

Automatické hodnocení úkolů v kurzech programování

Automatic assessment of assignments

Miroslav Benovič

Bakalářská práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Miroslav BENOVIČ**
Osobní číslo: **A07009**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Automatické hodnocení úkolů v kurzech programování**

Zásady pro vypracování:

Zásady pro vypracování:

1. Prostudujte doporučenou literaturu a další informační zdroje.
2. Upravte modul tak, aby fungoval i v Moodle 1.9 a vyšších.
3. Zaměřte se také na odstranění bezpečnostních a návrhových chyb, které jsou zmiňovány v oponentním posudku předchozí diplomové práce na toto téma.
4. Vytvořte několik ukázkových programů vhodných k otestování funkčnosti modulu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. BUCHNER, ALEX. Moodle Administration : An administrators guide to configuring, securing, customizing, and extending Moodle. [s.l.] : Packt Publishing Ltd., 2008. 357 s. ISBN 978-1-847195-62-3.
2. BERND , Michael ; ÖGGL, Kofler. PHP 5 a MySQL 5 : Průvodce webového programátora. [s.l.] : Computer Press, 2007. 608 s. ISBN 978-80-251-1813-9, 9788025118139.
3. VRÁNA, Jakub. 1001 tipů a triků pro PHP. [s.l.] : Computer Press, 2011. 456 s. ISBN 978-80-251-2940-1.
4. BORONCZYK, Timothy, et al. PHP 6, MySQL, Apache. [s.l.] : Computer Press, 2009. 816 s. ISBN 978-80-251-2767-4.
5. PECINOVSKÝ, Rudolf. OOP : Naučte se myslet a programovat objektivě. [s.l.] : Computer Press, 2010. 576 s. ISBN 978-80-251-2126-9.
6. NEMETH, Evi; SNYDER, Garth; HEIN, Trent R. Linux : Kompletní příručka administrátora, 2. aktualizované vydání. [s.l.] : Computer Press, 2008. 976 s. ISBN 978-80-251-2410-9.

Vedoucí bakalářské práce:

Ing. Tomáš Dulík

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011


prof. Ing. Vladimír Vašek, CSc.
děkan

L.S.


prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Pro potřeby kurzů programování na FAI bylo třeba vytvořit do modulárního školního výukového systému Moodle modul, který by automaticky hodnotil odevzdané úkoly v různých programovacích jazycích a dokázal by na základě určité míry podobnosti odhalovat plagiáty.

Modul splňující výše uvedené požadavky byl v roce 2005 vytvořen v rámci bakalářské práce studentkou Veronikou Vaškovou a v roce 2006 byl jako diplomová práce zdokonalen studentem Bc. Pavlem Jurou. Modul ale po nějakém čase přestal fungovat z důvodu změn API a struktury databáze v systému Moodle, které způsobil neustálý vývoj tohoto systému. Bylo potřeba tento modul přepsat, aby fungoval i v současné verzi, nebo jej vytvořit znovu. V této bakalářské práci je popsáno, které prostředky, postupy, moduly a zdroje jsem pro splnění zadání použil.

Klíčová slova: Moodle, plagiát, detekce plagiátů, programování, automatické hodnocení

ABSTRACT

For needs of programming classes on FAI there was needed to create a Moodle module in the modular school educational system, which would automatically evaluate homeworks in different programming languages and be able to discover plagiarisms on the basis of certain measure of similarity.

Satisfactory module was created back in 2005 as a part of a bachelor thesis by student Veronika Vašková and in 2006 it was improved as a part of a thesis by student Bc. Pavel Jura. But this module stopped working because of API and database structure changes in the Moodle system caused by constant development of this system. It was needed to rewrite this module to have it working in the current version or create it again.

In this bachelor thesis there are described methods, processes, modules and resources I used to fulfill the entry.

Keywords: Moodle, plagiarism, antiplagiarism, programming, Automatic assessment

Děkuji mému vedoucímu práce Ing. Tomáši Dulíkovi za pomoc, rady a názory, které přispěly k dokončení této práce. Dále bych chtěl poděkovat vývojářům Sergey Butakov a Sun Zhigang, kteří mi pomohli s instalací a odzkoušení jejich modulů.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1.1 SYSTEM MOODLE.....	11
1.1.1 Licence GNU GPL.....	11
1.2 PHP.....	12
1.2.1 Výhody a nevýhody	12
1.2.2 Licence PHP	12
1.3 HTTP SERVER	13
1.4 MYSQL	13
1.4.1 Způsob ukládání dat	13
1.4.2 Úložiště dat (enginy).....	14
1.5 ALTERNATIVNÍ ŘEŠENÍ	14
1.5.1 Modul Crot.....	14
1.5.1.1 Instalace	15
1.5.1.2 Využití modulu Crot.....	15
1.5.2 Modul Anti-plagiarism.....	16
1.5.2.1 Instalace	16
1.5.2.2 Využití modulu Anti-plagiarism.....	16
1.5.2.3 Výhody a nevýhody	17
1.5.3 Modul Online judge	17
1.5.3.1 Metoda online zpracování.....	17
1.5.3.2 Metoda lokálního zpracování.....	17
1.5.3.3 Instalace	18
1.5.3.4 Využití v praxi	18
1.6 MODUL ACA	19
1.6.1 Automatické hodnocení	19
1.6.2 Odhalování plagiátů	19
1.6.2.1 Program YAP3.....	19
II PRAKTICKÁ ČÁST	21
1.7 ANALÝZA ŘEŠENÍ	22
1.8 DŮLEŽITÉ ZMĚNY V MOODLE 1.9 A V MODULU ACA	22
1.8.1 Deklarace verze modulu.....	22
1.8.2 Instalační soubory	22
1.8.3 Kontextový systém.....	24
1.8.3.1 Přidělování rolí	24
1.8.3.2 Nutné změny v modulu.....	25
1.8.4 Kódování v UTF-8	25
1.8.5 Globální proměnné.....	26
1.8.6 Názvy funkcí	26
1.8.7 Zabezpečení souboru pd4m.make.php	26
1.8.8 Oprava chyb	26
1.9 INSTALACE A KONFIGURACE MODULU ACA.....	27
1.9.1 Požadavky modulu pro provoz.....	27
1.9.2 Návod k instalaci softwaru potřebného pro funkci modulu	28

1.9.2.1	Instalace Apache + MySQL + PHP	28
1.9.2.2	Instalace interpretu CH	28
1.9.2.3	Instalace kompilery pro jazyk Pascal.....	29
1.9.3	Instalace samotného modulu ACA.....	29
1.9.3.1	Nakopírování modulu	30
1.9.3.2	Konfigurace cest v souboru lib.php	30
1.9.3.3	Konfigurace cest v BASH skriptech.....	31
1.9.3.4	Nastavení připojení k databázi v BASH skriptech	31
1.9.3.5	Varianta bez konfigurace	32
1.10	POPIS MODULU Z POHLEDU UČITELE.....	32
1.10.1	Způsob bodování	33
1.10.1.1	Ruční hodnocení.....	33
1.10.1.2	Přímé body	34
1.10.1.3	Váhové funkce	34
1.10.1.4	Rovnoměrné rozložení	34
1.10.2	Testovací vektory	34
1.10.3	Spuštění testů	35
1.10.4	Výsledky testů	36
1.10.5	Kontrola plagiátorství.....	36
1.11	POPIS MODULU Z POHLEDU STUDENTA	37
ZÁVĚR		39
ZÁVĚR V ANGLIČTINĚ.....		40
SEZNAM POUŽITÉ LITERATURY.....		41
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		43
SEZNAM OBRÁZKŮ		45
SEZNAM TABULEK.....		46
SEZNAM PŘÍLOH.....		47

ÚVOD

E-learningové systémy jako je Moodle jsou rychle se rozvíjející aplikace, které umožňují vytvářet elektronické kurzy na internetu. Protože internet se také stále rozvíjí a dnes už je běžnou součástí téměř každé domácnosti, mají online výukové systémy velkou budoucnost. Pro většinu vysokých škol je v dnešní době využití takovýchto systémů nutnost a určitě by našly uplatnění i na některých středních odborných školách.

Na Fakultě aplikované informatiky běží systém Moodle už několik let a využívá se nejen ve všech předmětech vyučujících programování, ale i v předmětech teoretických. Často slouží jako prostředím mezi profesory a studenty, kurzy obsahují souhrn všech studijní materiálů a přednášek, nebo se využívá pro interaktivní testy, kdy dokáže sám sestavit a prokombinovat podle zadaných kritérií testové otázky, sám je opraví a ohodnotí.

V kurzech programování v jazyce C/C++, Pascal a PHP je tento systém v kombinaci s modulem pro automatické hodnocení úkolů a odhalování plagiátů velmi silným nástrojem, který díky velké automatizaci ušetří vyučujícím spoustu až zbytečné práce, kdy by museli každý program ručně stáhnout, kompilovat a testovat různé vstupní data pro ověření funkčnosti. Ušetřený čas můžou nyní věnovat třeba konzultacím se studenty a kvalita výuky se mnohonásobně zvýší. Automatickou kontrolu ocenění i studenti, protože uvidí výsledky z testování programu ihned po odevzdání a mají možnost program opravit a znovu odevzdat, tedy pokud jim to vyučující v konfiguraci kurzu dovolí.

Modul Aca, který byl původně vytvořen studentkou Veronikou Vaškou v roce 2005 uměl pouze automaticky hodnotit úkoly. O rok později byl zdokonalen studentem Bc. Pavlem Jurou o vyhledávání plagiátů pomocí programu YAP3. Modul fungoval původně v Moodle verze 1.6 a po upgradu na vyšší verzi 1.9 přestal fungovat z důvodu změn v samotném systému Moodle. Cílem mé bakalářské práce bylo systém přepsat pro verzi PHP 5 a Moodle 1.9 a vyšší, aby mohl být použitý v kurzech programování na UTB.

I. TEORETICKÁ ČÁST

1.1 Systém Moodle

Moodle [čti múdl] je open source software určený pro podporu prezenční i distanční výuky prostřednictvím online kurzů dostupných na Internetu. Jedná se o Course Management System (CMS), někdy též označovaný jako Learning Management System (LMS) nebo Virtual Learning Environment (VLE). Stal se oblíbeným nástrojem učitelů i studentů z celého světa. Moodle je třeba nainstalovat na webový server, buď na vlastní nebo poskytovatele webových služeb. [1]

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. [2]

1.1.1 Licence GNU GPL

Moodle je poskytnut jako Open Source software, který spadá pod obecnou veřejnou licenci GNU General Public License (v českém překladu „Všeobecná veřejná licence GNU“). Bývá uváděno, že 65 % až 70 % všech počítačových programů evidovaných v databázi SourceForge.net je distribuováno právě za těchto licenčních podmínek. Pokud přijmeme tuto licenci, tak máme právo dílo (v našem případě software Moodle) libovolně upravovat, kopírovat, rozšiřovat a vytvářet nové vlastní subverze odvozené z této hlavní verze.

Distribuční práva zaručená GPL pro modifikované verze díla nejsou bez podmínek. Když někdo šíří GPL dílo se svými vlastními modifikacemi, požadavky na distribuci celého díla nemohou být jakkoli zvětšeny oproti požadavkům, které jsou v GPL. Tento požadavek je znám jako copyleft. [3]

1.2 PHP

PHP – tato zkratka znamená „PHP: Hypertext Preprocessor“, což je rekurzivní programátorský vtíp. PHP je velmi oblíbený a rychle se vyvíjející skriptovací jazyk pro vytváření dynamických webových stránek. Jeho syntaxe je odvozena od několika programovacích jazyků C, Pascal, Perl, nebo Java. Kód jazyka PHP se vkládá přímo do HTML, nebo XHTML stránky a je uzavřen mezi tagy `<?php` a `?>`. Je spouštěn přímo na straně serveru a k uživateli se dostane pouze jeho výstup.

1.2.1 Výhody a nevýhody

Výhody jazyka PHP výrazně předčí jeho nevýhody. Mezi jeho velké klady patří licence (viz. bod 1.2.2). Za vývojem stojí firma Zend, která pravidelně vydává aktualizace, nové verze a záplaty. PHP je nezávislé na platformě, je možné jej používat na různých operačních systémech a podporuje velké množství knihoven pro práci s texty, obrázky, soubory a propojení s databázemi.

Jedna z hlavních nevýhod PHP je v jeho neukázněném vývoji a v nejednotnosti. Jednotlivé funkce nedodržují společnou koncepci, např. pro pojmenování funkcí je používána celá řada stylů jako například jméno_funkce i JménoFunkce. Podobné funkce mají často odlišné pořadí parametrů a podobně. Jde o zřejmý protiklad knihoven v Javě, u nichž je na první pohled patrné, že byly vytvářeny s rozmyslem a podle společných zásad.

PHP po programátorovi nevyžaduje, aby jeho kód byl kvalitní. Umožňuje velkou škálu "špinavých triků", které mnozí s oblibou využívají. [4]

1.2.2 Licence PHP

PHP patří pod licenci *PHP license*. Vztahuje se na předávání všech zdrojových a binárních kódů PHP a na vytváření jeho odvozenin. PHP licence je poměrně volná, jen zakazuje použití názvu PHP v názvu jeho odvozeniny.

Na skripty vytvořené v PHP se licence nevztahuje a můžeme je libovolně šířit s libovolnou licencí. Neplatilo by to jen v případě, když bychom naše kódy šířili i s PHP jako jeden velký balíček. [19]

1.3 HTTP Server

Pro umožnění přístupu k serveru uživatelům z internetu je potřeba mít nainstalovaný nějaký http server, který zajišťuje obsluhu požadavků mezi klientem a serverem. V dnešní době máme na výběr z velkého množství http serverů jako například Apache, Lighttpd [čti lajty] nebo Internet Information Services (IIS) od Microsoftu. Většina z těchto webových serverů pracuje podobně, komunikují nejčastěji na portu 80, nebo při použití šifrovaného spojení SSL na portu 443. Rozdíly jsou ve způsobu a rychlosti zpracování požadavků klientů.

Pro provoz e-learningového systému Moodle je na fakultě FAI použit webový server Apache. Jedná se o multiplatformní webový server s otevřeným kódem pro operační systémy BSD, GNU/Linux, Mac OS X, Solaris a Microsoft Windows. Dnes je jedním z nejrozšířenějších webových serverů vůbec, uvádí se, že jej používá více než 70% všech serverů.

1.4 MySQL

MySQL je databázový systém, původně vytvořený švédskou firmou MySQL AB (proto je výchozí kódování všech tabulek nastaveno na hodnotu *latin1_swedish_ci*), nyní vlastněný společností Oracle. Komunikace s MySQL databází probíhá pomocí standardizovaného dotazovacího jazyka pro práci s relačními databázemi SQL (Structured Query Language). MySQL byla od začátku zaměřována na rychlost, takže jí chybělo mnoho praktických vlastností jako trigger, procedury či pohledy (view). Díky tomu, že je multiplatformní, rychlá, výkonná a navíc volně šiřitelná, má vysoký podíl mezi v současné době používanými databázemi. Velmi oblíbená je kombinace Linux + Apache + MySQL + PHP jako základ webového serveru. Tato kombinace je dobře známá pod zkratkou LAMP a byla použita i mnou při testování modulu na virtuálním serveru. [5] [6]

1.4.1 Způsob ukládání dat

Každá databáze v MySQL je tvořena z jedné nebo více tabulek, které mají řádky a sloupce, v řádcích rozeznáváme jednotlivé záznamy (každý řádek je jeden záznam). Sloupce mají jméno a uvozují datový typ jednotlivých polí záznamu. Práce s databázemi, tabulkami a daty se provádí pomocí příkazů, respektive dotazů.

Každá databáze se v MySQL ukládá do souboru, který se nachází v datovém adresáři MySQL. Každá tabulka v databázi má svou definici uloženou v souboru se stejným jménem a příponou *.frm*. Data jsou uloženy v souborech *.MYD* a *.MYI*. [7]

1.4.2 Úložiště dat (enginy)

MySQL podporuje několik datových úložišť, které se chovají jako samostatné moduly a lze je doinstalovat. Nejznámější a nejpoužívanější enginy jsou MyISAM a InnoDB, pro webové aplikace se více hodí InnoDB. Úložiště můžeme v jedné databázi kombinovat, to znamená, že každá tabulka může používat jiné. [19]

- **MyISAM** – nenáročné a rychlé úložiště, umožňuje vytvářet indexy a fulltextové vyhledávání.
- **InnoDB** – dobře zvládá práci více uživatelů najednou a podporuje transakce.
- **HEAP** – data se ukládají do paměti, používáno pro ukládání sessions.
- **Archive** – nepodporuje indexy a editování dat, všechna data při ukládání komprimuje, proto se hodí hlavně pro ukládání logů.

1.5 Alternativní řešení

Na internetu je několik již hotových modulů, které by teoreticky bylo možné použít a popřípadě lehce poupravit pro splnění zadání této práce. V následujících podkapitolách jsou popsány možné alternativní řešení, které jsem nainstaloval a odzkoušel. U každého modulu jsou shrnuty výhody a nevýhody a hlavní důvody, kvůli kterým jsem se rozhodnul je nevyužít.

1.5.1 Modul Crot


Tento modul od autorů Sergey Butakov a Svetlana Kim je zaměřen na vyhledávání plagiátů. V první fázi odhaluje, které úkoly byly zkopírovány od ostatních účastníků kurzu a ve druhé fázi vyhledává pomocí API vyhledávače Bing podobnosti na webu. Dokáže odhalit z které webové stránky byly texty zkopírovány a výsledky zobrazí i s procentuálním hodnocením v přehledné tabulce. Modul existuje ve verzi pro Moodle 1.9.x i pro 2.x a jeho poslední aktualizace je z data 20.3.2011.


Course A

Home ► Courses ► CA ► Essay1 ► Anti-Plagiarism - Assignments

Assignments with similarity score less than 0% are not displayed

Student name	Similar assignments																																	
Student 1	<table border="1"> <thead> <tr> <th>Name</th> <th>Course</th> <th>Similarity score</th> </tr> </thead> <tbody> <tr> <td>http://dic.academic.ru/dic.nsf/ruwiki/64</td> <td>Web document</td> <td>100 %</td> </tr> <tr> <td>http://ru.wikipedia.org/wiki/%D0%98%D1%8</td> <td>Web document</td> <td>100 %</td> </tr> <tr> <td>http://dic.academic.ru/dic.nsf/ruwiki/11</td> <td>Web document</td> <td>100 %</td> </tr> <tr> <td>http://www.world-wide-gifts.ru/index.php</td> <td>Web document</td> <td>100 %</td> </tr> <tr> <td>http://www.otpusk.com/ref/kr/info-histor</td> <td>Web document</td> <td>93.94 %</td> </tr> <tr> <td>http://ru.pandapedia.com/wiki/%D0%AE%D0%</td> <td>Web document</td> <td>90.91 %</td> </tr> <tr> <td>http://www.carstock.ru/Dictionary/%D0%A0</td> <td>Web document</td> <td>90.91 %</td> </tr> <tr> <td>http://ru.science.wikia.com/wiki/%D0%98%</td> <td>Web document</td> <td>87.88 %</td> </tr> <tr> <td>http://unitut.ru/countrylist/korea-sout</td> <td>Web document</td> <td>87.88 %</td> </tr> <tr> <td>http://www.gnimweb.ru/</td> <td>Web document</td> <td>48.48 %</td> </tr> </tbody> </table>	Name	Course	Similarity score	http://dic.academic.ru/dic.nsf/ruwiki/64	Web document	100 %	http://ru.wikipedia.org/wiki/%D0%98%D1%8	Web document	100 %	http://dic.academic.ru/dic.nsf/ruwiki/11	Web document	100 %	http://www.world-wide-gifts.ru/index.php	Web document	100 %	http://www.otpusk.com/ref/kr/info-histor	Web document	93.94 %	http://ru.pandapedia.com/wiki/%D0%AE%D0%	Web document	90.91 %	http://www.carstock.ru/Dictionary/%D0%A0	Web document	90.91 %	http://ru.science.wikia.com/wiki/%D0%98%	Web document	87.88 %	http://unitut.ru/countrylist/korea-sout	Web document	87.88 %	http://www.gnimweb.ru/	Web document	48.48 %
Name	Course	Similarity score																																
http://dic.academic.ru/dic.nsf/ruwiki/64	Web document	100 %																																
http://ru.wikipedia.org/wiki/%D0%98%D1%8	Web document	100 %																																
http://dic.academic.ru/dic.nsf/ruwiki/11	Web document	100 %																																
http://www.world-wide-gifts.ru/index.php	Web document	100 %																																
http://www.otpusk.com/ref/kr/info-histor	Web document	93.94 %																																
http://ru.pandapedia.com/wiki/%D0%AE%D0%	Web document	90.91 %																																
http://www.carstock.ru/Dictionary/%D0%A0	Web document	90.91 %																																
http://ru.science.wikia.com/wiki/%D0%98%	Web document	87.88 %																																
http://unitut.ru/countrylist/korea-sout	Web document	87.88 %																																
http://www.gnimweb.ru/	Web document	48.48 %																																

Global plagiarism detection is supported by Bing search engine 

 Moodle Docs for this page
You are logged in as Admin User (Logout)

Purge all caches

Obr. 1. Náhled výsledků detekce plagiátorství modulu Crot

1.5.1.1 Instalace

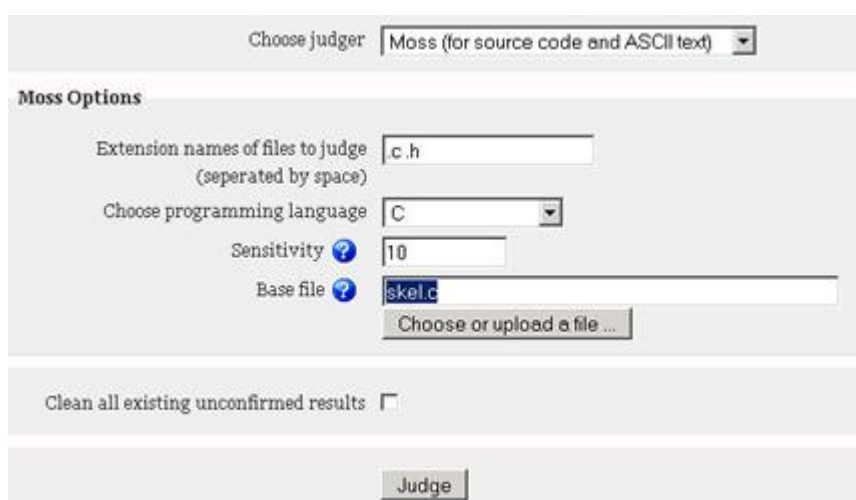
Instalace modulu je jednoduchá, stačí se řídit informacemi v přibaleném souboru *Readme.txt*. Před samotnou instalací je potřeba se registrovat na stránkách vývojářů vyhledávače Bing. Tím získáme unikátní App ID, přes které bude náš modul komunikovat s vyhledávačem Bing a které zadáme při konfiguraci modulu.

1.5.1.2 Využití modulu Crot

Po bližším prozkoumání toho modulu je zřejmé, že pracuje pouze s texty, po doinstalování doplňku Antiword do PHP i se soubory aplikace MS Word. Jednalo by se o silný nástroj pro fakulty humanitního nebo sociálního zaměření, kde studenti píšou dlouhé textové seminární práce nebo eseje, které často bez rozmyslu kopírují z webu. Pro odhalování plagiátů v kurzech programování však tento modul vhodný není.

1.5.2 Modul Anti-plagiarism

Modul Anti-plagiarism od autora Sun Zhigang je zaměřený na odhalování plagiátů ve zdrojových kódech jazyků C, C++, Java, C#, Python, Visual Basic, Javascript a dalších. Jedná se o modul typu *block*, který využívá online technologie MOSS. Modul vyžaduje operační systém Linux, nebo Windows (vyžaduje Cygwin s podporou Perl) a Moodle verze 1.9.x. Učitel si může u každého úkolu kdykoliv spustit kontrolu odevzdaných úkolů. Všechny úkoly jsou perlowským skriptem odeslány ke zpracování do systému MOSS, který vrátí výsledky procentuální shody odeslaných souborů a adresy URL, na kterých je možno si výsledky detailně prohlédnout i s vyznačením shodných řádků.



Choose judger: Moss (for source code and ASCII text)

Moss Options

Extension names of files to judge (seperated by space): c.h

Choose programming language: C

Sensitivity: 10

Base file: skel.c

Choose or upload a file ...

Clean all existing unconfirmed results:

Judge

Obr. 2. Náhled rozhraní k detekci plagiátorství modulu Anti-plagiarism

1.5.2.1 Instalace

V první řadě ověříme, jestli je direktiva *allow_url_fopen* v konfiguračním souboru *php.ini* nastavená na hodnotu *on*. Poté rozbalte archiv modulu, přejmenujte složku ve které se nachází na *anti_plagiarism* and vložte ji do *moodle_home/blocks/*. Přihlaste se do Moodle jako admin a dál se řiďte instrukcemi, které vám modul zobrazí.

1.5.2.2 Využití modulu Anti-plagiarism

Tento modul má své výhody, které jsou zároveň i jeho nevýhody. Každý odevzdaný program posílá k externí kontrole do systému MOSS, který běží na serverech standfordské univerzity. Tam se zdrojový kód porovná s ostatními kódy, které byly zaslány v jednom balíčku při spuštění kontroly plagiátů v Moodle. Výsledky jsou ihned po zpracování odeslány zpět a obsahují tabulku, která zobrazuje v procentech podobnosti mezi programy

v rámci jednoho balíčku. Dále obsahuje URL adresy, kde je možné si prohlédnout podobnosti mezi soubory s barevným vyznačením. [8]

1.5.2.3 Výhody a nevýhody

Výhody tohoto modulu jsou velké. Algoritmus MOSS zvládá spolehlivě kontrolovat velké množství souborů v různých programovacích jazycích. Všechno probíhá na externím serveru, takže není potřeba velký výkon serveru. Nevýhoda spočívá v tom, že by fakulta byla odkázána na cizí služby, bez garance dostupnosti, což je nepříjemné.

1.5.3 Modul Online judge

Tento modul slouží k automatickému hodnocení úkolů v kurzech programování, vytvořil jej stejný autor jako modul Anti-plagiarism (viz. bod 1.5.2) a velmi často jsou využívány společně. Modul pracuje se dvěma metodami automatického hodnocení, obě jsou popsány níže.

1.5.3.1 Metoda online zpracování

Tato metoda odesílá programy vytvořené studentem ke zpracování službě Ideone [9], která studentův program zkompiluje, spustí a zpět vrátí výstupy z programu. Modul tyto výstupy porovná s výsledky, které by program vrátit měl a které nadefinoval učitel v konfiguraci úkolu. Pokud výsledky souhlasí, je studentovi přičten určitý počet bodů, který odpovídá tomu, kolik výstupů souhlasí s výstupy definovanými učitelem.

Tato metoda dokáže online kompilovat a spouštět až 40 známých programovacích jazyků (viz. příloha P1) a další stále přibývají. Modul komunikuje se službou Ideone pomocí API, které je popsáno na jejich webu. [10]

1.5.3.2 Metoda lokálního zpracování

Modul Online judge umí kompilovat a spouštět programy i na lokálním serveru. Tato metoda funguje pouze pod operačním systémem Linux a využívá pro spouštění C/C++ zabezpečené prostředí Libsandbox. Toto prostředí brání, aby se program studenta při spouštění dostal k důležitým systémovým souborům, nebo mohl nějakým způsobem napadnout či rebootovat celý server.

1.5.3.3 Instalace

Před samotnou instalací je potřeba ověřit, jestli jsou na serveru nainstalované potřebné komponenty potřebné funkci modulu:

- make
- gcc, g++ (GNU Compiler Collection)
- libc-dev-i386, libstdc++-multilib

Instalace je následující:

- Stačí rozbalit instalační archiv, který je přiložený na CD do složky *moodle_home/mod/assignment/type/*.
- Poté ve složce *mod/assignment/type/onlinejudge/sandbox/* spustit příkaz *\$ make*.
- Poslední řádek výstupu kompilování by měl být **All testcases are OK**.

Pokud chceme používat i online kompilace službou Ideone, je nutné se registrovat [11]. Registrací získáme uživatelské jméno a heslo, které bude náš modul používat pro API komunikaci se serverem Ideone. Toto uživatelské jméno a heslo vložíme do konfiguračního souboru modulu. Konfigurační soubor se jmenuje *config.php* a nachází se v kořenovém adresáři modulu.

```
$CFG->assignment_oj_ideone_username = 'username';  
$CFG->assignment_oj_ideone_password = 'password';
```

Poslední důležitá věc je vytvořit hodnotícího daemona skrze cron:

```
php -q /PATH/TO/MOODLE/admin/cron.php
```

1.5.3.4 Využití v praxi

Kombinace Modulů Anti-plagiarism pro odhalování plagiátů a Online judge pro automatické hodnocení úkolů bylo v praxi otestováno a shledáno vhodným pro použití. Oba moduly bez problémů spolehlivě fungovaly a díky kompilování a spouštění na externích serverech byly nenáročné na provoz, údržbu a výkon serverů UTB. Také by se zvýšila bezpečnost, protože programy studentů by se nikdy nedostaly do kontaktu s programy jiných studentů a nemohly by ani ovlivnit chod serverů UTB se systémem Moodle. Oba moduly jsou navíc v neustálém vývoji a jsou udržovány aktuální. Už nyní jsou k dispozici a i pro Moodle 2.0.

Nakonec ale k využití těchto modulů nedošlo a to z jednoho hlavního důvodu. Fakulta si nemůže dovolit spoléhat na služby třetích stran, které negarantují dostupnost svých služeb.

1.6 Modul ACA

Modul ACA se skládá ze dvou důležitých částí: Automatické hodnocení úkolů a vyhledávání plagiátů. Každou část lze používat odděleně, dříve se jednalo o dva rozdílné moduly.

1.6.1 Automatické hodnocení

Modul ACA umožňuje vytvářet automaticky hodnocené úkoly pro jazyky C/C++, PHP a Pascal. Využívá se k tomu kompilery FPC a interpretu CH a PHP.

Modul porovnává výstupy z učitelského programu s výstupy z programů studentských a podle toho je ohodnotí. Každý definovaný vstup/výstup se nazývá testovací vektor a za každý správný vektor může být přidělen různý počet bodů. Díky tomu dostane každý student jen tolik bodů, kolik si zaslouží. O bezpečnost spouštěných aplikací se stará samotný interpret jazyka Ch, u PHP je to vlastní php.ini soubor používaný pro spouštění PHP skriptů přes CLI, ve kterém jsou zakázány nebezpečné funkce.

1.6.2 Odhalování plagiátů

Plagiát uvažujeme jako textový soubor se zdrojovým kódem, který byl vytvořen jiným uživatelem a byl použitý jen s velmi malým počtem změn. Tento počet změn může učitel definovat, jako povolenou míru shody u každého úkolu. Pro odhalování plagiátů používá modul ACA program YAP3. [18]

1.6.2.1 Program YAP3

Program, který využívá metody přeložení zdrojového kódu programu na kusy textových řetězců, které reprezentují algoritmus testovaného programu. Tyto textové řetězce se potom porovnávají a vede to k odhalení plagiátů, tedy zdrojových kódů, které mají velmi podobnou posloupnost operací. [18]

Proces vyhledávání plagiátů se skládá ze dvou částí:

- Vytvoření tokenů
- Porovnání tokenů

První část se nazývá tokenizace a jedná se o vytvoření textových řetězců reprezentující program. Tokeny se vytváří automaticky při odevzdání programu studentem, je možné je ale vytvořit i hromadně kliknutím na tlačítko „Vytvořit soubory tokenů“ (viz. obr. 8). Skripty pro vytváření tokenů jsou napsány ve skriptovacím jazyce BASH. [18]

Kroky tokenizace:

- Odstranění všech komentářů
- Převedení všech písmen na malá.
- Sjednocení synonym – všechna synonyma jsou nahrazeny jedním výrazem.
- Nahrazení funkcí a klíčových slov tokenem podle slovníku.
- Odstranění všech výrazů, které nejsou ve slovníku.

Druhá část se zabývá porovnáním vytvořených tokenů mezi sebou a vyhledáváním podobností. Tuto práci zajišťuje program *rkr_gst*, který je napsán v jazyce C výsledky zapisuje do textových souborů:

- yap.newfiles
- yap.numbers
- yap.status
- yap.submissions
- yap.summary

II. PRAKTICKÁ ČÁST

2.1 Analýza řešení

Po prozkoumání možností alternativního řešení (viz. bod 1.5) a praktickém vyzkoušených všech výše popsaných modulů jsem se vrátil zpět k modulu ACA vytvořeného před lety studenty na UTB. Je potřeba projít změny mezi verzemi Moodle 1.5 a 1.9 a aplikovat tyto změny na modul ACA. Bohužel dokumentace modulu v bakalářských a diplomových pracích předchozích studentů jsou nedostatečné a je potřeba zdrojové kódy modulu projít a z něj pochopit strukturu modulu a jeho funkce.

2.2 Důležité změny v Moodle 1.9 a v modulu ACA

Poslední úpravy modulu ACA bylo z roku 2006, což je téměř 5 let. To je velmi dlouhá doba na to, že aktualizace Moodle vychází průměrně každý týden a od té doby bylo vydáno hned několik majoritních verzí.

V následujících podkapitolách jsou popsány nejdůležitější změny, které bylo nutné v modulu ACA udělat, aby byl schopný provozu v systému Moodle 1.9.

2.2.1 Deklarace verze modulu

Každý modul musí mít soubor obsahující verzi modulu a verzi Moodle, která je nutná pro provoz modulu. Pokud je verze Moodle nižší, než vyžadovaná verze, k instalaci nedojde.

Soubor `version.php` musí být umístěn v kořenovém adresáři modulu a musí mít následující strukturu:

```
<?PHP // $Id: version.php,v 1.36.2.1 2011/04/01 23:02:44
$module->version = 2011040100;
$module->requires = 2005060230;
$module->cron      = 60;
?>
```

Proměnné `$module->requires` a `$module->version` obsahují čísla verzí, které jsou ve formátu RRRRMMDDXX. Poslední dvojčíslí XX znamená číslo podverze. Proměnná `$module->cron` obsahuje v milisekundách periodu spouštění cronu pro modul.

2.2.2 Instalační soubory

Velkou změnou prošly instalační soubory databáze. Už se nepoužívají klasické sql soubory pro import struktur a výchozích dat do databáze, ale používají se XML soubory.

Struktura těchto XML souborů je následující:

```
<TABLES>
  <TABLE NAME="aca" COMMENT="Defines aca module"
  NEXT="aca_pd_active">
    <FIELDS>
      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
      UNSIGNED="true" SEQUENCE="true" NEXT="course"/>
    </FIELDS>
    <KEYS>
      <KEY NAME="primary" TYPE="primary" FIELDS="id"/>
    </KEYS>
    <INDEXES>
      <INDEX NAME="course" UNIQUE="false" FIELDS="course"/>
    </INDEXES>
  </TABLE>
```

Každá tabulka je definována mezi tagy <TABLES> a </TABLES>, jednotlivé atributy jsou definovány v tabulce.

Tag	Význam
<TABLE> ... </TABLE>	Definice jednotlivé tabulky
NAME	Jméno elementu (tabulky nebo sloupce)
COMMENT	Komentář
NEXT	Jméno dalšího elementu stejné úrovně
PREVIOUS	Jméno předchozího elementu stejné úrovně
<FIELDS> ... </FIELDS>	Definice jednotlivých sloupců
TYPE	Datový typ sloupce (Int, Varchar, text, ...)
LENGTH	Hodnota typu
NOTNULL	Nenulový (true, false)
UNSIGNED	Bez znaménka (true, false)
SEQUENCE	Primární (true, false)
<KEYS> ... </KEYS>	Definice jednotlivých klíčů
<INDEXES> ... </INDEXES>	Definice jednotlivých indexů

Tab. 1. Tabulka tagů v XML DB

2.2.3 Kontextový systém

Zavedení kontextů je jednoznačně největší změnou, která přišla už ve verzi Moodle 1.8. Je to zásadní problém, který způsobil nefunkčnost celého modulu ACA.

V databázi byly odstraněny tabulky *teachers* a *users* a byly nahrazeny kontextovým systémem přidělování práv. Díky těmto kontextům může mít každý uživatel v jiné části Moodle jiné oprávnění, například v prvním kurzu může být zapsán jako učitel a ve druhém jako student. Také je možné vytvářet vlastní role, zatímco ve verzi 1.7 bylo v systému pevně nadefinováno jen šest rolí: Host, Student, Učitel bez práva úpravy, Učitel s právy úpravy, Autor kurzu a Správce. [13]

Kontext určuje oblast nebo úroveň Moodle, ve které se daný uživatel právě nachází. Celý systém je rozdělen do kontextů, které tvoří hierarchii stejně jako složky a soubory v souborovém systému počítače. Například zadání nějakého úkolu je vloženo do určitého kurzu, kurz je zase v nějaké kategorii kurzů, tato kategorie kurzů může patřit po jinou kategorii kurzů a ta se nachází na nějakém konkrétním serveru. Kontext takového úkolu by potom mohl vypadat třeba následovně:

```
Server:vyuka.fai.utb.cz
└─ Kategorie:První ročník
   └─ Kategorie:Informatika
      └─ Kurz:Programování v jazyce Pascal
         └─ Úkol:Výpočet kořenů kvadratické rovnice
```

Obr. 3. Příklad kontextové struktury

2.2.3.1 Přidělování rolí

Nejvyšší kontext se jmenuje systémový kontext a každý uživatel v něm má přiřazenou roli registrovaný uživatel, tedy kromě hosta. Pokud chceme studenta zapsat do nějakého kurzu, musíme mu pro tento kurz přiřadit roli Student. Jestliže má uživatel v některém kontextu přiřazenou roli, tak bude tato role platit a pro všechny podřízené podkontexty. Tedy každý uživatel s rolí student pro nějakou skupinu kurzů má roli student i pro všechny kurzy v této skupině a jejich podskupinách. [13]

2.2.3.2 Nutné změny v modulu

Kvůli zavedení kontextového způsobu přiřazování rolí bylo nutné přepsat SQL dotazy, které v modulu ACA zjišťují, zda jsme učitel a spouští pro porovnání učitelský příklad. Nestačí už jen jednoduše vybrat ID učitele z tabulky *teachers* nebo ID studenta z tabulky *students*, ale je nutné spojit tabulky: *user*, *role*, *role_assignments*, *context*. Každý kontext má ještě svoji úroveň, tzv. kontext level (viz. tab. č. 2).

Úroveň kontextu	Hodnota
CONTEXT_SYSTEM	10
CONTEXT_PERSONAL	20
CONTEXT_USER	30
CONTEXT_COURSE	40
CONTEXT_GROUP	60
CONTEXT_MODULE	70
CONTEXT_BLOCK	80

Tab. 2. Kontextové úrovně a jejich hodnoty

Ukázka kódu pro zjištění uživatelů, kteří mají v kurzu *\$courseid* přidělenou roli *editingteacher*, což je role pro učitele s právy upravovat:

```
$role = $DB->get_record('role', array('shortname' => 'teacher'));
$context = get_context_instance(CONTEXT_COURSE, $courseid);
$teachers = get_role_users($role->id, $context);
```

Byly přepsány SQL dotazy v souboru */aca/lib.php*.

2.2.4 Kódování v UTF-8

Od verze Moodle 1.8 je nutno data do databáze ukládat v kódování UTF-8. Překódovat bylo taky potřeba všechny jazykové soubory v adresáři */lang/cs* včetně všech souborů nápovědy a adresář přejmenovat na */lang/cs_utf8*. Některé chybové zprávy v modulu navíc nebyly obsaženy v českém překladu a vůbec nebyly interpretovány přes překladový engine, proto byly do překladu doplněny.

2.2.5 Globální proměnné

Ve verzi Moodle 1.9 je zakázáno používat globální proměnné. Je potřeba globální proměnné v PHP souborech přepsat a poté v konfiguračním souboru `php.ini` natavit:

```
register_globals = Off
```

Po každé úpravě souboru `php.ini` je potřeba restartovat webový server Apache. Globální je možné vypnout i pomocí PHP flagů v souboru `.htaccess`:

```
php_flag register_globals off
```

2.2.6 Názvy funkcí

Spousta funkcí změnila svůj název nebo počet, pořadí a typy parametrů. Například funkce `print_header()` a `print_header_simple()`, které zabezpečují generování html hlavičky výstupu do prohlížeče a generují hlavní drobečkovou navigaci. Tyto funkce změnilly typy vstupních parametrů a pro to bylo potřeba využít nové funkce `build_navigation()`, která předzpracovává data a sestavuje drobečkovou navigaci.

2.2.7 Zabezpečení souboru `pd4m.make.php`

V posudku oponenta předchozího diplomanta bylo zmíněno, že jedna z hlavních bezpečnostních chyb je soubor `pd4m.make.php`, který není ošetřen proti útokům typu XSS a není plně kompatibilní se systémem Moodle. Soubor není ochráněn proti přístupu zvenčí a tedy každý uživatel může k němu přistupovat, dokonce nemusí být ani přihlášený. Navíc výstup do prohlížeče byl nevalidní, bez html hlavičky a bez udání kódování.

Soubor byl opraven, kód je zpracováván až po ověření přihlášení a není možné se k němu dostat zvenčí bez potřebných učitelských práv. Byly přidány funkce pro generování kódování dokumentu, hlavičky a patičky.

2.2.8 Oprava chyb

Modul nejen, že nebyl kompatibilní se systémem Moodle 1.9 a vyšší, ale také obsahoval velkou spoustu chyb, které znemožňovaly jeho použití v reálném prostředí. Jednalo se spíše o drobné chyby, které předchozí vývojáři přehlédli jako například:

- Zobrazení učitele mezi studenty – učitel byl zobrazen v tabulce uživatelů zapsaných v kurzu mezi obyčejnými studenty a modul se tak k němu taky choval.

- Nevhodně řešené tlačítka – tlačítka pro spuštění hledání plagiátů a test všech odevzdaných programů obnovovala pomocí javascriptové funkce stránku s výsledky. Bohužel pokud bylo před kliknutím na vyhledávání plagiátů kliknuto na tlačítko pro test všech úkolů, tak se stránka obnovila i s předchozími daty odeslanými metodou POST a přepsala nové hodnocení zpět na předchozí.
- Nefunkční hodnocení nula body – kvůli použití špatného operátoru nefungovalo ohodnocení úkolu nula body, pokud bylo zjištěno, že se jedná o plagiát.
- Zobrazení studentovi výsledky kontroly plagiátorství – v případě, že program YAP najde plagiát, provede změnu hodnocení podle nastavených pravidel, ale nevložit do komentáře od učitele text, že se jedná o plagiát. Kód byl přepsán a do komentáře se nyní vkládají jména studentů a procentuální podobnosti s jinými programy.

2.3 Instalace a konfigurace modulu ACA

2.3.1 Požadavky modulu pro provoz

Modul není náročný na fyzické prostředky serveru, jen potřebuje doinstalovat některé doplňky a externí programy jako například interprety a kompilery programovacích jazyků. Podrobný popis instalace těchto programů je uveden v bodu č. 2.3.2 níže.

Prostředek	Minimální verze	Doporučená verze
Operační systém ¹	Linux	Linux
Operační paměť	512 MB	2 GB
Webový server Apache	2.0	2.2.17
Databázový systém	MySQL 4.0.24	MySQL 5.1.56
Skriptovací jazyk PHP	PHP 4.3.10	PHP 5.3.5
CH - interpret jazyka C/C++	4.7.0	6.3.0
FreePascal Compiler (FPC)	2.0.0	2.4.0
LMS Moodle	1.4.2	1.9.11

Tab. 3. Prostředky nutné pro funkci modulu ACA

¹ Teoreticky lze použít jakýkoliv operační systém unixového typu

2.3.2 Návod k instalaci softwaru potřebného pro funkci modulu

Před samotnou instalací je potřeba ověřit, zda jsou dostupné a aktualizované všechny potřebné doplňky a pokud nejsou, tak je doinstalovat, nebo upgradovat na požadovanou verzi.

2.3.2.1 Instalace Apache + MySQL + PHP

Instalaci webového serveru s podporou nejnovější verze PHP a MySQL lze jednoduše zvládnout nainstalováním balíčku Lamp (viz. bod 1.4). Balíček můžete stáhnout ze zdroje [14] a stačí se řídit pokyny pro instalaci v souboru *readme.txt*. Instalační wizard vás provede celým procesem. Adresářová struktura nainstalovaného balíčku je:

- **apache2/**: Apache Webový server.
- **php/**: PHP (skriptovací jazyk PHP).
- **mysql/**: MySQL Databázový server.
- **common/**: různé knihovny a lampctl.sh utility.
- **licenses/**: Licence všeho softwaru použitého v balíčku LAMP.
- **apps/**
 - **phpMyAdmin/**: Webový klient pro správu databází.

Ovládání přes příkazovou řádku balíčku je jednoduché, používá se skript *ctlscript.sh*, který je umístěn v kořenovém adresáři nainstalovaného balíčku.

- `./ctlscript.sh (start|stop|restart)`
- `./ctlscript.sh (start|stop|restart) mysql`
- `./ctlscript.sh (start|stop|restart) apache`

2.3.2.2 Instalace interpretu CH

Ch je interpret pro jazyk C/C++, který používá modul ACA ke spuštění programů napsaných v tomto programovacím jazyku. Při instalaci následujte tyto kroky:

- Stáhněte komprimovaný soubor z webových stránek SoftIntegration [15].
- Až budete vyzváni, zadejte cestu dočasného adresáře (např. `./tmp`) pro soubor **ChEdition-Version.OSversion.platform.tar.gz**. Kde *Platform* a *OSversion* je nahrazena skutečným názvem platformy a verzí OS, pro který je určen. Např. **chstandard-6.3.0.solaris2.6.sparc.tar.gz** je Ch pro verzi 6.3 a pro Sun Sparc

station with Solaris 2.6 nebo vyšší. Doporučená cesta k instalaci Ch je **/usr/local/ch6.3** nebo **HOME/ch**.

- Pro Ch verze 6.3, spusťte následující příkaz v dočasné složce, kde máme archiv uložený a následujte instrukce instalačního wizardu:

```
gzip -cd chstandard-6.3.0.solaris2.6.sparc.tar.gz | tar
xvof -
cd chstandard-6.3.0.solaris2.6.sparc
sh ./install.sh
```

Po instalaci si ověřte, jestli program funguje. Ověřit můžete příkazem: `ch -v`, který vypisuje nainstalovanou verzi a měl by se vrátit výstup:

```
Ch (64-bit)
Evaluation edition, version 6.3.0.14141
Copyright (C) SoftIntegration, Inc. 2001-2010
http://www.softintegration.com
```

2.3.2.3 Instalace kompilery pro jazyk Pascal

Pro funkci kompilování a spouštění programů napsaných v jazyce Pascal je potřeba nainstalovat Free Pascal Compiler, který je zdarma pod licencí GNU General Public License. Kompilér lze stáhnout z webových stránek vývojářů [16].

Pokyny k instalaci:

- Po stáhnutí balíčku kompilery, který může vypadat třeba následovně **fpc-2.4.2.i386-linux.tar**, si jej rozbalíme do dočasné složky (např. `/tmp`) příkazem:

```
tar -zxvf fpc-2.4.2.i386-linux.tar
```

- V rozbaleném archivu spustíme soubor `install.sh` a řídíme se jeho pokyny.

```
sh install.sh
```

- Ověřit jestli je FPC opravdu nainstalovaný můžeme např. příkazem `fpc -help`, který nám zobrazí instalovanou verzi, cíl instalace (např. `/usr/local/lib/fpc/2.4.0/ppcx64`) a nápovědu k obsluze kompilery.

2.3.3 Instalace samotného modulu ACA

Instalace samotného modulu není složitá, jedná se o zkopírování stromové struktury adresářů modulu do modulového adresáře Moodle. Moodle po přihlášení admina modul

automaticky rozpozná a nabídne instalaci, při níž sám vytvoří v databázi tabulky a vloží do nich potřebná data.

2.3.3.1 *Nakopírování modulu*

Adresář aca, který je na přiloženém CD nakopírujte do složky */moodlehomedirectory/mod.Moodlehomedata* je veřejná složka Moodle, ve které se nachází i *index.php*.

Dále je potřeba zkopírovat složku *aca-yap* někam mimo veřejnou složku Apache, například do */home/aca-yap* a přiřadit jim práva 777. V souboru */mod/aca/lib.php* je potřeba nastavit cesty ke skriptům.

2.3.3.2 *Konfigurace cest v souboru lib.php*

Dočasná složka pro rozbalení:

```
$ACACFG->unzipdir = $CFG->dataroot.'/temp/aca/';
```

Jedná se o dočasnou složku, kam se budou rozbalovat komprimované programy studentů před kompilací, spuštěním a hodnocením. Složka by měla mít práva 755 a 644 pro soubory.

Spouštění programů a autokill:

```
$ACACFG->autokill = "/home/aca-yap/myexec";  
$ACACFG->killtimeout = 30;
```

\$ACACFG->autokill je cesta ke skriptu pro spuštění a automatické ukončení spouštěného programu. Pokud běží program déle, než je doba definovaná v proměnné *\$ACACFG->killtimeout*, tak je program ukončen příkazem *kill*.

Cesty k dočasným souborům tokenů:

```
$ACACFG->newpath="/home/aca-yap/yap/tmp_tokens";  
$ACACFG->tmppath="/home/aca-yap/tmp";  
$ACACFG->tokenizer_c="/home/aca-yap/yap/tokenize_C";  
$ACACFG->tokenizer_pascal="/home/aca-yap/yap/tokenizers/tokenize_PASCAL2";  
$ACACFG->tokenizer_php="/home/aca-yap/yap/tokenizers/tokenize_PHP";  
$ACACFG->simchecker="/home/aca-yap/yap/yap3/check2";
```

Tokeny jsou obrazy zdrojových souborů, kdy se klíčová slova a funkce programovacího jazyka nahradily číselnými řetězci, které se mezi sebou porovnávají a na základě nich se potom vyhodnocují plagiáty.

2.3.3.3 Konfigurace cest v BASH skriptech

Bohužel cesty ke skriptům nestačí nakonfigurovat jen v souboru lib.php, ale je potřeba je nastavit i přímo v samotných BASH souborech. Podrobnosti o souborech, ve kterých je potřeba upravit cesty je v tab. č. 4.

Soubor	Řádek	Komentář
./yap/tokenize_C	2	Cesta ke skriptu /yap/tmp
./yap/tokenize_C	3	Cesta ke skriptu /yap/tokenizers
./yap3/check2	5	Cesta ke skriptu /yap/yap3
./yap3/check3	6	Cesta k dočasné složce /yap/tmp
./yap/tokenizers/tokenize_PHP	19	Cesta ke skriptu /yap/tmp
./yap/tokenizers/tokenize_C.sh	21	Cesta ke skriptu /yap/tokenizers
./yap/tokenizers/tokenize_Pascal2	19	Cesta ke skriptu /yap/tmp

Tab. 4. Seznam souborů s pevně danými cestami

2.3.3.4 Nastavení připojení k databázi v BASH skriptech

Poté, co máme nastaveny cesty je nutné vyplnit přihlašovací údaje k databázi do BASH souborů, které spouští studentské programy a výstupy zapisují do databáze. V souboru */yap/yap3/checkall* je potřeba na řádku č. 3 vložit za parametr *-u* uživatelské jméno a za parametr *-p* heslo do databáze. Jméno databáze je hned za heslem odděleno mezerou. Totéž je potřeba provést v souborech */yap/yap3/processResults.awk* a */yap/tokenizers/tokenizeAll* na řádku 3.

Všechny jména tabulek se musí v BASH souborech zadávat i s předponou, která je jinak nastavena v konfiguračním souboru *config.php* jako *\$CFG->prefix*. Proto je potřeba ji přidat do SQL dotazů v souborech:

- */yap/yap3/checkall*, řádek 27.
- */yap/yap3/processResults.awk*, řádek 4 a 24.
- */yap/tokenizers/tokenizeAll*, řádek 18 a 25.

2.3.3.5 Varianta bez konfigurace

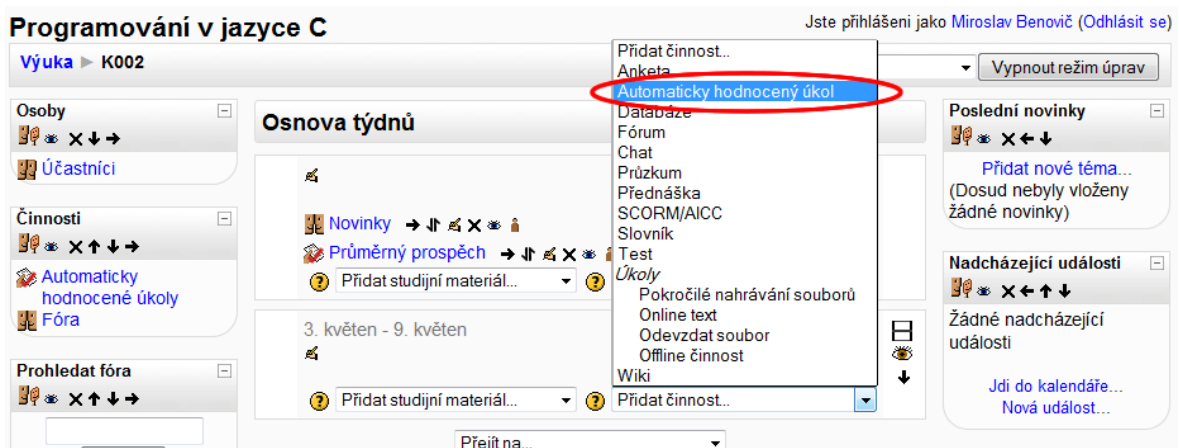
Druhá možnost než složitě přepisovat cesty k dočasným složkám, tokenizačním skriptům a přihlašovací údaje v desítkách zanořených souborů je použít ty údaje a cesty, co jsou ve skriptech už použity. Všechny soubory ze složky *aca-tmp* na přiloženém CD stačí zkopírovat do složky */home/jura* a přihlašovací údaje do databáze jsou uvedeny v tabulce č. 5. níže.

Uživatel	root
Heslo	pCYbV9hvDfcQNbquvgdHc
Databáze	moodle4vyuka

Tab. 5. Přihlašovací údaje do databáze

2.4 Popis modulu z pohledu učitele

Učitel může při zapnutém režimu úprav přidat novou činnost „Automaticky hodnocený úkol“. Tuto možnost musí učitel vybrat, i když nechce úkol hodnotit jen automaticky a chce ho použít pro vyhledávání plagiátů. Automatické hodnocení se dá vypnout v dalším kroku.



Obr. 4. Přidání nového automaticky hodnoceného úkolu

V dalším kroku se nastavuje typ jazyka, ve kterém je program napsán, zobrazení výsledků testovacích vektorů a výsledků kontroly plagiátů. Pokud nastavíme nezobrazovat, tak student výsledky kontroly plagiátorství a výsledky testovacích vektorů po odevzdání svého programu nevidí.

Největší velikost: 2MB

Umožnit znovuodevzdání: Ne

Upozorňovat učitele emailem: Ne

Způsob bodování:
 - ruční
 - váhové funkce
 - přímé body
 - rovnoměrné rozložení

Vložit soubor (Maximální velikost: 1000KB)

Procházet

Učitelství příklad: ?

Způsob bodování: Ručně (bez automatizace)

Při překročení maxima bodů: Přepočítat přes váhu

Volby přímého bodování: Při méně bodech než je maximum: Přepočítat přes váhu

Počet náhodně vybraných vektorů 1

Pokračovat

Možnosti pro přímé bodování

Obr. 5. První krok nastavení

V posledním kroku nahrajte učitelství vzorový příklad, s jehož výstupem se budou srovnávat výstupy studentských programů. Učitelství příklad může být samostatný soubor, nebo zip archiv.

2.4.1 Způsob bodování

Modul nabízí 4 různé způsoby bodování, kterými je program ohodnocen po odevzdání a spuštění. Za každý správně zpracovaný vektor může učitel definovat jiný počet bodů. Proto vznikne několik variant, jak výsledné body zpracovat.

2.4.1.1 Ruční hodnocení

Toto hodnocení je bez automatizace. Program bude sice spuštěn a otestován nadefinovanými vektory, ale body musí přidělit učitel každému studentovi za odevzdaný úkol ručně. Tento způsob hodnocení se hodí pro případy, kdy je program velmi složitý a jeho výstup musí být interpretován člověkem (program například pracuje s náhodnými čísly).

Dále se dá tato možnost použít v situaci, kdy chceme modul ACA využít ve funkci hledání plagiátů, tedy ke kontrole zda studenti od sebe programy nekopírují.

2.4.1.2 *Přímé body*

Výsledné body jsou součtem bodů všech testovacích vektorů, které byly pro studentský program použity. Pokud se na konci celkový součet bodů neshoduje s počtem bodů za úkol nastaveným při vytváření činnosti, tak jsou použita následující pravidla:

- Při překročení maxima – celkový součet bodů je větší jak body za úkol.
- Při nedosažení minima – celkový součet bodů je menší jak body za úkol.
- Počet náhodných vektorů – volba „*Náhodně vybrat*“. V tomto případě je počet bodů za každý vektor stejný.

2.4.1.3 *Váhové funkce*

Hodnocení se přepočítává přes váhy jejich bodů tak, aby výsledný součet bodů dával přesné maximum bodů pro úkol. Toto je asi nejvhodnější způsob bodování, používá se, pokud některé testovací vektory jsou důležitější, než jiné.

2.4.1.4 *Rovnoměrné rozložení*

Jednotlivé počty bodů za vektor jsou ignorovány a za každý vektor je přiděleno bodů stejně (rozpočítávají se aritmetickým průměrem). To znamená, že pokud máme celkem za úkol maximální počet 100 bodů a máme 10 testovacích vektorů, tak je za každý správný vektor přiděleno studentovi 10 bodů.

2.4.2 **Testovací vektory**

Testovací vektory jsou vstupní data, která jsou předány programu ke zpracování. Tyto testovací vektory jsou předány učitelskému i studentskému programu a porovnává se jejich výstup. Pokud jsou výstupy shodné, jsou studentovi přičteny body za vektor. Vektorů může být pro každý úkol nedefinováno neomezené množství.

Rozbalený učitelský program

Tlačítko ke spuštění a otestování učitelského programu

Přidat nový testovací vektor

Metoda testování	Nastavení	Počet nastavených vektorů	Součet bodů za vektory	Přidat nový vektor
Výchozí test C	Nastavení	3	100	<input type="button" value="Přidat"/>
Celkové součty		3	100	

Hledání plagiátů

Povolená shoda

Akce při nalezení plagiátu

Obr. 6. Nastavení testovacích vektorů a vyhledávání plagiátů

Umístění vstupních parametrů je možné přímo v definici vektoru, nebo je načítat ze souboru, který musí být přibalen v učitelském programu. Pokud chceme vkládat vstupní parametry přímo do vektoru, musíme vybrat volbu „umístění vstupních parametrů – seznam“ a parametry vložíme přímo do pole vstupní parametry. V opačném případě, pokud chceme vstupní data brát ze souboru, musíme vybrat volbu „umístění vstupních parametrů – soubor“ a do pole pro vstupní data zapíšeme jméno souboru (viz. obr. 7).

Jazyk C - Výchozí test C

Metoda prověřuje funkčnost programů v jazyce C. Do spouštěcího C souboru jsou importovány všechny C soubory. Vstupem může být seznam parametrů nebo lokální soubor jako parametr, z něž program data čte.

id	citlivost porovnání výstupu	umístění vstupních parametrů	vstupní parametry	Znamka	Testováno	Operace
2	ignorovat velikost a prázdné místo	soubor	indata.txt	30	Ano (Úterý, 3. květen 2011, 06:36)	<input type="button" value="Vytvořit kopii"/> <input type="button" value="Smazat vektor"/>
3	ignorovat velikost a prázdné místo	soubor	indata2.txt	30	Ano (Úterý, 3. květen 2011, 06:36)	<input type="button" value="Vytvořit kopii"/> <input type="button" value="Smazat vektor"/>
4	ignorovat velikost a prázdné místo	soubor	indata3.txt	40	Ano (Úterý, 3. květen 2011, 06:36)	<input type="button" value="Vytvořit kopii"/> <input type="button" value="Smazat vektor"/>

Soubory se vstupními daty

Body za vektory

Obr. 7. Nastavení testovacích vektorů

2.4.3 Spuštění testů

Testy studentských programů se provádí automaticky ihned po uploadu programu studentem do Moodle. Zároveň se vytvoří soubory tokenů pro pozdější kontrolu plagiátorství. Spustit test může ale i učitel u všech úkolů najednou tlačítkem „Test vektorů všech úkolů“, nebo u každého individuálně tlačítky „Spustit test vektorů“ u každého úkolu.

Programování v jazyce C

Výuka > K002 > Automaticky hodnocené úkoly > Průměrný prospěch > Odevzdané úkoly

Upravit tuto činnost - Automaticky hodnocený úkol

Spuštění všech testů

Test vektorů všech úkolů

Vytvořit soubory tokenů pro všechny odevzdané úkoly

Spustit hledání plagiátů

Spuštění hledání plagiátů

Křestní jméno : Vše AÁBCČĎĚÉĚFGHIJKLNMNOÓPQRŘSŠTŤUÚÚVWXYÝŽŽ
 Příjmení : Vše AÁBCČĎĚÉĚFGHIJKLNMNOÓPQRŘSŠTŤUÚÚVWXYÝŽŽ

Křestní jméno / Příjmení	Známka	Komentář	Naposledy změněno (Student)	Naposledy změněno (Učitel)	Stav
Student 1	100 / 100		Výsledky testů PrumernyProspech.c pp.h studentok.zip Pondělí, 2. květen 2011, 20.56	Není podobný Spustit test vektorů Pátek, 20. květen 2011, 21.59	Aktualizovat
Student 2	0 / 100		Výsledky testů PrumernyProspech.c pp.h studentfail.zip Úterý, 3. květen 2011, 06.26	Není podobný Spustit test vektorů Pátek, 20. květen 2011, 21.59	Aktualizovat
Student 3	100 / 100		Výsledky testů PrumernyProspech.c pp.h studentok.zip Úterý, 3. květen 2011, 06.35	Není podobný Spustit test vektorů Pátek, 20. květen 2011, 21.59	Aktualizovat

Bodové zisky za úkol

Individuální testy

Obr. 8. Přehled všech odevzdaných programů v kurzu

2.4.4 Výsledky testů

Výsledky testů si je možné prohlédnout po spuštění a zpracování programu. Slouží pro to odkaz „Výsledky testů“ u jména studenta (náhled na obr. č. 9). Pokud je výstup programu moc dlouhý (například vygenerovaná celá html stránka), tak je v tabulce výstupů výstup nahrazen dalším odkazem „Výsledky testů“.

Výsledky testů

id testu	vstupy	výstupy	správné výstupy	bodů za vektor	přepočtené body	získané body
2	Veronika Vaskova 4.0 Pavel Prochazka 4.0 Michal Sedlar 2.0 Martin Mikulka 2.0	Prumerny prospěch je 3.00 bodu.	Prumerny prospěch je 3.00 bodu.	30	30	30 Výsledky testu
3	Veronika Vaskova 4.0 Pavel Prochazka 7.0 Michal Sedlar 1.0 Martin Mikulka 2.0	Prumerny prospěch je 3.50 bodu.	Prumerny prospěch je 3.50 bodu.	30	30	30 Výsledky testu
4	Veronika Vaskova 10.0 Pavel Prochazka 5.0 Michal Sedlar 3.0 Martin Mikulka 2.0	Prumerny prospěch je 5.00 bodu.	Prumerny prospěch je 5.00 bodu.	40	40	40 Výsledky testu

Získané body: 100/100

Vstupy do programu

Výsledek hodnocení

Výstupy z programu

Výstupy z učitelského programu

Obr. 9. Výsledky testovacích vektorů

2.4.5 Kontrola plagiátorství

Po vytvoření automaticky hodnoceného úkolu a nastavení způsobu bodování je možné nastavit detekci plagiátorství programem YAP3. Kontrolu všech programů, které studenti

odevzdali může učitel spustit tlačítkem „Spustit kontrolu plagiátů“. Při nalezení plagiátu může modul automaticky strhnout určité procento bodů, nebo úkol obodovat pevným počtem bodů.

Výsledky hledání plagiátů, tedy procentuální shody s programy jiných studentů si můžete zobrazit kliknutím na odkaz „Zobrazit podobnosti“. Pokud nejsou nalezeny shodné programy, bude tam místo odkazu jen text „Není podobný“. Výsledky jsou také vloženy do komentáře k programu, aby student věděl, proč mu byly body strženy a kteří studenti odevzdali podobné programy.

Kontrolu plagiátorství je vhodné používat pro větší a složitější programy, které mají více než 40 řádků kódu. Kratší programy nemá smysl kontrolovat, protože ty budou alespoň částečně podobné vždy, i když se nebude jednat o plagiát.

Autor úkolu	Podobnost
Student 1	100 %
Student 2	77.27 %
Student 3	80 %
Student 4	100 %

Obr. 10. Výsledky kontroly plagiátů

2.5 Popis modulu z pohledu studenta

Z pohledu studenta je modul velice jednoduchý. Může pouze nahrát soubor svého programu, nebo komprimovaný archiv a prohlédnout si výsledky testů a počet bodů, které mu byly přiděleny. Zobrazení výsledku testů může ale učitel v konfiguraci úkolu vypnout.

Pokud učitel povolí možnost opětovného odevzdání úkolu, tak může student program opravit a odevzdat znovu. To je praktické v případech, kdy student není spokojen s počtem bodů které obdržel, nebo byl jeho program vyhodnocen jako plagiát.

Programování v Pascalu ◀ Přejít na... ▾

Výuka ▶ K003 ▶ Automaticky hodnocené úkoly ▶ Vykreslení funkce

Pondělí, 2. květen 2011, 20.10

Program pro výpočet a vykreslení bodů funkce.

Dostupné od: Neděle, 1. květen 2011, 20.00
Termín odevzdání: Sobota, 9. červenec 2011, 20.00

id testu	vstupy	výstupy	správné výstupy	bodů za vektor	přepočtené body	získané body
1	0	Výsledky testu	Výsledky testu	100	100	100 Výsledky testu

Získané body: 100/100

Odpověď od: Učitel



Učitel 1
Pátek, 13. květen 2011, 11.53

[? student](#)

[? student.o](#)

[? student.pas](#)

Známka: 100 / 100

Obr. 11. Rozhraní s pohledu studenta

ZÁVĚR

Cílem této práce bylo upravit stávající modul ACA, který byl v předchozích letech vytvořen studenty FAI. Tento modul byl upraven, aby fungoval i v prostředí Moodle 1.9 a je velmi pravděpodobné, že bude fungovat i v Moodle 2.0.

V první fázi práce jsem se zaměřil na použití některých již existujících modulů, které vytvořila komunita LMS Moodle. Prošel jsem všechny varianty hotových modulů, postupně je nainstaloval a odzkoušel. Jejich popisy, náhledy, výhody a nevýhody jsem shrnul a sepsal do teoretické části práce.

Protože žádný z alternativních modulů plně neodpovídal představě použití na FAI, zaměřil jsem se ve druhé fázi na přepsání již hotového modulu ACA. S pomocí komunity se mi modul podařilo přepsat, je tedy funkční a vhodný k nasazení do praxe. V modulu byly také opraveny bezpečnostní chyby, které byly objeveny, nebo zmiňovány oponentem v posudku předchozí práce.

Věřím, že nově opravený modul pomůže zvýšit úroveň v kurzech programování na FAI UTB ve Zlíně, nebo i jiných vysokých škol, které se rozhodnou pro jeho využití.

ZÁVĚR V ANGLIČTINĚ

The goal of this work was to adjust current ACA module, created few years ago by students of FAI. This module was adjusted to work properly in Moodle 1.9 and it probably will work even in Moodle 2.0.

In the first phase of this work I've focused on using some already existing modules created by LMS Moodle community. I went through all variants of finished modules, installed and tried out one after another. Descriptions, previews, advantages and disadvantages are scheduled in theoretical part of the work.

Because none of the alternative modules did match to usage vision on FAI, in the second phase I've focused on rewriting of the finished ACA module. With some help from community I succeeded in module rewriting, so it's functional and ready to be set up into practice. There were also fixed some security bugs, whether discovered by me or mentioned by the opponent in the review of the previous work.

I believe that newly repaired module will help to raise the standard in programming courses on FAI UTB in Zlín or on other universities, which will decide to use it.

SEZNAM POUŽITÉ LITERATURY

- [1] *Moodle* [online]. [cit. 2011-04-26]. Dostupná z WWW: <<http://moodle.org/about/>>.
- [2] *Moodle* [online]. [cit. 2011-04-24]. Dostupná z WWW: <http://docs.moodle.org/cs/Co_je_Moodle>.
- [3] *Root.cz* [online]. [cit. 2011-04-24]. Dostupná z WWW: <<http://www.root.cz/specially/licence/informace-k-licencnim-podminkam/>>.
- [4] *Root.cz* [online]. [cit. 2011-04-26]. Dostupná z WWW: <<http://mystik.blog.root.cz/2009/01/04/je-php-jazyk-pro-amatery/>>.
- [5] BELL, Charles A. *Expert MySQL*. USA : Apress, 2007. 577 s. ISBN 1-59059-741-5.
- [6] *Abclinuxu.cz* [online]. [cit. 2011-04-27]. Dostupná z WWW: <<http://www.abclinuxu.cz/software/server/database/>>.
- [7] *MySQL manuál* [online]. [cit. 2011-04-27]. Dostupná z WWW: <<http://www.junext.net/mysql/>>
- [8] *Systém Moss* [online]. [cit. 20011-05-06]. Dostupný z WWW: <<http://theory.stanford.edu/~aiken/moss/>>
- [9] *Ideone.com* [online]. 2011 [cit. 2011-05-07]. Ideone.com. Dostupné z WWW: <<http://ideone.com/>>.
- [10] *Ideone API* [online]. 2011 [cit. 2011-05-07]. Dostupné z WWW: <<http://ideone.com/files/ideone-api.pdf/>>.
- [11] *Ideone.com* [online]. 2011 [cit. 2011-05-07]. Ideone.com. Dostupné z WWW: <<http://ideone.com/>>.
- [12] *Online Judge Plugin for Moodle* [online]. 2011 [cit. 2011-05-07]. Dostupné z WWW: <<http://code.google.com/p/sunner-projects/wiki/OnlineJudgeAssignmentType/>>.
- [13] *MoodleDocs* [online]. 2011 [cit. 2011-05-011]. Role. Dostupné z WWW: <<http://docs.moodle.org/cs/Role/>>.
- [14] *Bitnami* [online]. 2011 [cit. 2011-05-18]. Stacks. Dostupné z WWW: <<http://bitnami.org/stacks/>>.
- [15] *Ch* [online]. 2011 [cit. 2011-05-18]. An embeddable C/C++ interpreter and C/C++ scripting languag. Dostupné z WWW: <<http://www.softintegration.com/>>.

- [16] *Free Pascal* [online]. 2010 [cit. 2011-05-19]. Compiler. Dostupné z WWW: <<http://www.freepascal.org/>>.
- [17] VAŠKOVÁ, Veronika. *Automatické hodnocení úkolu v kurzech programování*. Zlín, 2005. 25 s. Bakalářská práce. UTB ve Zlíně.
- [18] JURA, Pavel. *Moduly eLearning systému Moodle pro potřeby výuky na UTB ve Zlíně*. Zlín, 2006. 62 s. Diplomová práce. UTB ve Zlíně.
- [19] VRÁNA, Jakub. *1001 tipů a triků pro PHP 1001 tipů a triků pro PHP*. Praha : Computer Press, 2011. 456 s. ISBN 9788025129401.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ACA	Automatically Checked Assignment – automaticky hodnocený úkol. Název modulu pro systém Moodle vytvořen v letech 2005 na FAI UTB ve Zlíně.
Awk	Utilita pro zpracování textu, jazyk textových manipulací.
BASH	Bourne Again Shell – unixový shell, který vytváří interpret pro příkazový řádek.
LMS	Learning Management System (systém pro řízení výuky) – online aplikace řešící administrativu a organizaci výuky.
MOSS	Measure Of Software Similarity – online služba pro odhalování plagiátu.
MySQL	Databázový systém založený na jazyku SQL.
SQL	Structured Query Language – strukturovaný dotazovací jazyk.
PHP	PHP: Hypertext Preprocessor – rekurzivní název skriptovacího jazyka (programátorský vtíp).
Sed	Utilita pro Linux k dávkové práci s textem.
YAP	Yet Another Program Plagiarism Detector – program pro odhalování plagiátu.
CSV	Comma-separated values – Souborový formát, data oddělená čárkami.
InnoDB	Databázové úložiště, podporuje cizí klíče, transakce, body obnovení.
MyISAM	Jedno z nejpoužívanějších databázových úložišť, vychází z úložiště ISAM.
ISAM	Indexed Sequential Access Method – databázové úložiště.
Bing	Internetový vyhledávač provozovaný společností Microsoft.
LAMP	Zkratka pro balíček svobodného softwaru Linux, Apache, MxSQL, PHP.
FPC	Free Pascal Compiler – nástroj pro kompilování jazyka Pascal.
ZIP	Metoda komprimování souborů.
FAI	Fakulta aplikované informatiky
UTB	Univerzita Tomáše Bati

App ID	Unikátní identifikátor použitý k propojení různých aplikací.
Ch	Interpret jazyka C/C++.
CLI	Command-line interface – rozhraní, kdy uživatel komunikuje s programem pomocí příkazové řádky.
RKR-GST	Running Karp-Rabin and Greedy String Tiling – program obsahující algoritmus pro porovnávání textových řetězců (tokenů).

SEZNAM OBRÁZKŮ

Obr. 1. Náhled výsledků detekce plagiátorství modulu Crot.....	15
Obr. 2. Náhled rozhraní k detekci plagiátorství modulu Anti-plagiarism	16
Obr. 3. Příklad kontextové struktury	24
Obr. 4. Přidání nového automaticky hodnoceného úkolu.....	32
Obr. 5. První krok nastavení	33
Obr. 6. Nastavení testovacích vektorů a vyhledávání plagiátů.....	35
Obr. 7. Nastavení testovacích vektorů	35
Obr. 8. Přehled všech odevzdaných programů v kurzu	36
Obr. 9. Výsledky testovacích vektorů.....	36
Obr. 10. Výsledky kontroly plagiátů	37
Obr. 11. Rozhraní s pohledu studenta.....	38

SEZNAM TABULEK

Tab. 1. Tabulka tagů v XML DB.....	23
Tab. 2. Kontextové úrovně a jejich hodnoty.....	25
Tab. 3. Prostředky nutné pro funkci modulu ACA.....	27
Tab. 4. Seznam souborů s pevně danými cestami	31
Tab. 5. Přihlašovací údaje do databáze.....	32

SEZNAM PŘÍLOH

P I Jazyky podporované službou Ideone

PŘÍLOHA P I: JAZYKY PODPOROVANÉ SLUŽBOU IDEONE

Programovací jazyky podporované službou Ideone (1. část).

Jméno jazyka	Verze	Zpracovaných kódů	API ID
Ada	gnat-4.3.2	10953	7
Assembler	nasm-2.07	10402	13
AWK (mawk)	mawk-1.3.3	1281	105
Bash	bash 4.0.35	28612	28
bc	bc-1.06.95	1695	110
Brainf**k	bff-1.0.3.1	27772	12
C	gcc-4.3.4	351568	11
C#	mono-2.8	68620	27
C++	gcc-4.3.4	654016	1
C++0x	gcc-4.5.1	15778	44
C99 strict	gcc-4.3.4	19993	34
CLIPS	clips 6.24	2689	14
Clojure	clojure 1.1.0	5390	111
COBOL	open-cobol-1.0	4926	118
COBOL 85	tinycobol-0.65.9	5391	106
Common Lisp (clisp)	clisp 2.47	11814	32
D (dmd)	dmd-2.042	9962	102
Erlang	erl-5.7.3	4803	36
F#	fsharp-2.0.0	3440	124
Factor	factor-0.93	688	123
Falcon	falcon-0.9.6.6	634	125
Forth	gforth-0.7.0	3102	107
Fortran	gfortran-4.3.4	16709	5
Go	gc-2010-07-14	5992	114
Groovy	groovy-1.7	1616	121
Haskell	ghc-6.8.2	21724	21
Icon	iconc 9.4.3	1945	16
Intercal	c-intercal 28.0-r1	2348	9
Java	sun-jdk-1.6.0.17	373841	10

Programovací jazyky podporované službou Ideone (2. část)

Jméno jazyka	Verze	Zpracovaných kódů	API ID
JavaScript (rhino)	rhino-1.6.5	12498	35
JavaScript (spidermonkey)	spidermonkey-1.7	13619	112
Lua	luac 5.1.4	7786	26
Nemerle	ncc 0.9.3	1957	30
Nice	nicec 0.9.6	1757	25
Nimrod	nimrod-0.8.8	343	122
Objective-C	gcc-4.5.1	2093	43
Ocaml	ocamlopt 3.10.2	3825	8
Oz	mozart-1.4.0	1297	119
Pascal (fpc)	fpc 2.2.0	75548	22
Pascal (gpc)	gpc 20070904	11441	2
Perl	perl 5.12.1	54358	3
Perl 6	rakudo-2010.08	2550	54
PHP	php 5.2.11	113055	29
Pike	pike 7.6.86	2022	19
Prolog (gnu)	gprolog-1.3.1	3353	108
Prolog (swi)	swipl 5.6.64	41883	15
Python	python 2.6.4	96773	4
Python 3	python-3.1.2	22042	116
R	R-2.11.1	4333	117
Ruby	ruby-1.9.2	28688	17
Scala	scala-2.8.0.final	6698	39
Scheme (guile)	guile 1.8.5	8024	33
Smalltalk	gst 3.1	3243	23
SQL	sqlite3-3.7.3	6934	40
Tcl	tclsh 8.5.7	4968	38
Text	text 6.10	6061	62
Unlambda	unlambda-2.0.0	948	115
Visual Basic .NET	mono-2.4.2.3	15612	101
Whitespace	wspace 0.3	6365	6

