

POSSIBILITIES OF COMPUTATIONAL THINKING DEVELOPMENT AT EIGHT- OR SIX- YEAR GRAMMAR SCHOOLS THROUGH ONLINE MEANS

MOŽNOSTI ROZVOJE INFORMATICKÉHO MYŠLENÍ ON- LINE FORMOU V PODMÍNKÁCH VÍCELETÝCH GYMNÁZIÍ

Milan Klement

Abstract

Teaching programming at non-technical schools represents a new and relatively unexplored area. It is therefore necessary to ask the questions related to the possible necessity of such teaching and about its potential influence and/or impact on students` s further development. Within the framework we were looking for answers to the increasingly pressing issues which will need to be addressed in connection with the planned implementation of the Digital Education Strategy by 2020. One of the main pillars of the upcoming change in the content of IT subjects introduced by the latter is the inclusion of a thematic unit focused on algorithmization and programming.

Key words: *Computational thinking, algorithmization and programming, eight-year and six-year grammar schools, Visual Studio.NET*

Abstrakt

Výuka programování je v podmínkách netechnicky orientovaných škol novou, relativně neprobádanou oblastí. Je tedy nutné zabývat se otázkami, zda je tato výuka na těchto školách potřebná a jaký může mít dopad na další rozvoj studentů. V rámci předložené práce jsme tedy hledali odpovědi na stále palčivější otázky, které bude potřebné řešit v souvislosti s plánovou implementací Strategie digitálního vzdělávání do roku 2020, kdy jedním z hlavních pilířů chystané změny obsahu kurikula informatických předmětů je zařazení tematického celku zaměřeného na algoritmicizaci a programování.

Klíčová slova: *Informatické myšlení, algoritmicizace a programování, víceletá gymnázia, Visual Studio.NET*

ÚVOD

Změna RVP pro vzdělávací oblast Informační a komunikační technologie, která je plánována v souvislosti s Implementací strategie digitálního vzdělávání do roku 2020, implikuje některá velká očekávání, neboť se změnou kurikula se očekává, že přinese určité zlepšení, a tudíž, že napomůže zvýšit kvalitu vzdělávání. Jednou ze zásadních změn je tedy zavedení povinného tematického celku zaměřeného na problematiku algoritmicizace a programování, jehož cílem je rozvoj informatického myšlení (computational thinking) u žáků a studentů. Tato problematika byla doposud v kurikulu informatických předmětů na základních školách opomíjena a bylo pouze na uvážení učitelů, zda ji budou do výuky zařazovat či nikoliv.

Pokud je tedy již výuka problematiky algoritmicizace a programování ze strany učitelů realizována, tak se názor na ni často pohybuje mezi dvěma póly. Prvním pólem je využití výukových programovacích jazyků a vývojových prostředí jako jsou například Python či Kodu Game Lab. Druhý pól potom tvoří „reálné“ programovací či skriptovací

jazyky jako Visual Basic, PHP a Java, které umožňují tvorbu produkčních aplikací. Odborná veřejnost vede již řadu let diskusi, kdy zastánci jednoho pólu argumentují nutností didaktického přístupu k výuce programování formou hry, a naproti tomu zastánci druhého pólu potom argumentují nutností výuky „reálných“ programovacích jazyků, jejichž základy potom žáci využijí v praktickém životě či dalším vzdělávání (např.: Pitner, 2000; Klement, Kubrický, 2009; Klement, Bártek, 2019 apod.). Nicméně je neoddiskutovatelným faktem, že znalost některého z programovacích jazyků se dnes i pro běžného uživatele stává stále citelnější potřebou. Ať již jde o programování a tvorbu WWW stránek nebo o vytváření maker dokumentů, až po vytváření vlastních aplikací pro běžnou denní potřebu (Klement, Lavrinčík, 2011). Hlavním cílem takto zaměřené výuky na základních školách a odpovídajících ročnících víceletých gymnázií, by ale neměla být výchova produkčních programátorů, kteří zvládnou i náročné algoritmizační úlohy, ale především všeobecná příprava žáků a studentů na to, aby byli schopni pomocí metod algoritmizace řešit i běžné úlohy v praktickém životě. Toto je také podstatou chystané změny kurikula informatických předmětů, kde má být již problematika algoritmizace a programování nedílnou součástí výuky.

Implementace výuky algoritmizace a programování v této fázi stále naráží na neujasněnou koncepci a volbu konkrétních nástrojů ze strany tvůrců nového kurikula informaticky zaměřených předmětů na základních školách a odpovídajících ročnících víceletých gymnázií. Není také prozatím jasné, zda stávající dotace informatických předmětů bude, s ohledem na zvyšující se rozsah učiva, rozšiřována či bude ponechána ve stávajícím rozsahu s tím, že některá ze stávajících témat budou redukována. Je nutné také podotknout, že s ohledem na rozvoj některých konceptů zaměřených na podporu technického vzdělávání, které sledujeme naprosto oprávněnými, může docházet ke zvyšování výukové zátěže i v jiných předmětech. Může nastat stav, kdy budou do stávajícího kurikula zařazovány stále nové tematické celky (informatické, technické, environmentální, zaměřené na etickou výchovu, zaměřené na finanční gramotnost), které jsou pro žáky a studenty bezesporu také velice přínosné, ale celkový objem výuky je nemožné rozšiřovat nad zákonný rámec. Bude tedy potřeba najít spoustu kompromisů, jak na jedné straně kurikulum aktualizovat, tak aby vyhovovalo potřebám žáků a studentů, ale na druhou stranu neohrožoval všeobecný vzdělanostní základ.

1 JAKÉ NÁSTROJE TEDY VYUŽÍT PRO VÝUKU ZÁKLADŮ ALGORITMIZACE A PROGRAMOVÁNÍ NA VÍCELETÝCH GYMNÁZIÍCH?

Na základě výše uvedených důvodů se domníváme, že bude potřebné hledat kompromisy i v situaci, kdy bude upravováno kurikulum informatických předmětů směrem k rozšíření výuky algoritmizace a programování, a proto se kapitola zabývá možnými přístupy k výuce základů algoritmizace a programování jako součástí vzdělávání studentů víceletých gymnázií. Uvádí, mimo jiné, i možné metody výuky, včetně některých aspektů, které se vztahují k základům programování v objektově orientovaných jazycích pro tuto skupinu studentů. Pro úplnost je nutné dodat, že dále prezentovaná koncepce se jeví jakožto nevhodná pro nižší stupně vzdělávání (primární a sekundární), neboť v tomto věku ještě nejsou žáci schopni vysoké míry abstrakce a je tedy vhodné využívat jiné formy výuky této problematiky (Klement, Bártek, 2019).

S ohledem na dřívější zkušenosti s výukou základů programování (např. projekt PROŠ – viz www.pros.upol.cz) doporučujeme se zaměřit, vzhledem k omezené časové

dotaci, na některý z produkčních objektově zaměřených jazyků, který i při nižší znalosti problematiky umožní vytvořit efektivní aplikaci. V reálné aplikaci a činnostním pojetí výuky algoritmizace a programování totiž vidíme značnou přidanou hodnotu, která umožní studentům nejen pochopit základní principy, ale také jim umožní praktické využití při dalších činnostech. Chápeme, že existuje i druhá strana pohledu, která tuto koncepci odmítá a zaměřuje se na využití některých z výukových programovacích jazyků a vývojových prostředí jako jsou například Python, Scratch či Kodu Game Lab. S příznivci této strategie výuky problematiky algoritmizace a programování proto nepolemizujeme a pouze předkládáme alternativní pohled na tuto problematiku, která vychází z našich reálných zkušeností s výukou na víceletých gymnáziích.

Jako vhodný objektově orientovaný programovací jazyk se nám potom jeví Visual Basic.NET (viz <http://www.pros.upol.cz/files/others/ucebnice-online/index.htm>), který jako jeden z nejrozšířenějších objektových jazyků umožňuje vytváření aplikací běžících pod operačními systémy Microsoft Windows (Halvorson, 2015) s přesahem do tvorby webových aplikací či maker (Visual Basic for Application). Je potřebné podotknout, že podobná koncepce výuky algoritmizace a programování, založená na využití produkčního jazyka Visual Basic.NET, je využívána i na některých školách v zahraničí (Strnad, 2015) a proto se nejedná o pouhou anomálii.

Volba tohoto jazyka má ještě několik výhod, které spatřujeme především v těchto bodech:

- student je schopen i bez vysoké míry abstrakce a znalosti programovacího kódu, vytvářet jednoduché, ale fungující aplikace,
- pro studenta je programování ve Visual Basic.NET kombinací vizuální úpravy komponent nebo ovládacích prvků ve formuláři (RAD - Rapid Application Development) a specifikování vlastností vybraných komponent pomocí grafického rozhraní,
- student vytváří uživatelské prostředí pomocí panelů s ovládacími prvky (toolbox), které se přenášejí technikou drag-and-drop přímo na formulář vytvářené aplikace,
- řazení programového kódu je lineární a student se v něm tedy může snadno orientovat bez nutnosti použití návěstidel či skoků,
- pro studenta je lehké v něm navrhovat různé jednodušší GUI aplikace, ale také skládat celkem složitější komplexní kódy pomocí již existujících API, což mu umožňuje takřka neomezený rozvoj vyvíjené aplikace,
- student, který si osvojí základy práce s Visual Basic.NET, může po krátkém zaškolení přejít na jiný objektově orientovaný programovací jazyk,
- vzhledem k existenci programovacího rozhraní Visual Basic for Applications může student vytvářet makra a jiné doplňky v aplikacích produkovaných firmou Microsoft (Klement, Klement & Lavrinčík, 2012),
- při základních znalostech z oblasti programování ve Visual Basic.NET může student využít Visual Basic Script, jako doplněk při vytváření WWW stránek v HTML (Vrbík, 2012).

I v případě využití produkčního jazyka pro výuku algoritmizace a programování je ale nutné, s ohledem na přiměřenost věkové skupině studentů, provést didaktickou transformaci (rekonstrukci) obsahu, a vybrat pouze ta nejzákladnější a nejzásadnější témata, která jsou pro dosažení cíle nejdůležitější. Cílem výuky v této oblasti není příprava programátorů, ale poučených uživatelů, kteří mohou reálnou zkušenost s algoritmizací a programování dále rozvíjet. Jako optimální se také jeví rozložit problematiku tak, aby na sebe logicky navazovala a zaměřit se především jen na jisté

okruhy znalostí z oblasti algoritmizace a využívání programovacího jazyka (Klement, 2005). Některé specializované oblasti je třeba vynechat s tím, že student po získání základních znalostí a dovedností další znalosti získá samostatným studiem této problematiky, pokud se v ní bude chtít dále rozvíjet.

2 JAK ZAHÁJIT VÝUKU ZÁKLADŮ ALGORITMIZACE VE VISUAL BASIC.NET?

Pro pochopení principů funkce a vytváření aplikací pomocí programovacího jazyka je nutné nejprve studenty seznámit se základními pojmy z oblasti počítačového zpracování dat. V této fázi výuky je vhodné zkombinovat teoretický výklad s praktickými ukázkami jednotlivých pojmů přímo v prostředí uživatelského jazyka a umožnit i studentům manipulaci s připravenými ukázkovými příklady. Je tedy vhodné i v této fázi využít činnostní charakter výuky a uvedenou problematiku nejen frontálně demonstrovat, ale v rámci individualizované či skupinové výuky nechat studenty pracovat s předem připravenými ukázkami.

V prvotní fázi seznámit studenty s pojmy jako (Klement, Lavrinčík, 2012):

- algoritmus a jeho význam,
- program a jeho části,
- definiční a deklarační části programu,
- příkazová část programu,
- pojem funkce a procedury,
- popis procesu vykonávání programu.

Uvedené seznámení vychází z potřeb programování v objektově zaměřeném jazyce, a to především v oblasti vysvětlení příkazů, funkcí, procedur a syntaxe jejich zápisu (Klement, Lavrinčík 2011).

V další fázi je třeba rozvíjet některé konkrétní oblasti z problematiky algoritmizace, ale opět s ohledem na výše definované podmínky, kdy by měli mít žáci opět k dispozici ukázková řešení se zakomponovanými komentáři, se kterými mohou samostatně pracovat.

V této oblasti by se měla pozornost věnovat především těmto pojmům (Klement, Lavrinčík, 2012):

- jednotlivé datové typy,
- proměnné a hodnoty a jejich deklarace,
- základní operace s proměnnými,
- matematické operace s deklarovanými i vygenerovanými proměnnými,
- zápis a použití jednoduchých a strukturovaných příkazů,

Obsah a rozsah uvedených témat by měl opět vycházet především z potřeb programování v objektově zaměřeném jazyce Visual Basic. Některé výše uvedené pojmy potom nejsou závislé na programovací platformě a jejich znalost pomůže studentům k vytvoření základního přehledu o vytváření a funkci programů. Tyto znalosti umožní také lepší orientaci ve vnitřních procesech počítačového zpracování dat, které studenti mohou s úspěchem využít při dalším vzdělávání.

Po zvládnutí této části výuky mohou studenti přikročit k vlastní práci s programovacím jazykem a znalosti, které získali při teoretickém studiu, mohou v plné míře uplatnit při vytváření vlastních, zatím jednoduchých programů (Kocich, Spilka, 2010).

3 JAK KONCIPOVAT VÝUKU ZÁKLADŮ PROGRAMOVÁNÍ VE VISUAL BASIC.NET?

Po absolvování základů algoritmizace mohou studenti přikročit k vlastnímu programování v objektovém jazyce Visual Basic.NET. Znalosti, které získali v první a druhé fázi přípravy, nyní mohou prakticky aplikovat při tvorbě jednoduchých programů. Zde se jako přínosné jeví využití připravených ukázkových příkladů, které mohou studenti doplňovat, či upravovat dle vlastních návrhů. Vhodným přístupem je také využití aplikací, které obsahují chyby a studenti je mohou pomocí debuggeru odhalovat a opravovat, neboť hledání a opravování chyb je jednou s velmi názorných metod výuky programování. Učitel tak může poskytovat studentům sady ukázkových příkladů ve stupni: ukázka s chybami – najdi chybu; ukázka s kódem pro doplnění – oprav chybu; ukázka s možností rozšíření – uprav a rozšiř aplikaci. Vzhledem k omezenému rozsahu výuky se tak jako optimální jeví praktická aplikace poznatků na konkrétní příklady, včetně vytváření programů, které studentům umožní lépe pochopit probíranou látku (Klement, 2005).

V třetí fázi výuky tohoto tématu je vhodné se zaměřit především na tyto činnosti a pojmy (Klement, Lavrinčík, 2012):

- práce s aplikačním prostředím programovacího jazyka Visual Basic,
- možnosti vytváření formulářů,
- používání objektů a nástrojů obsažených v nabídce programu,
- zápis zdrojového kódu objektů a anatomie handlerů událostí,
- deklarace proměnných, procedur a funkcí,
- možnosti ladění pomocí Debug Window.

Tato fáze je zaměřena především na základní přehled a orientaci v prostředí jazyka Visual Basic.NET. Rozšiřuje znalosti studentů tím, že pomocí praktické aplikace znalostí získaných v teoretické části výuky upevňuje tyto znalosti a ukazuje jim praktické užití příkazů a deklarácí.

Čtvrtá fáze je potom zaměřena na vlastní tvorbu aplikací ve Visual Basic.NET. V této části je vhodné se zaměřit zejména na tyto oblasti (Klement, Lavrinčík, 2012):

- zásady tvorby aplikací a návrh uživatelské rozhraní,
- návržení a možnosti formuláře a vhodný výběr události,
- zápis obslužného kódu, jeho modifikace a možnosti rozšíření,
- vytváření nabídek (Menu Bar) navrhovaného programu,
- příkazy pro práci se soubory a databázemi,
- tvorba vlastních jednoduchých programů.

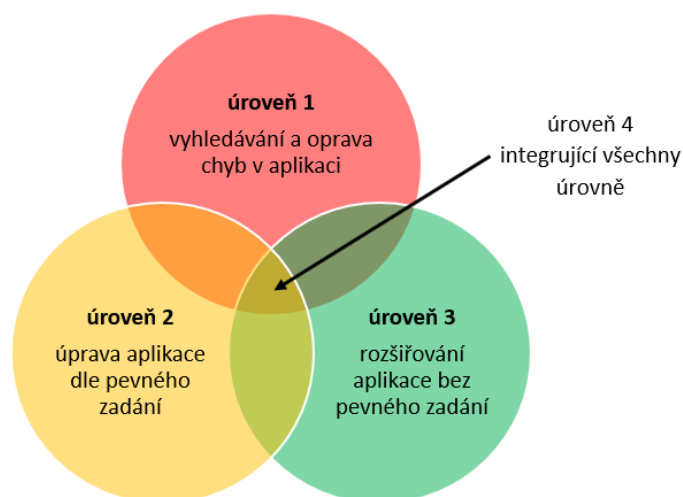
Pátá a závěrečná fáze výuky základů programování ve Visual Basic.NET má za cíl spojit získané dílčí vědomosti s používáním programovacího jazyka. Především se jedná o problematiku vytváření vhodných výukových či produkčních aplikací (Halvorson, 2015), se zřetelem na dodržení didaktických zásad prezentace informací ve výuce. Složení obsahu tohoto oddílu výuky je vhodné volit takto (Klement, Lavrinčík, 2012):

- typy výukových či produkčních aplikací,
- ukázka základních přístupů a postupů při programování těchto aplikací,
- využití multimédií (zakomponování zvuku, videa a grafiky) jako vhodného doplňku výukového programu,

- přizpůsobení programů různým typům zobrazovacích soustav PC a mobilních zařízení,
- možnosti vytváření instalačních celků pro potřeby šíření programu (vytváření EXE instalačních souborů),
- archivace vzniklých instalací na různá média.

4 JAK VYMEZIT STRUKTURU A OBSAH VÝUKY ZÁKLADŮ PROGRAMOVÁNÍ VE VISUAL BASIC.NET?

Jak již bylo několikrát zdůrazněno, vzhledem k činnostnímu pojetí výuky se jako optimální jeví praktická aplikace poznatků na konkrétní příklady, včetně vytváření programů, které studentům umožní lépe pochopit probíranou látku (Klement, 2005). Jelikož ve skupinách studentů existují difference, dá se očekávat, že ne všichni studenti budou schopni ve stejném čase a ve stejné úrovni postupovat učivem. Zde se jako optimální jeví přizpůsobení obsahu výuky dle níže uvedeného modelu, kdy studentům s méně kvalitním výkonem můžeme zadávat příklady ve kterých pouze vyhledávají a opravují chyby pomocí debuggeru, a studentům s vyšším výkonem potom zadávat příklady ve kterých se očekává vyšší podíl samostatnosti a vzhledu do problematiky, ve kterých mohou rozvinout svoji tvořivost a ukázkové aplikace dále rozvíjet. Model takto koncipované výuky je tedy možné načrtnout v těchto čtyřech úrovních (Klement, Bártek, 2019).



Obrázek 1 Struktura úrovněvého modelu výuky algoritmicke a programování

Cílem takto koncipované výuky je dosažení 4 – integrující úrovně, což se ale nutně nemusí podařit u všech studentů, neboť míra jejich abstraktního myšlení toto neumožní. Jako jedna z vhodných cest, jak se s tímto vyrovnat je využití prvků projektového vyučování či skupinové práce.

Pro ilustraci uvádíme jeden z příkladů konkrétního obsahu výuky, který odpovídá výše uvedenému modelu:

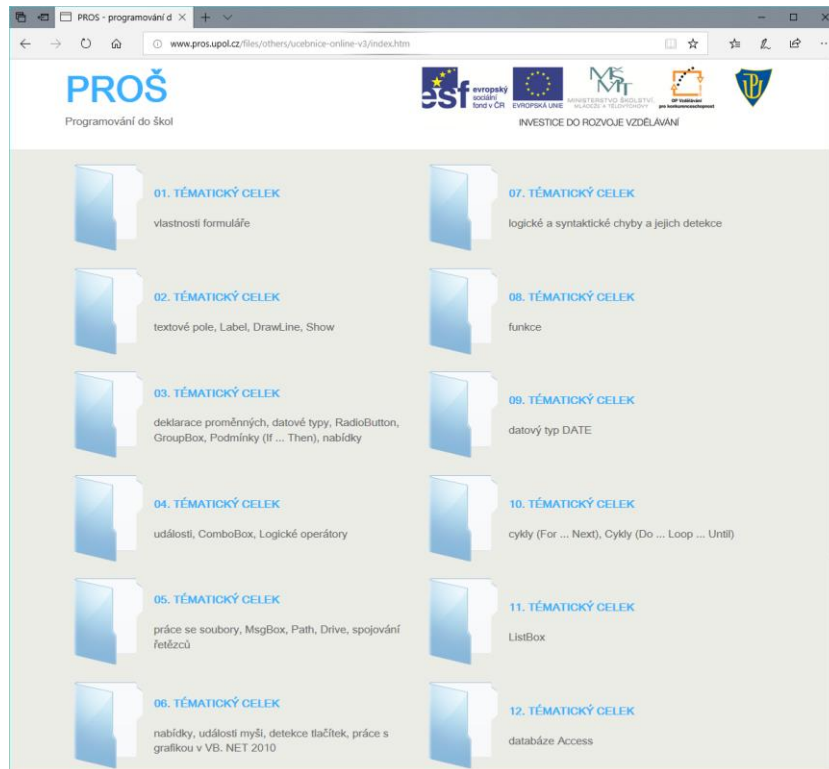
- Úvod do algoritmicke v jazyce Microsoft® Visual Basic.NET (dále jen VB). Pojem funkce a procedury, syntaxe jazyka. probírané pojmy, příkazy, funkce a procedury Sub, Function, Dim, End, Private.
- Popis prostředí jazyka VB - práce s prostředím programu, nástrojové lišty a práce s programem.

- Seznámení s vývojovým prostředím VB – probírané pojmy, příkazy, funkce a procedury: Form, Command, End, Beep.
- Vývojové prostředí VB – probírané pojmy, příkazy, funkce a procedury: Line, Text, Form, Command, Unload, Show, Label.
- Vývojové prostředí VB – probírané pojmy, příkazy, funkce a procedury: OptionButton, If...Then, Booleovské operátory.
- Práce s ovládacími prvky VB - probírané pojmy, příkazy, funkce a procedury: Combo, AddItem, Frame, Label, Caption, EXE.
- Práce se soubory ve VB – probírané pojmy, příkazy, funkce a procedury: Drive, Dir, File, MsgBox, LoadPicture, &, \, „ .
- Práce s proměnnými ve VB – probírané pojmy, příkazy, funkce a procedury: MouseMove, MouseDown, Click, Nabídka, Dim, Public.
- Práce s multimédií ve VB – probírané pojmy, příkazy, funkce a procedury: Dir, Drive, File, AddItem, MediaPlayer.
- Práce s řetězci ve VB – probírané pojmy, příkazy, funkce a procedury: String, Label, Replace, Split, Len, Frame.
- Práce s databázemi ve VB – probírané pojmy, příkazy, funkce a procedury: Data, FlexGrid, RecordSource, Database, Table.
- Práce s uživatelsky definovanými datovými typy – probírané pojmy, příkazy, funkce a procedury RichTextBox, Ole, Datový typ.

Jak je z uvedeného přehledu patrné, neobsahuje vymezený ukázkový obsah všechny atributy, funkce, příkazy, objekty a syntaxe jazyka Visual Basic.NET, neboť je přizpůsoben možnostem studentům víceletých gymnázií. Záleží tak především na úvaze učitele, zda tento obsah bude rozšiřovat o další celky, či jej naopak bude redukovat dle aktuální věkové či výkonnostní charakteristiky svých studentů. Cílem takto koncipované výuky totiž není výchova produkčních programátorů, ale především poučených uživatelů, kteří takto mohou pochopit zákonitosti fungování počítačového software a v případě zájmu jej potom dále rozvinou pomocí dalších informačních zdrojů, které jsou k dispozici na Internetu. Učitel by tedy měl být připraven poskytnout studentům potřebnou podporu i v této fázi samostudia a proto by měl být s vyučovaným programovacím jazykem patřičně obeznámen.

5 POPIS ON-LINE NÁSTROJE PRO REALIZACI VÝUKY ZÁKLADŮ ALGORITMIZACE A PROGRAMOVÁNÍ VE VISUAL BASIC.NET

Pro potřeby podpory výuky tematického celku zaměřeného na základy algoritmizace a programování ve Visual Basic.NET byl vytvořen specializovaný výukový nástroj v podobě interaktivní multimediální výukové aplikace, postavené na platformě jazyka HTML (Vrbík, 2012) a PHP (Thompson & Nowicky, 2010). Toto pojetí umožňuje použití aplikace jak v on-line (dostupné na adrese: <http://www.pros.upol.cz/files/others/ucebnice-online-v3/index.htm>), tak i off-line verzi (dostupné na adrese: http://www.pros.upol.cz/files/others/PROS_el_Ucebnice_programovani_v3.iso) a také jeho rychlou úpravu, dle aktuálních potřeb. Jak ukazuje níže uvedený obrázek číslo 2, aplikace respektuje výše uvedenou koncepci výuky základů algoritmizace a programování ve Visual Basic.NET a naplňuje ji konkrétním obsahem.



Obrázek 2 Interaktivní učebnice (Klement & Lavrinčík, 2011b)

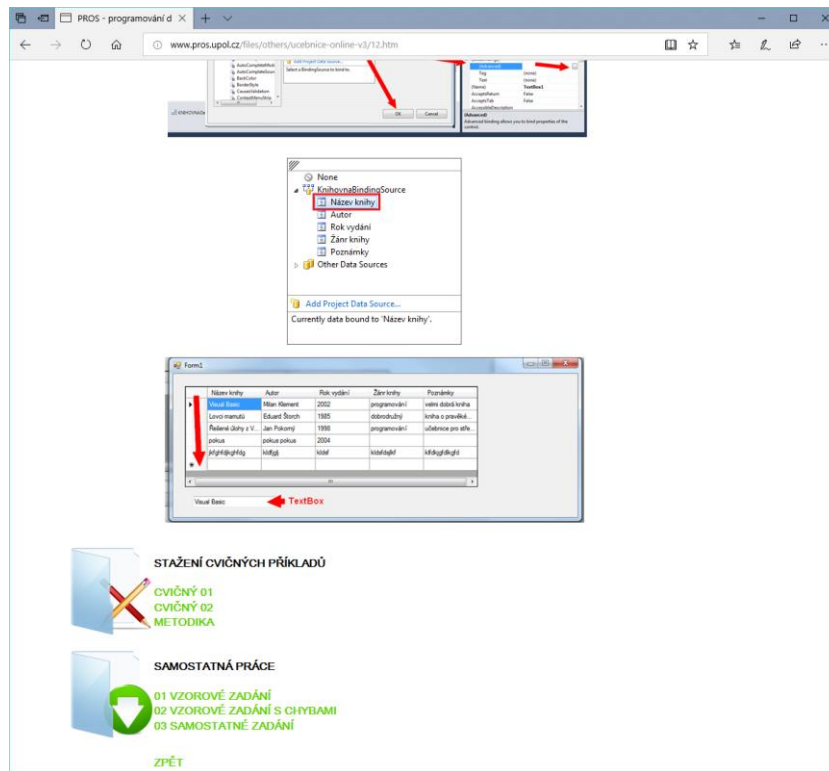
Aplikace je tedy tvořena 12-ti samostatnými výukovými moduly, které pokrývají problematiku znalostí a dovedností, které jsou bezprostředně nutné pro základní orientaci v oblasti základů algoritmizace a programování ve Visual Basic.NET.



Obrázek 3 Konkrétní ukázka výukového modulu

Výuková aplikace neobsahuje pouze statický text, ale je doplněna celou řadou interaktivních objektů ve formě multimediálních animací jednotlivých postupů řešení a také ukázek zdrojových kódů, se kterými mohou studenti samostatně manipulovat.

Důležitou součástí výukové aplikace tvoří cvičné příklady a zadání pro samostatnou práci studentů. Zatímco ukázkové příklady jsou určeny pro frontální část výuky, kde učitel pomocí nich vysvětluje danou problematiku a studenti mohou vše zkusit či upravovat na svých počítačích, tak zadání pro samostatnou práci jsou určeny pro individualizovanou či skupinovou část výuky, při které studenti pracují samostatně na daném úkolu. Tuto situaci ukazuje níže uvedený obrázek číslo 4.



Obrázek 4 Cvičné příklady a vzorová a samostatná zadání

Pro úplné využití všech možností výukové aplikace je potřebné mít na učitelských i studentských počítačích nainstalován produkt Visual Studio minimálně verze 2010. Aktuální verze Visual Studia 2019 je bezplatně ke stažení na stránkách firmy Microsoft na adrese <https://visualstudio.microsoft.com/cs/> a studenti i učitelé jej tedy mohou používat nejen na školních počítačích, ale mohou si jej bezplatně stáhnout a používat i na svých osobních počítačích.

ZÁVĚR

Programování se nedá osvojit pouhým studiem teorie. Je třeba získávat a třídit teoretické informace, pochopit algoritmy programovacího jazyka, řešit problémy, studiem a praktickým cvičením upevňovat postupy, řešit vzorové příklady, dokončovat rozpracované varianty, a konečně zkusit psát vlastní programy. Z toho důvodu je potřebné, s ohledem na didaktickou stránku, rozčleněn vyučovanou problematiku základů algoritmizace a programování do takových celků tak, aby studenti postupně a co nejtěsněji s realitou každodenní praxe získávali a postupně si osvojovali jednotlivé skupiny informací, které následně doplní praktickou činností. Proniknout samostatně do tajů programování, cílevědomě rozvíjet svou kreativitu, zároveň se naučit

nenásilnou formou pracovat s výpočetní technikou, a navíc chápat svůj úkol jako práci na projektu je pro řadu z nás úkol složitý, pro mnohé z nás pak individuálně a bez dopomoci neřešitelný.

Představený on-line nástroj tedy nabízí studentům víceletých gymnázií možnost seznámit se a zdokonalit se v programování s využitím objektového programovacího jazyka Microsoft Visual Studio, a rozvíjet jednotlivé dovednosti nově, poutavě a samostatně řešit úkoly v návaznosti na individuální znalosti práce s ICT, efektivně a tvořivě využívat prostředky komunikace, a přijmout odpovědnost za vlastní práci. Všechny uvedené kompetence společně mohou přispět k širší adaptabilitě studentů při dalším studiu na vysokých školách.

References

1. Halvorson, M. (2015). *Microsoft Visual Basic: krok za krokem*. Brno: Computer Press. 647 s. ISBN 978-80-251-4412-1.
2. Vrbík, V. (2012). *Webové technologie*. Hradec Králové: Gaudeamus. 145 s. ISBN 978-80-7435-193-8.
3. Klement, M. & Lavrinčík, J. (2012). *Metody realizace a hodnocení výuky základů programování*. Olomouc: Jiří Dostál. 96 s. 978-80-87658-01-7.
4. Klement, M. & Lavrinčík, J. (2011). *Úvod do MS Visual Basic 2010*. Olomouc: Velfel. 49 s. ISBN 978-80-87557-07-5.
5. Klement, M. (2005). *Výuka algoritmizace a programování v jazyce Visual Basic*. In: Sborník přednášek z mezinárodní vědecko-odborné konference: *Trendy technického vzdělávání. Pedagogická fakulta UP Olomouc*. 211-214 s. ISBN 80-244-0107-X.
6. Kocich, P. & Spilka, O. (2010). *1001 tipů a triků pro Microsoft Visual Basic*. Brno: Computer Press. 520 s. ISBN 978-80-251-2118-4.
7. Klement, M. & Lavrinčík, J. *Elektronická učebnice programování on-line*. Olomouc: Jiří Dostál. 189 s. Dostupné na: <http://www.pros.upol.cz/files/others/ucebnice-online-v3/index.htm>
8. Thompson, E. & Nowicki, S. (2010). *PHP 6 – programujeme profesionálně*. Brno: Computer Press. 718 s. ISBN 978-80-251-3127-5.
9. Klement, M. & Bártek, K. (2019). *Od digitální gramotnosti k informatickému myšlení – koncepce, obsah a realizace výuky informatiky z pohledu jejich aktérů*. Olomouc: Vydavatelství UP, 2019, 244 s. ISBN 978-80-244-5549-5.

Contacts

doc. PhDr. Milan Klement, Ph.D.
Univerzita Palackého v Olomouci, Pedagogická fakulta
Žižkovo náměstí č. 5, 771 40 19 Olomouc
Tel: +420 585 635 811
E-mail: milan.klement@upol.cz