

Automatické hodnocení úkolů v kurzech programování

Automatic assessment
of assignments
in programming courses

Veronika Vašková

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Vložit oficiální zadání diplomové/bakalářské práce

Prohlašuji, že jsem na celé diplomové práci pracovala samostatně a použitou literaturu jsem citovala.

Ve Zlíně, 05. 06. 2005

.....

jméno diplomanta

Abstrakt

Výukový systém Moodle má modulární strukturu. Kvůli požadavkům kurzů programování bylo třeba vytvořit nový modul, který provádí automatické hodnocení odevzdaných úkolů (programů). V této bakalářské práci je stručně vysvětleno využití tohoto modulu a také jsou zde popsány všechny postupy, prostředky a zdroje, které byly použity k jeho vytvoření.

Moodle is a course management system which has a modular structure. Due to programming courses demands it was required to produce a new module, which executes automatic assessment of submitted assignments (programs). In this bachelor thesis there is briefly explained the use of this module and there are also described all methods, tools and resources used for its creation.

OBSAH

ÚVOD	6
1 TEORETICKÁ ČÁST	7
1.1 MOODLE	7
1.2 HTML	7
1.3 PHP	7
1.4 WEBOVÝ SERVER APACHE.....	8
1.5 MYSQL.....	8
1.6 INTERPRET	8
2 UŽIVATELSKÁ ČÁST	9
2.1 INSTALACE MODULU	9
2.2 ROZHRANÍ PRO UČITELE.....	9
2.2.1 Přidání nové činnosti Program.....	9
2.2.2 Kontrola odevzdaných prací	11
2.3 ROZHRANÍ PRO STUDENTY	12
2.3.1 Odevzdání programu.....	12
2.3.2 Zobrazení výsledků.....	13
3 PROGRAMÁTORSKÁ ČÁST	15
3.1 VÝBĚR INTERPRETU.....	15
3.2 JEDNOTLIVÉ SOUČÁSTI MODULU	15
3.2.1 Ikona modulu	15
3.2.2 Jazykové soubory.....	16
3.2.3 Stránka config.html.....	16
3.2.4 Stránka mod.html.....	16
3.2.5 Soubor version.php.....	17
3.2.6 Databázové soubory.....	17
3.2.7 Zálohovací soubory.....	18
3.2.8 Skript index.php.....	18
3.2.9 Skript submissions.php.....	18
3.2.10 Skript view.php.....	18
3.2.11 Skript upload.php.....	19
3.2.12 Skript lib.php	19
ZÁVĚR.....	24
SEZNAM POUŽITÝCH ZDROJŮ	25
SEZNAM OBRÁZKŮ.....	26

ÚVOD

V poslední době se ve výuce stále více používají různé elektronické pomůcky jako jsou počítačové programy či internetové stránky. Smyslem jejich využití může být snadnější přístup studentů k novým informacím nebo například rychlejší shromažďování a hodnocení studentských prací vyučujícími. Cílem je pak vždy co nejefektivnější výuka.

Na Fakultě technologické Univerzity Tomáše Bati ve Zlíně (dále jen Fakulta technologická) je za tímto účelem používán výukový systém Moodle v současné době přístupný na internetové adrese <http://www.vyuka.ft.utb.cz>. Jeho prostřednictvím vyučující nejčastěji studentům poskytují studijní materiály, vytvářejí diskusní fóra nebo zadávají úkoly, které studenti opět pomocí tohoto systému odevzdávají. Pro každou z těchto činností je v systému Moodle vytvořen samostatný modul.

Do výuky na této fakultě jsou mimo jiné zařazeny kurzy seznamující studenty s programováním v jazyce C či C++. Modul Úkol je pro potřeby těchto kurzů nedostačující, protože nenabízí nic kromě zadávání a hodnocení prací učitelem a odevzdávání vypracovaných úkolů studenty.

Vzhledem k tomu, že je systém Moodle modulární, je snadné jej rozšířit o libovolný nový modul, který splňuje určitá pravidla (má určitou strukturu a vlastnosti). Proto vznikla myšlenka upravit již existující modul Úkol a vytvořit tak modul nový – Program, který by nabízel pro kurzy programování něco navíc. Každý student bude mít možnost ihned zjistit, zda je jeho vypracovaný program v pořádku či kde má chyby a jaké. Vzhledem k tomu, že kontrola bude okamžitá, student se rychleji dozví o nedostacích svého programu a bude je moci dříve odstranit. Informace o funkčnosti odevzdaných programů zjistí rovněž vyučující a to v seznamu odevzdaných prací. To mu umožní oproti modulu Úkol zjistit nejen kolik studentů již svůj program odevzdalo, ale navíc také které z nich fungují správně. Výuka jako celek pak bude mnohem efektivnější a to jak pro studenta, tak i pro vyučujícího.

1 TEORETICKÁ ČÁST

1.1 Moodle

Moodle [čti můdl] je softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu. Jedná se o neustále vyvíjející se projekt, navržený na základě sociálně konstruktivistického přístupu ke vzdělávání. Moodle je poskytován zdarma jako Open Source software (spadající pod obecnou veřejnou licenci GNU).

Slovo Moodle bylo původně akronymem pro Modular Object-Oriented Dynamic Learning Environment (Modulární objektově orientované dynamické prostředí pro výuku); tato informace může být zajímavá především pro programátory a teoretické pedagogy. Lze ho také považovat za sloveso, které popisuje proces líného bloumání od jednoho k druhému, dělání věcí podle svého, hravost, která často vede k pochopení problému a podporuje tvořivost. V tomto smyslu se vztahuje jak k samotnému zrodu Moodlu, tak k přístupu studenta či učitele k výuce v on-line kurzech [3].

1.2 HTML

HTML je zkratka ze slov HyperText Markup Language. Je to značkovací jazyk určený pro vytváření internetových stránek (vzhled stránky je určován pomocí značek – tzv. tagů). Tyto stránky jsou po internetu přenášeny pomocí protokolu HTTP (HyperText Transfer Protocol). Vývoj jazyka HTML řídí konsorcium W3C (World Wide Web Consortium), v současné době je již delší dobu v platnosti verze HTML 4.01. Od klasického HTML se upouští a HTML 4.01 nahrazuje XHTML 1.0.

1.3 PHP

PHP je skriptovací jazyk šířený zdarma jako Open Source, vhodný pro vývoj internetových aplikací a vkládání do HTML. Tento jazyk je odvozen od programovacího jazyka C, Javy a Perlu. U zrodu PHP v roce 1994 stál Rasmus Lerdorf, který jej vytvořil pro evidování přístupu ke svým stránkám (odtud i původní název – Personal Home Page). Obsah zkratky PHP nyní zcela ztrácí na svém původním významu a doporučené označení celého systému je hypertextový preprocesor PHP (PHP: Hypertext Preprocessor). Vývoj PHP stále ještě není ukončen, poslední verzí je PHP 5.0.4.

1.4 Webový server Apache

Webový server je nezbytný, pokud chceme pracovat s databázemi a skriptovacím jazykem PHP. Princip je založen na modelu klient – server. Klientský program (většinou je to nějaký prohlížeč internetových stránek) posílá serveru požadavky. Těmito požadavky jsou příkazy skriptu. Server je zpracuje a zpět do klientského programu přichází jen výsledná data. Uživatel se tak nedozví přesný obsah skriptu, ale pouze výsledek vytvořený na základě jeho vyhodnocení. Tento princip je důležitý při zobrazování stránek, které dynamicky mění svůj obsah. Apache je jen jedním z mnoha webových serverů. Může běžet na mnoha platformách a je rovněž šířen jako Open Source. Je nejoblíbenějším a nejdoporučovanějším webovým serverem při programování stránek v jazyce PHP.

1.5 MySQL

MySQL je velmi oblíbený a často používaný databázový systém určený pro správu relačních databází, který vyvíjí Švédská firma MySQL AB. Pro neziskovou činnost je distribuován zdarma, navíc je také šířen jako Open Source. K manipulaci s daty používá jazyk SQL – Structured Query Language neboli strukturovaný dotazovací jazyk.

1.6 Interpret

Interpret je program, kterému dáváme jako vstupní data zdrojový kód programu napsaného ve vyšším programovacím jazyce. Interpret pak postupně načítá jeho jednotlivé části (příkazy) a interpretuje nebo také „vykládá si“ jejich význam. Nedochází k překladu a výstupem také není žádný binární proveditelný soubor, ale pouze interpretovaný výsledek. Výhoda použití interpretu spočívá především v rychlosti, důvodem proti jeho použití však může být opakované spouštění programu. V takovém případě je většinou výhodnější program přeložit než jej pokaždé znovu interpretovat příkaz po příkazu.

2 UŽIVATELSKÁ ČÁST

2.1 Instalace modulu

Instalace modulu Program je velmi snadná. Do adresáře *moodle/mod/* nainstalovaného systému Moodle zkopírujeme adresář *program*, který je na přiloženém CD v adresáři *modul_program/*. Do adresáře *moodle/lang/en/* systému Moodle je nutné dále zkopírovat jazykový soubor *program.php* z adresáře *modul_program/lang/en/* a do adresáře *moodle/lang/en/help* adresář *program*, který obsahuje soubory nápovědy v daném jazyce (na CD je tento adresář umístěn v adresáři *modul_program/lang/en/help*). Totéž je ještě nutné zopakovat pro češtinu (tedy analogicky nakopírovat soubory z přiloženého CD do adresářů *moodle/lang/cs* a *moodle/lang/cs/help*). Nakonec je potřeba navštívit stránku Správa modulů (*moodle » Správa » Nastavení » Správa modulů*) a systém Moodle by již měl sám modul rozpoznat a zařadit mezi ostatní.

2.2 Rozhraní pro učitele

2.2.1 Přidání nové činnosti Program

Na stránku zadání nového programu se učitel dostane stejným způsobem, jako při volbě jakékoliv jiné nové činnosti – z menu *Přidat činnost...* jednoduše vybere *Program*. Část stránky, která je určená k zadávání nového programu a která se po této volbě zobrazí, je vidět na obrázku (Obr. 1).

Položky *Název programu*, *Popis*, *Typ programu*, *Umožnit znovuodevzdání*, *Body*, *Největší velikost* a *Termín odevzdání* mají stejnou funkci jako v modulu *Úkol*, proto zde nebudou detailně popisovány. Některé z položek jsou opatřeny nápovědou (ta se zobrazí po kliknutí na obrázek otazníku ve žlutém kruhu), avšak jejich význam lze snadno vytušit i z názvu.

Jednou z nových položek zadávacího formuláře je zaškrťací políčko pojmenované *Volby příkazu diff*. Zaškrtnutím tohoto políčka dovolíte, aby při porovnávání výsledků (výsledků učitelova a studentova řešení programu) nebyl brán ohled na rozdíly v počtu prázdných řádků a velikosti mezer.

Přidání nové činnosti Program v týden 1 ?

Název programu:

Popis: 1 (8 pt) **B** **I** **U** **S** **x₂** **x₂**

Pište pozorně ?
Pokládejte správné dotazy ?
O HTML editoru ?

HTML cesta: `body » table.generalbox » tbody » tr » td.generalboxcontent » form#[object] » table » tbody » tr » td » input`

Typ programu: ?

Umožnit znovuodevzdání: ?

Volby příkazu diff: ☐ ?

Parametry učitelského příkladu: ?

Parametry studentského příkladu:

Body: ?

Největší velikost:

Termín odevzdání: -

Podklady k programu: ?

Obr. 1. Zadání nového programu

Dalšími novými položkami ve formuláři jsou pole určená pro definování parametrů programu (Parametry učitelského příkladu a Parametry studentského příkladu). Jedná se o parametry příkazového řádku – učitelský či studentský příklad bude interpretován s příslušnými parametry, budou-li zadány. Vzhledem k adresářové struktuře, která se vytváří pro odevzdané programy, je nutné, aby učitel zadával parametry pro studentské příklady ve tvaru `../source/vstupnisoubor`, pokud chce, aby studentský příklad četl vstupní data ze souboru `vstupnisoubor`, který je vytvořen učitelským příkladem nebo je načten spolu se zdrojovými soubory učitele.

Poslední přidanou položkou na této stránce jsou Podklady k programu. Je požadováno, aby učitel vložil zip archiv obsahující kompletní (fungující) program. Vložený archiv může navíc obsahovat soubor se vstupními daty nebo funkci, která jej vytvoří. Soubor se vstupními daty by pak měl být zadán jako parametr v patřičném poli formuláře

(viz. předchozí odstavec) nebo může jít o soubor s příponou .dat – předpokládá se však, že v odevzdávaných podkladech bude tento soubor jen jeden.

Nejvyšší prioritu při určování vstupního souboru mají parametry zadané ve formuláři. Nejsou-li zadány, je hledán soubor s příponou .dat. Neexistuje-li ani tento soubor, je program interpretován běžným způsobem bez vstupního souboru.

2.2.2 Kontrola odevzdaných prací

Pořadí:	Křestní jméno	Příjmení	Naposledy změněno ↑	Body
Uložit všechny mé odpovědi				
	Hana Dosoudilová Naposledy změněno: Pátek, 3. červen 2005, 12:13 (1 den 22 hodin pozdě)			
	 PrumernyProspech.c  cprgtorun.c  outcome.txt  pp.h  result.php  test-ne-pp-st.zip			Chyby
	Odpověď: Žádné body ▾			
	Julia Roberts Naposledy změněno: Pátek, 3. červen 2005, 12:12 (1 den 22 hodin pozdě)			
	 PrumernyProspech.c  cprgtorun.c  outcome.txt  pp.h  result.php  test-pp-st.zip			OK
	Odpověď: Žádné body ▾			

Obr. 2. Seznam odevzdaných programů

Kontrola odevzdaných prací probíhá stejně jako u modulu Úkol. Výpis všech odevzdaných programů (včetně všech potřebných informací k nim) se zobrazuje na stránce moodle » název_kurzu » Programy » název_programu » Odevzdané programy (na tuto stránku je možné se dostat například z již vytvořeného zadání daného programu). Informace o odevzdaných programech jsou téměř totožné s informacemi u modulu Úkol – jméno studenta, fotka studenta, datum odevzdání úkolu, odevzdané soubory, dále fotka učitele a jeho odpověď spolu s případným bodovým ohodnocením. Jedinou informací zobrazovanou navíc je sdělení o funkčnosti úkolu. Tato informace se vypisuje do sloupce, který se vytváří napravo od seznamu odevzdaných souborů. V případě odevzdání správně fungujícího programu se učiteli na tomto místě napíše tučným písmem OK, v případě zjištěných rozdílů se zde vytvoří text Chyby (v anglické jazykové verzi Faults), který je zároveň odkazem na soubor, jenž tyto chyby zobrazí. Příklad výpisu seznamu odevzdaných programů je na obrázku (Obr. 2).

2.3 Rozhraní pro studenty

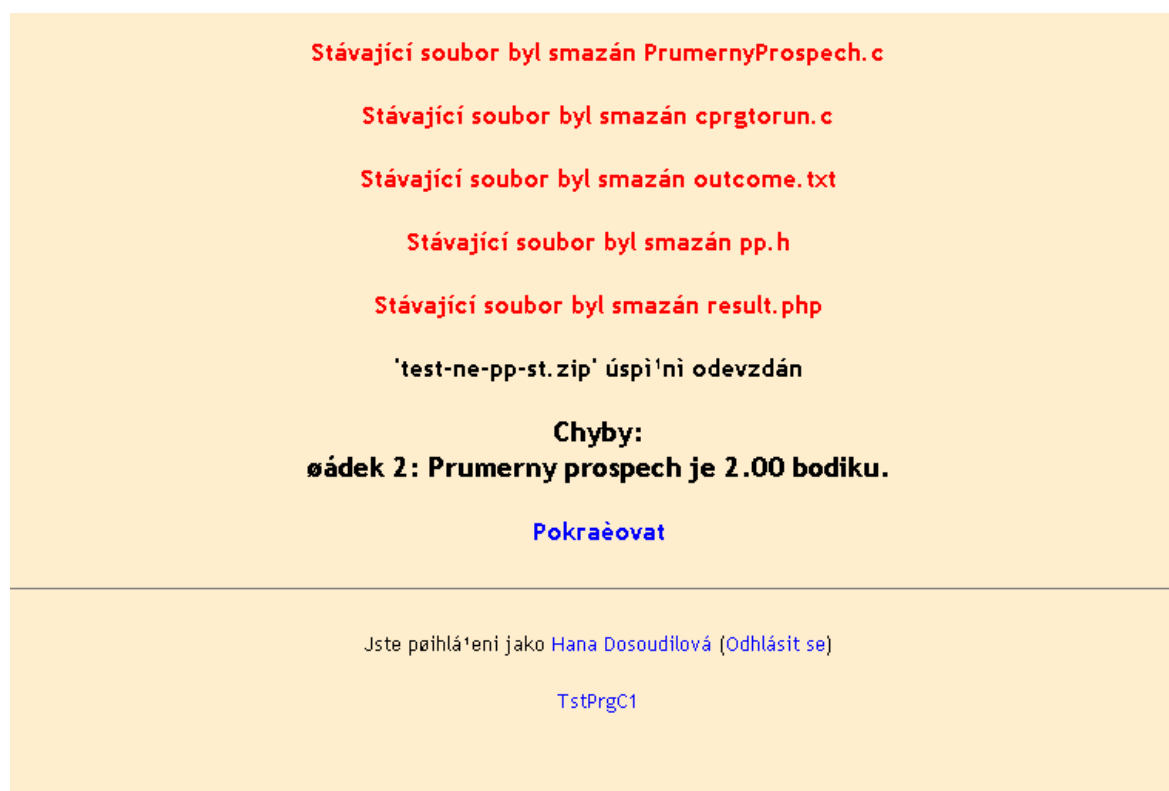
2.3.1 Odevzdání programu

Odevzdání programu je u studenta stejné jako v případě modulu Úkol. V dolní části stránky se zadáním daného programu se nachází jednoduché formulářové pole k tomu určené. Od studenta se předpokládá odevzdání zip archivu, který obsahuje minimálně jeden soubor s příponou .c, .C nebo .cpp a minimálně jeden soubor s příponou .h. Tyto soubory nesmí být prázdné.

2.3.2 Zobrazení výsledků



Obr. 3. Výpis výsledku při správném odevzdání



Obr. 4. Výpis výsledku při chybném odevzdání

Okamžitě po odevzdání svého programu vidí student výsledek – tučně zobrazený text informující o správném či chybném řešení (dle příslušné jazykové verze) nebo informativní hlášku, pokud nastal nějaký nečekaný stav a vyhodnocení nebylo možné provést. Příklady textů, které se mohou studentovi zobrazit po odevzdání programu, jsou zachyceny na obrázcích (Obr. 3 a Obr. 4).

3 PROGRAMÁTORSKÁ ČÁST

3.1 Výběr interpretu

Požadavkem při vytváření modulu Program bylo použití interpretu. Důvodem byla především ochrana proti odevzdání nežádoucích programů, které by mohly být na serveru spuštěny. Avšak výhodou tohoto řešení je i rychlost – překlad a spuštění trvá déle než interpretace.

Při výběru interpretu byl kladen důraz především na spolehlivost, korektní funkčnost na operačním systému Linux (na serveru na kterém běží výukový systém Moodle na Fakultě technologické je používán právě tento operační systém) a možnost interpretace jak jazyka C, tak i C++, aby bylo možné vytvořený modul využít v širším měřítku. Nebyl brán ohled na žádné další funkce či na velikost interpretu.

Výběr byl prováděn ze tří různých volně dostupných interpretů – Ch, CINT a EiC. Všechny z nich jsou určeny pro různé operační systémy, včetně požadovaného Linuxu. Avšak interpret EiC je určen pouze k interpretaci jazyka C. Interpret CINT zahrnuje přibližně 95 % ANSI C a 85 % C++, jeho nevýhodou však byly mírné potíže při spouštění některých programů (vyskytly se neznámé chyby) a také ponechávání nadbytečných souborů v adresáři *TEMP*. Nejspolehlivějším a téměř bezproblémovým se jevil poslední ze zmiňovaných – interpret Ch, který byl nakonec vybrán pro realizaci nového modulu.

3.2 Jednotlivé součásti modulu

V následujících kapitolách budou popsány jednotlivé soubory (převážně skripty) nového modulu Program. Jak již bylo zmíněno v Úvodu, tento modul je z velké části odvozen od modulu Úkol (Assignment), a proto zde nebudou vysvětlovány všechny funkce každého jednotlivého skriptu, ale pouze ty, které byly nově vytvořeny či byly nějakým způsobem pozměněny.

3.2.1 Ikona modulu

Pro nově vzniklý modul byla vytvořena zvláštní ikona znázorňující počítač. Důvodem pro vytvoření odlišné ikony byla lepší přehlednost všech aktivit v rámci jednotlivých kurzů vytvořených v systému Moodle. Počítač pak byl zvolen proto, že se jedná o výuku

programování ve vyšších programovacích jazycích, což se ve velké míře používá právě pro tvorbu počítačových programů.

3.2.2 Jazykové soubory

Pro nový modul byly vytvořeny dvě jazykové verze – česká a anglická. Pro oba tyto jazyky byl vytvořen soubor s názvem `program.php` obsahující pouhý seznam proměnných. Tyto proměnné jsou zapsané ve tvaru `$string['názevproměnné'] = 'hodnota proměnné – text';` a jsou volány z ostatních souborů modulu prostřednictvím funkcí `get_string()` a `print_string()`. Kromě odlišných textových řetězců specifických pouze pro modul Program se tyto jazykové soubory nijak neliší od jazykových souborů ostatních modulů.

Další soubory patřící do skupiny jazykových souborů jsou texty nápovědy. Všechny soubory nápovědy jsou prosté HTML stránky.

3.2.3 Stránka `config.html`

Stránka `config.html` zobrazuje formulář sloužící pouze k nastavení maximální velikosti odevzdávaného souboru v rámci nastavení vlastností modulu Program. Tento soubor zůstal oproti původnímu (přejatému z modulu Úkol) téměř beze změny s tím rozdílem, že se nastavuje hodnota proměnné `$CFG->program_maxbytes` a textové řetězce jsou přejímány z jazykových souborů pro modul Program.

3.2.4 Stránka `mod.html`

Stránka `mod.html` zobrazuje formulář, jenž slouží k zadání nového programu. Oproti formuláři pro zadávání úkolů obsahuje čtyři další formulářová pole, která jsou důležitá pro správnou činnost celého modulu a korektní interpretaci programů v závislosti na přáních učitele. Jedná se o pole Volby příkazu `diff`, Parametry učitelského příkladu, Parametry studentského příkladu a Podklady k programu. Význam těchto polí byl osvětlen již v uživatelské části. Vzhledem k tomu, že vytváření formulářů nebylo podstatou této práce a vytvoření těchto přidavných polí bylo realizováno pomocí původních definic ostatních polí formuláře, není zde tato část kódu detailněji popsána.

3.2.5 Soubor version.php

Dalším důležitým souborem modulu Program je skript version.php, který musí být povinně přítomen v každém modulu systému Moodle. V tomto souboru byla změněna hodnota proměnné `$module->version`, jenž vyjadřuje datum poslední modifikace modulu ve tvaru RRRRMMDD00. Tento údaj je důležitý při volání funkce `moodle_needs_upgrading()` a při jakýchkoliv významných změnách v modulu (především při změně databázových tabulek) musí být tato hodnota opět změněna.

3.2.6 Databázové soubory

Zadání programu bylo rozšířeno o položky, které musí být dlouhodobě uchovávány. Jedná se o informace nezbytně nutné k interpretaci programů, přičemž všichni musí mít stejné zadání a stejné podmínky. Také hodnocení studenta, které je vytvářeno automaticky funkcí definovanou ve skriptu `lib.php`, musí být uloženo.

K ukládání důležitých informací o modulech nebo o vykonaných činnostech uživatelů slouží na Fakultě technologické databáze MySQL. Jedná se například o odpovědi ve Fóru nebo údaje o odevzdaných úkolech. Samotné soubory, které tvoří odevzdané úkoly, však nejsou ukládány do databáze, ale do oddělené adresářové struktury.

Vzhledem k tomu, že se zvětšilo množství informací, které je nutné do databáze uložit, bylo nutné pozměnit i původní skripty vytvářející obě tabulky databáze – *program* a *program_submissions* (obě tabulky databáze mají ve svém názvu navíc prefix definovaný konfigurací celého systému Moodle).

Protože je na Fakultě technologické používána databáze MySQL, byly vytvořeny soubory pouze pro tuto databázi – `mysql.php` a `mysql.sql`. Do kódu obou těchto souborů byly přidány řádky, které při vzniku tabulky *program* vytvářejí sloupce *options*, *teacherparameters* a *studentparameters*. Kód určený k vytvoření databázové tabulky *program_submissions* byl rozšířen o položku přidávající sloupec *result*. Jako datový typ sloupce *options* bylo zvoleno velmi malé celé číslo – v databázi bude tato hodnota reprezentována buď číslem 1 nebo 0 (tedy logické ano či ne). Implicitně je tato hodnota nastavena na nulu. Jako datový typ ostatních položek byl zvolen text, protože v těchto sloupcích budou uloženy rozsáhlejší nečíselné informace.

3.2.7 Zálohovací soubory

Modul Úkol obsahuje dva skripty určené k zálohování dat z databáze. Jedná se o soubory `backup.lib` a `restore.lib`. V těchto souborech musely být příkazy pro práci se zápisem či čtením záznamů z databáze (popřípadě souborů) rozšířeny o další údaje. Důvodem bylo zvýšení počtu sloupců v databázových tabulkách *program* a *program_submissions* (tedy aby byly zálohovány také informace o hodnocení odevzdaných prací a o parametrech a volbách zadaných programů).

3.2.8 Skript `index.php`

K zobrazení všech programů, které byly zadány v rámci jednoho kurzu, slouží skript `index.php`, jenž je rovněž povinnou součástí každého modulu. Jedinou změnou v tomto skriptu bylo přepsání všech proměnných a všech parametrů volaných funkcí na proměnné a parametry příslušící modulu Program.

3.2.9 Skript `submissions.php`

Skript, který umožňuje vidět všechny odevzdané programy včetně všech potřebných informací a jenž je přístupný pouze učitelům, se nazývá `submissions.php`. Stejně jako u souboru `index.php` došlo pouze k přepsání názvů proměnných a parametrů funkcí. Počet vypisovaných údajů a jejich umístění se sice liší od výpisu odevzdaných úkolů, nicméně tento seznam je generován funkcí, jejíž definice je v souboru `lib.php`, proto nebylo nutné v souboru `submissions.php` provádět žádné větší změny.

3.2.10 Skript `view.php`

Skript `view.php` vypisuje informace o jednom zvoleném programu – jeho zadání a odevzdané soubory. V případě, že se jedná o vyučujícího, jsou to soubory, které byly poskytnuty jako podklady pro kontrolu studentských programů.

Do skriptu `view.php` proto musel být přidán kód, který v případě, že si stránku prohlíží učitel, vypíše seznam odevzdaných podkladů nebo napíše text informující o tom, že žádné podklady pro práci studentů odevzdány nebyly. Pro realizaci tohoto bloku kódu jsou použity funkce, které již byly v systému Moodle definovány dříve.

Ostatní změny souboru `view.php` se opět týkaly přepsání proměnných a parametrů funkcí.

3.2.11 Skript upload.php

V předposledním z popisovaných skriptů, upload.php, již bylo nutné provést větší úpravy. Tento skript zpracovává formulář sloužící pro odevzdání programů studenty.

Do skriptu bylo přidáno volání funkce `program_submission_testing()`, která se spustí poté, co byl vložený soubor úspěšně uložen do adresářové struktury. To znamená, že byl odevzdán soubor, jenž nebyl prázdný a zároveň nepřesahoval maximální povolenou velikost. Tato funkce, jak již její název napovídá, testuje odevzdaný program, avšak celý proces testování bude popsán u skriptu `lib.php`, kde je tato funkce definována.

Na základě výsledku testu, který je uložen v proměnné `$testresult`, se generuje text, který bude následně uložen do databáze jako výsledek automatického hodnocení (sloupec *result*). Uložení informací do databáze se uskuteční jen v případě, že všechny činnosti proběhly správně, což je ve skriptu `upload.php` ošetřeno několika podmínkami. Pokud došlo k chybě, informace uloženy nejsou a v prohlížeči se zobrazí varovné hlášení. Student je však v každém případě informován, což už bylo popsáno v uživatelské části.

Samozřejmě i u skriptu `upload.php` došlo k přepsání názvů proměnných a parametrů funkcí oproti souboru přejatému z modulu Úkol a některé proměnné byly přidány navíc.

3.2.12 Skript lib.php

Skript `lib.php` sám o sobě nic nevypisuje, ale přesto je téměř neustále volán ostatními skripty a je tedy jakoby jádrem celého modulu. Obsahuje všechny důležité funkce.

Kromě změn shodných s jinými skripty bylo nutné změnit nejpodstatnější funkce modulu – `program_add_instance($program)` a `program_update_instance($program)` – a dále vytvořit několik zcela nových funkcí. Následuje seznam těchto funkcí spolu s informacemi o provedených změnách a o jejich činnosti.

program_add_instance(\$program)

Funkce `program_add_instance($program)` slouží k zadání nového programu. Nejprve jsou zde uložena data týkající se zadání a posléze je testován vložený soubor (v tomto případě se jedná o funkční učitelovu verzi) obdobně jako je tomu při odevzdávání úkolů studenty. Tedy je zjišťováno, zda byl vůbec nějaký soubor odevzdán a zda má správnou velikost nepřekračující maximální povolenou hodnotu. Pokud soubor splňuje všechny podmínky,

je přemístěn do adresáře, který je vytvořen funkcí `program_sourcefile_area($program, $USER)`. Nakonec je zavolána funkce `program_source_processing($course, $USER, $mod, $program)`, jež bude popsána dále. V případě, že dojde k chybě, jsou vypsaná varovná hlášení.

program_update_instance(\$program)

K téměř stejnému procesu musí dojít i při obnově údajů o daném programu s tím rozdílem, že adresář vytvářený funkcí `program_sourcefile_area($program, $USER)` by měl již existovat, a tak se znovu nevytváří. Ošetření týkající se znovuvytvoření či nevytvoření tohoto adresáře obsahuje tato funkce sama ve své definici a z tohoto důvodu je volána stejně jako v případě prvního zadávání úkolu.

program_sourcefile_area_name(\$program, \$user)

Funkce `program_sourcefile_area_name ($program, $user)` je obdobou funkce `program_file_area_name($program, $user)` (resp. původní funkce definované v modulu Úkol). Všechny tyto funkce pouze tvoří název adresáře vhodný pro funkci `make_upload_directory()`, která vytváří adresářovou strukturu pro odevzdané práce. Od původní funkce se liší tím, že posledním adresářem v takto vytvořené cestě není adresář pojmenovaný identifikačním číslem uživatele, ale adresář *source*.

program_sourcefile_area(\$program, \$user)

Funkce `program_sourcefile_area($program, $user)` volá výše zmíněnou funkci `make_upload_directory()` s příslušným parametrem v závorce, kterým je právě funkce `program_sourcefile_area_name($program, $user)`.

program_print_submission(\$program, \$user, \$submission, \$teachers, \$grades)

Za zobrazení každého jednotlivého odevzdání na stránce `submissions.php`, která je k dispozici pouze učitelům, je zodpovědná funkce `program_print_submission($program, $user, $submission, $teachers, $grades)`. Úprava spočívala v jednoduché změně výsledné zobrazované tabulky, kde byl druhý řádek studentovy části tabulky rozdělen na dva sloupce, přičemž v levém byl ponechán původní obsah (tedy výpis odevzdaných souborů) a v pravém úzkém sloupci vedle tohoto výpisu se zobrazuje informace o tom, zda je odevzdané řešení správné či chybné. Zbytek tabulky byl ponechán beze změny.

program_print_sourcefiles(\$program, \$user)

Funkce `program_print_sourcefiles($program, $user)` je opět obdobou již dříve definované funkce – funkce `program_print_user_files($program, $user)` – s tím rozdílem, že vypisuje obsah adresáře *source* (tedy toho, který je určen k uložení učitelova řešení).

program_delete_sourcefiles(\$program, \$user, \$exception)

Pro funkci `program_delete_sourcefiles($program, $user, $exception)` platí v podstatě totéž co pro předchozí funkci. V tomto případě se však nejedná o výpis obsahu adresáře, ale o vymazání jeho obsahu. Smazány jsou všechny soubory až na výjimku uvedenou v proměnné `$exception`.

program_source_processing(\$course, \$user, \$mod, \$program)

U funkce `program_source_processing($course, $user, $mod, $program)` je nutné si vysvětlit celý princip, na kterém modul Program vlastně funguje. Hlavní myšlenkou celé realizace je porovnávání výsledků získaných z učitelova a studentova programu. Porovnání je realizováno ve funkci, která bude zmíněna následně (protože v momentě, kdy jsou odevzdávány podklady, není ještě co porovnávat). Výsledky programů se získají pomocí interpretu (o němž byla řeč v úvodu Programátorské části) a uloží se do zvláštního souboru – v případě učitelova odevzdání je to vždy textový soubor `model.txt`.

Funkce `program_source_processing($course, $user, $mod, $program)` tedy slouží ke zpracování učitelova vzorového programu tak, aby bylo možné jeho výsledek využít k hodnocení studentova vypracování. Na začátku této funkce je načtena globální proměnná `$CFG` a dále knihovna `pclzip.lib.php`, která je již součástí systému Moodle a obsahuje mnoho užitečných funkcí pro práci se zip archivy. Následuje kontrola, zda je v odevzdaném archivu alespoň jeden neprázdný soubor s příponou `.h` a alespoň jeden neprázdný soubor s některou z přípon `.c`, `.C` nebo `.cpp`. Názvy všech souborů s některou z těchto přípon jsou ukládány do zvláštního souboru s názvem `cprgtorun.c` spolu s direktivou *pragma*. Je-li obsah archivu v pořádku, je rozbalen a soubor `cprgtorun.c` je předán interpretu Ch jako parametr. To se může udát různými způsoby, v závislosti na tom, co zapsal učitel do formuláře určeného pro zadání programu.

Spuštění interpretu se vykoná díky prováděcím operátorům – obráceným apostrofům. Pokud je v PHP skriptu něco uvedeno mezi dvěma obrácenými apostrofy, pokusí se PHP provést obsah uzavřený mezi těmito apostrofy jako příkaz shellu. V našem případě tedy

PHP spouští interpret Ch, jemuž je jako parametr předán soubor cprgtorun.c a výsledek je přesměrován do souboru uloženém v proměnné \$outputfile – tedy v tomto případě do souboru model.txt. Pokud učitel nevložil soubor s příponou .zip nebo archiv neobsahoval požadované typy souborů, vrací funkce hodnotu false, což se ve funkcích program_add_instance(\$program) a program_update_instance(\$program) projeví chybovou hláškou.

program_submission_testing(\$program, \$user)

Poslední funkce – funkce program_submission_testing(\$program, \$user) – slouží k ohodnocení studentova vypracovaného programu v závislosti na porovnání s učitelovým řešením. Úvodní příkazy této funkce jsou obdobné jako u té předchozí – načtení knihovny pro práci se zip archivy a testování odevzdaného souboru. Navíc je pouze zjišťování, zda jsou k dispozici zdrojové kódy učitele (tedy především soubor model.txt, se kterým bude porovnáván výsledek studentova odevzdání) a zda je v učitelově odevzdání k dispozici vstupní soubor s příponou .dat či nikoliv. Archivní soubor je v případě úspěšného testování rozbalen, zdrojové soubory s příponami .c, .C a .cpp jsou stejným způsobem jako u předchozí funkce ukládány do souboru cprgtorun.c a nakonec je tento soubor interpretován (opět jednou z několika možností dle zadání programu). Výstup je tentokrát přesměrován do souboru s názvem outcome.txt.

V této fázi vykonávání funkce již existují dva výsledky různých odevzdání programů a je třeba je porovnat. Za tímto účelem byl použit příkaz diff, který je dostupný v operačním systému Linux. Pro spuštění tohoto příkazu jsou opět použity prováděcí operátory a výstup je uložen do proměnné \$diffoutput. Obsah této proměnné je následně zapsán do souboru s názvem result.php. Byly-li však výsledky obou programů shodné, proměnná \$diffoutput je prázdná a do souboru result.php nebylo zapsáno nic. Nastane-li taková situace, zapíše se do tohoto souboru text informující o správném řešení. V opačném případě je zapsaný obsah upraven do srozumitelnější podoby. Na tento soubor se totiž odvolává odkaz v informacích o odevzdaném programu (skript submissions.php) v případě, že při porovnání výstupů programů byly nalezeny rozdíly. Navíc obsah souboru result.php je vypsan také studentovi do prohlížeče ihned po odevzdání jeho programu (text je skriptu upload.php předán pomocí proměnné \$return). Proto je nutné, aby byl obsah tohoto souboru snadno čitelný a srozumitelný. Obsah je jednoduše upraven tak, že se celý text

souboru načte do proměnné, poté se po jednotlivých řádcích upraví a nakonec je původní soubor přepsán.

V závislosti na činnostech provedených touto funkcí je vrácena buď hodnota 0 nebo některý z textových řetězců. Podle vrácené hodnoty jsou vyhodnocovány další podmínky, které jsou uvedeny ve skriptu upload.php, jak již bylo uvedeno výše.

ZÁVĚR

V této práci jsou zdokumentovány veškeré postupy vedoucí ke vzniku nového modulu určeného pro výukový systém Moodle – modulu Program. Změnou databázových tabulek a skriptů modulu Úkol bylo dosaženo možnosti interpretovat odevzdané programy, porovnat je se vzorovým učitelským řešením a uložit (případně zobrazit) výsledky pro potřeby studenta i učitele. Formulář určený k zadání nového programu navíc obsahuje několik nových položek, které rozšiřují možnosti interpretace a porovnávání programů. Program je možné interpretovat se vstupními daty i bez nich a také je možné povolit použití volitelných položek příkazu diff.

Vzhledem k tomu, že je k interpretaci programů používán program Ch, který je spolehlivý a který umožňuje interpretaci programů napsaných jak v jazyce C tak i v jazyce C++, může být nový modul Program uplatněn ve větším počtu typů programovacích kurzů.

Tento modul byl stejně jako ostatní moduly systému Moodle vytvořen jako jazykově nezávislý a byly pro něj zároveň napsány dvě jazykové verze – česká a anglická.

Modul byl úspěšně nainstalován a zprovozněn na testovacím výukovém systému Moodle na Fakultě technologické. Testování proběhlo na několika příkladech, z nichž dva jsou uloženy na přiloženém CD v adresáři *testovací příklady*. Skutečným testem však bude instalace tohoto modulu přímo do již používaného systému Moodle a jeho nasazení do výuky v příštím semestru.

SEZNAM POUŽITÝCH ZDROJŮ

Monografické publikace

- 1) KOSEK, Jiří. PHP tvorba interaktivních internetových aplikací Podrobný průvodce. 1. vyd. Praha: Grada, 1998 492 s. ISBN: 80-7169-373-1
- 2) SCHLOSSNAGLE, George. Pokročilé programování v PHP 5. 1. vyd., Brno: Zoner Press, 2004. 640 s. ISBN: 80-86815-14-5

WWW stránka

- 3) Moodle [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://moodle.org/>>.
- 4) W3C – WORLD WIDE WEB Consortium [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://www.w3.org/>>.
- 5) php [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://www.php.net/>>.
- 6) The Apache Software Foundation [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://www.apache.org/>>.
- 7) MySQL [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://mysql.com/>>.
- 8) Softintegration [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://www.softintegration.com/>>.
- 9) The CINT C/C++ Interpreter [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://root.cern.ch/root/Cint.html>>.
- 10) eic [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://eic.sourceforge.net/>>.
- 11) Builder [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://www.builder.cz/>>.
- 12) Linuxsoft.cz [online]. [cit. 2006-06-03]. Dostupná z URL: <<http://www.linuxsoft.cz/>>.

SEZNAM OBRÁZKŮ

Obr. 1. Zadání nového programu	10
Obr. 2. Seznam odevzdaných programů	11
Obr. 3. Výpis výsledku při správném odevzdání	13
Obr. 4. Výpis výsledku při chybném odevzdání	13