

OBJECT ORIENTED PROGRAMMING IN TEACHING

Jan KUBRICKÝ - Milan KLEMENT

Abstract: *The paper deals with Object Oriented Programming. It reviews the opinions of eminent educators - programmers and focuses on object-oriented possibilities in the design and direction of web applications.*

Key words: *object oriented programming, teaching of programming, web application, programming language.*

OBJEKTOVĚ ORIENTOVANÉ PROGRAMOVÁNÍ VE VÝUCE

Resumé: *Príspevok je venovaný problematice Objektově Orientovaného Programování. Shrnuje názory předních odborníků pedagogů-programátorů a zabývá se objektově orientovanými možnostmi při tvorbě a správě webových aplikací.*

Klíčová slova: *objektově orientované programování, výuka programování, webové aplikace, programovací jazyk.*

1 Úvod

Objektově orientované programování (dále jen OOP) se začalo prosazovat jako klíčová programátorská technologie již v 90. letech minulého století. V zápětí mnozí z uznávaných programátorů-pedagogů na tuto skutečnost začali reagovat ve výuce příslušných kurzů a také ve výuce zájmových kroužků na základních a středních školách. Za hlavní poznávací znak OOP je totiž považována jeho relativní podoba s charakteristikou reálného světa, což umožní i mladším žákům lepší představu základní myšlenky objektově orientovaného přístupu. OOP je totiž založeno na síti vzájemně komunikujících objektů, tak, jak je známe v určité podobě v běžném životě. Jak uvádí R. Pecinovský [1, s. 2], „svět objektů, které si navzájem posílají zprávy jsou pro žáky daleko pochopitelnější, než svět posloupnosti příkazů“, které jsou charakteristické pro programování strukturované.

Objektově orientovaný přístup našel své místo také v programování webových aplikací. Dobře známe např. programovací jazyk JAVA nebo novou technologii od společnosti Microsoft ASP .NET a programovací jazyky C# nebo Visual Basic .NET. Pravdou ale zůstává, že nejoblíbenějším a nejrozšířenějším programovacím jazykem při vytváření webových aplikací je volně distribuovaný jazyk PHP. Ve výuce na školách tomu není naopak a s příchodem jeho verze s označením PHP 5 je možné využívat jeho objektově orientovaných možností [4] [5]. V dalším textu se nejprve

zaměříme na výchozí charakteristiku objektového přístupu a budeme hledat vzájemné vazby z podstaty objektů reálného světa.

2 Reálný svět OOP

Jak již bylo řečeno, základním principem OOP je volná síť vzájemně komunikujících objektů. Pokud na tuto formulaci narazíte poprvé možná ještě ve svých prvotních závěrech přesně nevystihnete základní myšlenku objektově orientovaného přístupu. Jakmile ale začnete uvažovat více s ohledem na obraz reálného světa dospějete k překvapivému zjištění. O objektech reálného světa můžeme říci, že každý z nich má řadu nejrozličnějších atributů [3], stejně jako objekty počítačové programu. Řekneme například, že každý člověk má jinou barvu vlasů, barvu očí atp. Tyto atributy jsou pro různé objekty specifické a mohou se v mnohém lišit. Můžeme tvrdit, že automobil nebo motocykl mají nějaký počet kol, ale hovořit o tomto počtu v souvislosti s obrazem na stěně asi nemá smysl. Objekty lze proto celkem přirozeně dělit do skupin téhož druhu. V každé skupině jsou objekty s týmiž (nebo velmi podobnými atributy), jenž se liší pouze hodnotou. Dále platí, že objekty nejsou pouze pasivními nositeli atributů, ale dokáží i aktivně vytvářet odezvu na akce okolního světa, jenž je reprezentován ostatními objekty.

Pokud si dokážeme na základě výše uvedeného asociovat jednotlivé pojmy do programových konstruktů, můžeme „objekt“ realizovat také ve vývojovém prostředí

výpočetního programu [3]. Dále je před námi tedy otázka volby vhodného programovacího jazyka do výuky.

3 Programovací jazyk pro výuku OOP

V plánování výuky si pedagogové-programátoři kladli základní otázku, „jaký zvolit vhodný programovací jazyk“. Sama skutečnost, že existuje více programovacích jazyků, které podporují OOP nebo jsou přímo objektově orientované jest důkazem toho, jaký význam OOP ve výuce má. Naučíme-li žáky programovat a myslet objektově v jednom z programovacích jazyků, naučíme je nejenom techniky a koncepty OOP, ale připravíme jim také pevné základy naučit se další z programovacích jazyků a využívat přitom dříve nabytých znalostí a dovedností. V podobné souvislosti uvádí R. Pecinovský [1, s. 1], že „v současné době před námi nestojí otázka zda učit objektově orientované programování, ale jak je učit“. Autor ve své práci dále uvádí, jakými vlastnosti by zvolený programovací jazyk měl disponovat. Uvedme ty nejpodstatnější [1]:

- Měl by být především doopravdy objektově orientovaný. Tím automaticky vypadává starý Visual Basic (Visual Basic .NET již objektově orientovaný je);
- Měl by převzít na svá bedra co největší množství úkolů, které by programátor nemusel bezpodmínečně řešit – např. správu paměti. Tato podmínka vyřazuje Delphi až do verze 7 včetně a C++ (C++ pro .NET již správu paměti zprostředkovává);
- Měl by být maximálně jednoduchý, abychom se při výuce mohli soustředit na programování a nemuseli trávit dlouhé chvíle vysvětlováním vlastností a hlavně záladností jazyka (tady definitivně vypadává C++);
- Měl by být maximálně bezpečný a odhalit co nejvíce chyb již ve fázi překladu;
- Měl by mít minimální nároky na hardware počítače (hardwarové nároky jsou ale silně ovlivněny použitým vývojovým prostředím);
- Překladače, vývojová prostředí a nástroje by měly být dostupné za minimální cenu, nejlépe zdarma;
- Měl by být dostatečně rozšířený, aby bylo možno vyměňovat zkušenosti v rámci co nejširší komunity;

Vzhledem k výše uvedeným vlastnostem preferují pedagogové-programátoři i jiní odborníci zejména jazyk C#, a dále pak Delphi

pro .NET, Java nebo velmi oblíbený Visual Basic .NET.

4 OOP ve výuce

Programování je ve výuce považováno za překonávání počítačové gramotnosti, a zkoumání a vytváření nitra procesů, které většině populace zůstanou skryty za prostředím, jenž je jim důvěrné známé a prostupuje dnes již téměř každou sférou lidského života. Nicméně i programování je běžnému životu blíže, než by se na první pohled mohlo zdát. V druhé části našeho příspěvku jsme stručně nastínili, jak nahlížet na OOP z pohledu procesů v reálném světě.

Výuka programování by proto měla tyto aspekty u žáků a studentů podněcovat. Čím více studující přijmou skutečnost, která je povede řadou asociací s běžným životem, tím lépe budou schopni se OOP naučit. R. Pecinovský [1] [2] doporučuje výuku započatou oklikou. Nejprve žáky nechat „hrát si“ s hotovým programem a vytvářet jen malé doplňky do nějakého již existujícího světa. Efektivním a přitažlivým způsobem tak nenásilně vstoupí do světa programování. Vlastní metodika OOP pak vychází z následujícího:

- Úvodem představit základní pojmy **třída** a **objekt**, a nahlédnout do struktury předem připraveného programu a vysvětlit vzájemné vztahy;
- V další fázi zapojit vlastní aktivitu spojenou s objevováním vztahu mezi **třídou** a **instancí**. Volání **metod**, analyzování jejich chování a zkoumání hodnot jejich atributů;
- Vlastní programování. Prvotně se naučit upravovat a ladit existující programy a postupně se na nich naučit vše, co je potřeba k vytváření vlastních programů;

Úvodní metodika výuky OOP je nezbytným předpokladem pro další kontinuitu výuky. Na základě získaných znalostí a dovedností si žáci a studenti utváří kompetence, které jsou nezbytné pro další rozšiřování v oblasti dané problematiky. Všechny uvedené kroky mají v hierarchii výuky programování nezastupitelné místo a není možné je vynechat nebo se k nim bez předchozího vysvětlení vracet.

5 OOP ve výuce tvorby webových aplikací

Webové aplikace tvoří nedílnou součást komunikace nejrozličnějších objektů. Komunikují spolu uživatelé na internetových fórech,

komunikují spolu bankovní účty při platebních transakcích nebo spolu komunikuje uživatel aplikace Google earth a družice na oběžné dráze. To vše se děje prostřednictvím sítě Internet a mnoha webových aplikací, které nabízejí stále vyspělejší, dokonalejší a rychlejší způsoby různých forem komunikace. Není proto divu, že objektově orientovaný přístup byl tvůrci webových aplikací vysněným cílem. Umožňuje jim to jazyk Java, prostřednictvím technologie ASP. NET jazyky C# nebo Visual Basic .NET. Ovšem nejoblíbenější PHP v tomto ohledu stále mírně zaostával. To vše se změnilo po roce 2005 s příchodem verze 5 a plné podpory objektově orientovaného přístupu [4]. I přesto ve výuce jazyka PHP převládá procedurální přístup a dle R. Pecinovského [1] [2] se přikláníme k důvodu, že OOP je pro učitele v podstatě náročnější než pro samotné žáky. Jazyk PHP je ale nazýván hybridním, neboť záleží na každém jedinci, jakou cestou se vydá.

A právě tato skutečnost, nám umožňuje využít obou programovacích technik. Základy OOP je ovšem nutné zařadit do výuky již od samého počátku a ne jim věnovat pozornost pouze okrajově. Žáci si pak odnášejí kompetence, které mohou přenášet i do jiných vývojářských prostředí v této oblasti. Nejsilnějším konkurentem PHP, technologie ASP .NET od Microsoft a programování v jazyce C# je plně objektově orientované a ještě do nedávné doby bylo považováno za technologie pro profesionály a velmi pokročilé. Bereme v ohled především jeho licenční politiku a zejména pak objektově orientovaný přístup jazyka C#, který byl pro mnoho programátorů webových aplikací zvyklých na klasické programování v jazyce PHP těžkým oříškem. Ovšem jak razantně se situace změnila když inovujeme výuku OOP v jazyce PHP. Možný přechod na technologii ASP .NET bude o poznání snazší.

6 Závěry a doporučení

OOP má v prostředí jazyka PHP své příznivce i odpůrce. Jeden z nejznámějších pedagogů programátorů R. Pecinovský [1] [2] prosazuje do výuky od samého počátku objektově orientovaný přístup. Dle našich poznatků z výuky programování se s ním ztotožňujeme. Je potřeba zdůraznit, že OOP nenahrazuje klasické strukturované programování, ani jej nevytlačuje na vedlejší kolej. Spíše jej povyšuje do nové progresivní úlohy. Navíc není nutno programovat vždy objektově. Hybridní jazyk PHP nám umožní kdy se nám zachce se vrátit

k čistě strukturovanému programování [5]. Ale výuka by měla být založena a připravena od začátku respektovat objektově orientované možnosti.

6 Literatura

- [1] PECINOVSKÝ, R. *Proč a jak učit OOP žáky základních a středních škol* [online]. Publikováno 10.1.2004.
<URL: http://vyuka.pecinovsky.cz/prispevky/Proc_a_jak_ucit_OOP_na_ZS_a_SS.pdf>.
- [2] PECINOVSKÝ, R. *Výuka objektově orientovaného programování žáků základních a středních škol* [online].
<URL: http://vyuka.pecinovsky.cz/prispevky/Vyuka_OOP_zaku_zakladnich_a_strednich_skol.pdf>.
- [3] ČADA, O. *Objektové programování*. 1. vyd. Praha : Grada, 2009. ISBN 978-80-247-2745-5.
- [4] LAVIN, P. *PHP objektově orientované*. 1. vyd. Praha : Grada, 2009. ISBN 978-80-247-2137-8.
- [5] KUBRICKÝ, J. Jazyk PHP ve vzdělávání učitelů informační výchovy. *Journal of Technology and Information Education*. 2009, Olomouc - EU, Univerzita Palackého, Ročník 1, Číslo 1, s. 136 - 138. ISSN 1803-537X (print). ISSN 1803-6805 (on-line).
- [6] KLEMENT, M. *Základy programování 1 – Úvod do MS Visual Basic 6.0*. (Texty k distančnímu vzdělávání v rámci kombinovaného studia) Olomouc : UP, 2008. 60 s. ISBN 978-80-244-2177-3.
- [7] DOSTÁL, J. *Počítač ve vzdělávání - modul 1*. Olomouc: Votobia, 2007. 125 s. ISBN 978-80-7220-295-2.

Mgr. Jan Kubrický

Katedra technické a informační výchovy

Pedagogická fakulta UP

Žižkovo nám. č. 5

771 40, Olomouc, ČR

Tel: +420 585 635 818

E-mail: jan.kubricky@upol.cz

Www pracoviště: www.kteiv.upol.cz

PhDr. Milan Klement, Ph.D.

Katedra technické a informační výchovy

Pedagogická fakulta UP

Žižkovo nám. č. 5

771 40, Olomouc, ČR

Tel: +420 585 635 811

E-mail: milan.klement@upol.cz