počítačové hry nástroje moderního vzdělávání. Journal of Technology and Information Education. 2009, Olomouc -EU, Univerzita Palackého, Ročník 1, Číslo 1, s. 24 - 28. ISSN 1803-537X (print). ISSN 1803-6805.
- ERBEN, K. J. Kytice. 2. vydání. Český Těšín: Infiniti art, s. r. o.: 2016. 133 s. ISBN 978-80-87491-03-4.
- HALVORSON, Michael. *Microsoft Visual Basic 2010: Krok za krokem*. 1. vydání. Brno: Computer Press, 2012. ISBN 978-80-251-3146-6.

- HANÁK, Jan. Programování v jazyce Visual Basic 2010.
 vydání. Kralice na Hané: Computer Media, 2011.
 ISBN 978-80-7402-12-1.
- KLEMENT, M. Základy programování v jazyce Visual Basic. 1. vydání. Olomouc: Univerzita Palackého, 2008. ISBN 978-80-244-2177-3.
- 12) KOCICH, Pavel a Ondřej SPILKA. 1001 tipů a triků pro Micrososft Visual Basic.Vyd. 1. Brno: Computer Press, 2010, 520s. ISBN 978-80-251-211-4.
- 13) KROPÁC, Jiří a Miroslav CHRÁSKA. Výchova v obecně technických předmětech.
 1. vyd. Olomouc: Univerzita Palackého v Olomouci, 2004, 190 s. ISBN 80-244-0897-X.
- 14) MAŠEK, Jan,Petr MICHALÍK a Václav VRBÍK. Otevřené technologie ve výuce.
 Plzeň: Západočeská univerzita v Plzni, 2004. ISBN 80-7043-254-3.
- PRUCHA, Jan, EliškaWALTEROVÁ a Jiří MAREŠ. *Pedagogický slovník*. 6., rozš. a aktualiz. vyd. Praha: Portál, 2009, 395 s. ISBN 978-807-3676-476.
- ROZEHNAL, P. Multimédia a jejich využití ve výuce. 1. vydání. Ostrava: Vysoká škola báňská Technická univerzita Ostrava: 2010. ISBN 978-80-24-2182-5.
- 17) SHELDON, Bill et al. *Progessional Visual Basic 2010 and .NET 4*. Indianapolis, Ind.: Wiley Pub., c2010, 1276 p. ISBN 04-705-0224-X.
- 18) ZAK, Diane. Clearly visual basic: programming with Microsoft Visual Basic 2010.2nd Ed. Boston, MA: Cngage Learning, 2011, p. 572. ISBN 978-111-1530-150.
- ERBEN, J. K. Kytice [zvukový záznam na CD]. Čte Jiří Štědroň. Praha: Vydavatelství SOCHA, 2015.

Internetové zdroje

- Abeceda pro děti výukové aplikace pro děti (Android, iOS, Windows). 302 Found [online]. Copyright © PMQ software 2013. Dostupné z: <u>http://www.pmq-software.com/sw/cz/android/abeceda-pro-deti/</u>
- Ajťácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další... *itnetwork.cz Ajťácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další*. [online]. Copyright © 2017 itnetwork.cz.Veškerý obsah webu. Dostupné z: <u>http://www.itnetwork.cz</u>
- Comenia Script výuková aplikace pro tablety, telefony a interaktivní tabule.
 [online]. Copyright © 2014 J4W. Dostupné z: <u>http://comeniascriptapp.cz/cs</u>
- 4) Český jazyk pravopis. 302 Found [online]. Copyright © PMQ software 2013.
 Dostupné z: <u>http://www.pmq-software.com/sw/cz/android/cesky-jazyk-pravopis/</u>
- 5) Diktatorek.cz [online]. Dostupné z: https://www.diktatorek.cz
- 6) Fakulta informatiky Masarykovy univerzity [online]. Dostupné z: https://www.fi.muni.cz/usr/jkucera/pv109/2003p/xskriva1.historie.htm
- 7) Jak psát web. [online]. Dostupné z: http://jakpsatweb.cz
- 8) Java portál. Java portál [online]. Dostupné z: http://java.cz
- 9) Největší český web zaměřený na .NET framework. Největší český web zaměřený na .NET framework [online]. Copyright © 2017 [cit. 12.06.2017]. Dostupné z: http://www.dotnetportal.cz
- 10) Terasoft, a.s.. Terasoft, a.s. [online]. Dostupné z: http://www.terasoft.cz/index2.htm
- Tobiáš učebnice Výukové programy. Tobiáš učebnice [online]. Copyright
 2017 Tobiáš. Dostupné z: http://www.tobias-ucebnice.cz/vyukove-programy

- 12) Tvorba webu. [online]. Dostupné z: http://www.tvorba-webu.cz
- 13) Visual Basic. Learn to Develop with Microsoft Developer Network / MSDN [online]. Copyright © 2017 Microsoft. Dostupné
 z: https://msdn.microsoft.com/cs-cz/library/2x7h1hfk.aspx
- 14) Výukové programy a výukový software pro základní školy (SILCOM Multimedia) [online]. Dostupné z: http://www.silcom-multimedia.cz/tituly/
| Jméno a příjmení: | Karolína Šmatlová |
|-------------------|--------------------------------|
| Katedra: | Technické a informační výchovy |
| Vedoucí práce: | PhDr. Jan Lavrinčík, DiS. |
| Rok obhajoby: | 2017 |

Název práce:	Tvorba multimediálního výukového softwaru pro výuku
	českého jazyka a literární výchovy
Název v angličtině:	Creation of multimedia educational software for teaching
	Czech langure and literky education
Anotace práce:	Diplomová práce pojednává o možnostech výukového
	softwaru primárně zaměřených na výuku českého jazyka a
	literární výchovy a dále o možnostech vlastní tvorby
	výukového softwaru.
Klíčová slova:	Výukový software, Programovací jazyk, Visual Basic 2010,
	Český jazyk a literární výchova
Anotace v angličtině:	This diploma thesis deals with the possibilities of teaching
	software primarily focused on the teaching of Czech language
	and literary education, as well as about the possibilities of
	own teaching software development.
Klíčová slova v angličtině	Learning Software, Programming Language, Visual Basic
	2010, Czech Language and Literary Education
Rozsah práce:	73
Příloha:	CD-ROM
Jazyk práce:	Čeština