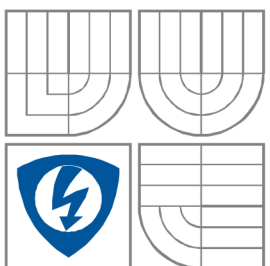


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

AUTOMATICKÉ GENEROVÁNÍ TESTOVÝCH OTÁZEK

AUTOMATIC GENERATION OF TEST QUESTION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

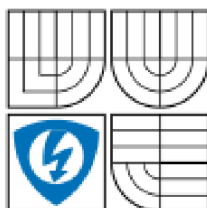
AUTOR PRÁCE
AUTHOR

Bc. Pavel Cídl

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Tomáš Frýza, Ph.D.

BRNO, 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Pavel Cídl

ID: 89675

Ročník: 2

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Automatické generování testových otázek

POKYNY PRO VYPRACOVÁNÍ:

V systému MySQL vytvořte strukturu databáze umožňující generování testových otázek. Pomocí jazyků HTML a PHP vytvořte jednoduchou aplikaci umožňující přístup k datům z databáze: prohlížení, editace, přidávání záznamů, apod.

Databázový systém rozšiřte o možnosti náhodného výběru testových otázek z několika okruhů, příp. podle typu generovaného testu, včetně záměny pořadí otázek: písemka, zkouška, test, apod.

Stávající systém doplňte funkcí generace zadání i správných odpovědí ve formátu PDF a HTML pomocí systému LaTeX.

DOPORUČENÁ LITERATURA:

[1] DALY, P.W., KOPKA, H. LaTeX – kompletní průvodce. Praha: Computer Press, 2005. 576 stran. ISBN 80-7226-973-9

[2] W3C. Converting from HTML. [online]. 1999 – [cit. 21. prosince 2007]. Dostupné na WWW: <http://www.w3.org/Tools/html2things.html>

Termín zadání: 9.2.2009

Termín odevzdání: 29.5.2009

Vedoucí práce: Ing. Tomáš Frýza, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Pavel Cídl
Bytem: Nedvědice 121, 59262
Narozen/a (datum a místo): 16.1.1984 v Novém Městě na Moravě

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Automatické generování testových otázek

Vedoucí/ školitel VŠKP: Ing. Tomáš Frýza, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy
(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 29. května 2009

.....
Nabyvatel

.....
Autor

Anotace

Tento projekt se zabývá automatickým generováním testových otázek. Najdeme tu komplexní řešení problému náhodné generace otázek. Pomocí jazyků HTML a PHP je vytvořena jednoduchá aplikace umožňující přístup k datům z databáze: prohlížení, editace, přidávání záznamů, apod. Systém dále umožňuje náhodný výběr testových otázek z několika okruhů a podle typu generovaného testu. Další funkcí je generace otázek i správných odpovědí ve formátu PDF a HTML pomocí systému LaTeX.

Annotation

This project deal with automatic generation testing questions. We will find that turn-key solution problem randomly generation questions. By the help of languages HTML and PHP is created simple application susceptible data access from database: adding, editing, addition entries, etc . System further makes it possible to potluck testing questions of several perimeters and according to type generated test. Next function is generation question and correct answers in size PDF and HTML by the help of system LaTeX.

Klíčová slova

Webový server, databázový systém, hypertextový preprocesor.

Key words

Web server, database system, hypertext pre-processor.

CÍDL, P. *Automatické generování testových otázek*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 46s.

Prohlášení

Prohlašuji, že svůj semestrální projekt na téma Automatické generování testových otázek jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 29.května 2009

.....
podpis autora

Poděkování

Děkuji vedoucímu semestrálního projektu Ing. Tomáši Frýzovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V Brně dne 29.května 2009

.....
podpis autora

Obsah

1. Úvod.....	10
2. Úvod do použitých technologií a do hnutí Open Source	11
2.1 Apache	11
2.2 PHP	12
2.3 MySQL	13
2.4 HTML.....	14
2.5 CSS (Cascading Style Sheets).....	17
3. Tvorba databáze a její obsluha.....	19
3.1 Návrh a vytvoření databáze	19
3.2 Vkládání, získávání a zobrazování dat z databáze	20
3.3 Úprava dat v databázi	23
3.3.1 Vkládání nových otázek do databáze	24
3.3.2 Editace otázek v databázi.....	24
3.3.3 Mazání záznamů v databáze.....	25
3.4 PSPad editor	26
3.5 phpMyAdmin	27
4. Generování otázek.....	30
4.1 Funkce pro náhodné generování	30
4.2 Generování zadání pro písemku.....	31
4.3 Generování zadání pro test.....	33
4.4 Generování zadání pro zkoušku	35
4.5 Export zadání pomocí systému LaTeX.....	37
5. Závěr.....	41
6. Seznam použité literatury	42
7. Přílohy	43
Příloha A – Ukázka skriptů pro práci s databází:	43
Příloha B – Ukázka skriptů pro generování otázek:	44
Příloha C – Vygenerovaný soubor *.tex pro systém LaTeX:	46

Seznam obrázků

1. Rozdíl mezi statickou a dynamickou stránkou.....	15
2. Primární tabulka databáze.....	19
3. Další tabulky databáze.....	20
4. Seznam testových otázek.....	21
5. Seznam otázek zadání.....	22
6. Použité okruhy testových otázek.....	22
7. Formulář pro přidání nové testové otázky.....	24
8. Formulář pro úpravu testové otázky.....	25
9. Okno textového editoru PSPad.....	27
10. Hlavní okno webové aplikace phpMyAdmin.....	27
11. Vytvořená databáze „test“ a její tabulky.....	28
12. Struktura sloupců v tabulce „otazky“.....	28
13. Okno pro přidání nové testové otázky.....	29
14. Menu aplikace pro generování otázek – záložka Hlavní menu.....	31
15. Nastavení generování písemky.....	32
16. Generování zadání pro písemku.....	32
17. Nastavení generace testu.....	33
18. Generování zadání pro test.....	34
19. Tabulka správných odpovědí.....	35
20. Nastavení generování zkoušky.....	35
21. Generování zadání pro zkoušku.....	36
22. Tabulka správných odpovědí pro zkoušku.....	37
23. Vygenerované zadání v PDF souboru pomocí systému LaTeX.....	39
24. Vygenerované zadání v HTML pomocí systému LaTeX.....	40

1. Úvod

V diplomové práci se seznámíme s moduly AMP (Apache, MySQL a PHP). Rozebereme jejich jednotlivé součásti a dozvíme se tak, jakým způsobem spolu komunikují. Úkolem práce je v systému MySQL vytvořit strukturu databáze umožňující generování testových otázek a pomocí jazyků HTML a PHP vytvořit jednoduchou aplikaci umožňující přístup k datům z databáze: prohlížení, editace, přidávání záznamů, apod. Dále databázový systém rozšířit o možnost náhodného výběru testových otázek z několika okruhů, příp. podle typu generovaného testu, včetně záměny pořadí otázek: písemka, zkouška, test, apod. Posledním krokem bude vytvořený systém doplnit funkcí generace zadání i správných odpovědí ve formátu PDF a HTML pomocí systému LaTeX.

Druhá kapitola se věnuje použitým technologiím, kterých je třeba využít k úspěšnému vypracování projektu. K vytvoření uživatelského prostředí nám poslouží HTML a CSS kaskádní styly. Pro samotný program je pak dále nutné se seznámit s PHP a MySQL jazykem. Aby to celé dohromady mělo kde pracovat, potřebujeme také znát použití virtuálního serveru Apache.

Třetí kapitola se zabývá návrhem a vytvářením databáze, do které nahráváme potřebné informace o generovaných zadáních. Ukládáme zde tedy otázky sloužící ke generování písemek, testů nebo zkoušek. Otázky třídíme podle kategorie a okruhu otázky. Pomocí jazyků HTML a PHP vytvoříme aplikaci umožňující přístup k datům z databáze. Toho dosáhneme dvojnásobným způsobem a to pomocí formulářů nebo pomocí webové aplikace phpMyAdmin.

Čtvrtá kapitola pojednává o nejdůležitější části celého projektu a tou je bezesporu vyřešení náhodné generace zadání. Zde je nutné brát v úvahu, aby každé generované zadání i s možnostmi odpovědí bylo opravdu náhodného charakteru a odpovídalo také rozložení otázek v databázi podle použitých okruhů. Další podmínkou je současně s vygenerovaným zadáním zobrazovat také správné výsledky na otázky v případě generování testu. Vytvořené zadání nemusíme zobrazovat pouze na obrazovce monitoru, ale můžeme ho i uložit do souboru, který nám poslouží později. Zde využijeme systému LaTeX, který ze souboru, který je uložen v odpovídajícím formátu, může generovat zadání ve formátu PDF a HTML.

2. Úvod do použitých technologií a do hnutí Open Source

Termín *open source* (otevřený zdroj) vznikl v roce 1998 poté, co se firma Netscape rozhodla zveřejnit zdrojový kód svého oblíbeného prohlížeče Navigator. Toto oznámení podnítilo malou skupinu softwarových vývojářů, kteří byli dlouhodobými členy skupiny, jež se měla brzy přerodit v hnutí open source, k tomu, aby formálně oznámili vznik iniciativ OSI (Open Source Initiatives) a definice OSD (Open Source Definition). Přestože byla ideologie OSI původně podporována pouze komunitou hackerů, začali po uvolnění zdrojového kódu prohlížeče Netscape Navigator přispívat svými nápady a opravami programátoři všech vrstev a postavení. Poselství OSI bylo vysláno a začalo žít vlastním životem. Hlavní proud počítačového světa s ním musel začít počítat.

Programy typu open source jsou velmi dobré, protože:

- Jsou zadarmo, tedy volně dostupné široké veřejnosti. Softwaroví vývojáři a programátoři věnují dobrovolně svůj čas zdokonalování existujícího softwaru a tvorbě nových programů. Otevřený software nevyžaduje už samotné své postavy žádné formy licencování nebo prodejních poplatků.
- Jsou nezávislé na platformě a jsou nezávislé na technologii.

2.1 Apache

Apache působí jako webový server. Jeho hlavní úlohou je nejen zpracování požadavků, které uživatelé odesílají prostřednictvím svých webových prohlížečů, ale také zobrazení výsledků připravených pomocí kódu umístěného ve vyžádaných souborech. Apache je velmi výkonný stroj a může splnit prakticky všechna přání, jaká můžeme jako správce webu potřebovat.

Mezi základní funkce webového serveru patří:

- Stránky chráněné pomocí hesel různých uživatelů,
- Přizpůsobené stránky oznamující chyby,
- Zobrazení kódu v různých úrovních HTML, jakož i schopnost určit, na které úrovni může prohlížeč přijmout nabízený obsah,
- Protokoly užití a chyb v mnoha různých konfigurovatelných formátech,
- Nastavení různých adres IP mapovaných na stejný webový server,
- Direktivy *Directory Index* pro více souborů,
- Přezdívání adres URL nebo přepisování bez pevného limitu.

Webový server apache je velmi oblíbený a jeho popularitu ovlivňuje nepochybně flexibilita, výkon a samozřejmě také cena. Lze ho používat jako hostitele veřejného webového serveru, jako hostitele podnikového intranetu nebo jednoduše jako testovací stroj pro vyvíjené stránky, které není třeba za těmito účely odesílat na zabezpečený server nebo jiný počítač.

2.2 PHP

PHP je velmi komplexní programovací jazyk pro dynamické webové aplikace. Je to nejrozšířenější serverový jazyk pro tvorbu webu. Jazyk umožňuje skriptování na straně serveru. Umožňuje dynamizaci webového serveru. Jedná se o hypertextový procesor, který na serveru interpretuje stránky HTML s vlastními příkazy před jejich odesláním ke klientovi (obvykle webový prohlížeč) a to znamená, že PHP umožňuje vkládat vlastní skripty přímo do hypertextových stránek podobně jako třeba Java Skript. Rozdíl je však v tom, že Java Skript je interpretován přímo u klienta, kdežto PHP na straně serveru. Jeho flexibilita z něj činí jeden z nejoblíbenějších skriptovacích jazyků vůbec.

K tomu vede celá řada věcí:

- je relativně jednoduché na pochopení
- má syntaxi velmi podobnou jazyku C a je tedy většině vývojářů dost blízký
- podporuje širokou řadu souvisejících technologií, formátů a standardů
- je to otevřený projekt s rozsáhlou podporou komunity
- dají se najít kvanta již hotového kódu k okamžitému použití nebo funkční PHP aplikace. Podstatná část z hotového kódu je šířena pod nějakou svobodnou licenci a dá se použít ve vlastních projektech
- dobře si rozumí s webovým serverem Apache
- snadno komunikuje s databázemi, jako je MySQL, PostgreSQL a řada dalších
- je to multiplatformní a lze jej provozovat s většinou webových serverů a na většině dnes existujících operačních systémů
- podporuje mnoho existujících poskytovatelů webhostingových služeb

Existuje však i několik nevýhod:

- je interpretovaný, ne kompilovaný jazyk
- kdokoli má přímý přístup k serveru, může nahlédnout do vašich PHP skriptů
- podpora objektového programování není v PHP na moc dobré úrovni.
- protože je PHP aktivně vyvíjen, v budoucích verzích jazyka se mohou některé funkce změnit nebo se mohou chovat jinak než dosud.

Dříve byly stránky pouze statické a k jejich výrobě se používalo jen samotné HTML. To ovšem přestávalo stačit a do stránek se začaly přidávat programovací jazyky. V případě webu je můžeme rozdělit na klientské a serverové. Každý jazyk má svůj zdrojový kód. V případě klientských jazyků se spolu s HTML kódem do počítače uživatele odešle ze serveru také zdrojový kód daného jazyka, kde se také provede (v prohlížeči). Mezi tyto jazyky patří např. JavaScript nebo Visual Basic Script. Tyto jazyky umějí stránku např. různě animovat. U těchto jazyků je ovšem problém s podporou. Třeba JavaScript je velice rozsáhlý jazyk a některé jeho funkce jsou různě podporované u různých prohlížečů. Proto při programování v JavaScriptu je nezbytné stránku průběžně testovat aspoň v těch nejrozšířenějších prohlížečích.

Naopak u serverových jazyků (mezi které patří třeba PHP, ASP, Python či Ruby) k těmto problémům nedojde. Zdrojový kód se totiž vůbec s prohlížečem nedostane do styku, neboť se provede už na serveru, který odešle výsledek

prohlížeči a tomu je jedno, kde se ty informace vzaly. Server je počítač, který je volně dostupný z internetu, a jsou na něm umístěny vytvořené stránky. Serverem se také označuje aplikace, která vytváří virtuální server. PHP samo o sobě fungovat nemůže, potřebuje právě nějaký ten server/aplikaci. Nejčastěji se u PHP používá server Apache. Z principu fungování PHP vyplývá hned několik výhod. Na rozdíl od klientských jazyků uživatel nemá přístup ke zdrojovému kódu. Tedy co si naprogramujeme, to nám nikdo neodcizí. A navíc díky tomu může PHP pracovat s citlivými daty (osobní data apod.). Soubor s příponou *.php nemusí obsahovat pouze PHP skript. Již z uvedených příkladů je jasné, že se mohou třeba střídat bloky PHP a HTML. Soubor PHP také nemusí PHP obsahovat vůbec. Naopak pokud do souboru HTML přidáme nějaké ty PHP skripty, zůstanou nevyhodnoceny a přijdou prohlížeči tak, jak jsou.

Mezi jeho základní schopnosti patří určitě práce se soubory (čtení i zápis) a práce s databázemi (např. MySQL). PHP si také rozumí s cookies. Jinak je to plnohodnotný programovací jazyk, který má všechny možné podmínky, cykly, proměnné apod. Tyto vlastnosti nám tedy umožňují:

- internetové obchody
- vyhledávání
- diskusní fóra
- redakční systémy
- návštěvní knihy
- různá počítadla, ankety...

Další věc, co PHP umí, je pospojovat stránku z více souborů do jedné. Když se nám tedy nějaká část stránky opakuje neustále (v rámci stránky i v rámci webu), můžeme ji uložit do zvláštního souboru a ten potom vždy do stránky vložit. Příkladem může být třeba menu. Funkčně to nahrazuje rámy, pro prohlížeč to je ale na rozdíl od rámu zcela transparentní, ten nepozná, z kolik jednotlivých souborů je stránka poslepovaná, jemu vždy přijde až celkový výsledek.

2.3 MySQL

Je to databázový systém, jenž umožňuje technologiím PHP a Apache spolupracovat na zpřístupnění a zobrazení dat ve formátu čitelném v internetových prohlížečích. Jedná se o server zpracovávající dotazy ve strukturovaném dotazovacím jazyce (Structured Query Language - SQL) navržený pro zpracování velkého množství velmi složitých dotazů. Jde o relační databázový systém, MySQL tedy umožňuje spojování mnoha různých tabulek. Díky tomu nabízí maximální efektivitu a rychlost. MySQL je dokonalou volbou pro zpřístupnění dat skrze internet, zejména kvůli schopnosti obsloužit velký počet požadavků a také kvůli pokročilým bezpečnostním opatřením. MySQL tedy umožňuje ukládat samostatné informace do *tabulek* neboli do oblastí souvisejících informací. V nerelačních databázových systémech jsou všechny informace uloženy v jedné obrovské oblasti, v níž jsou jakékoli operace mnohem obtížnější a těžkopádnější. To se týká nejen třídění, ale také běžného vyčlenění určité informace. V MySQL se každá tabulka skládá ze samostatných *polí* neboli *sloupců*, které zastupují jednotlivé části uložené informace. V jednom poli je uloženo například jméno uživatele, v dalším poli je třeba jeho

příjmení. Jednotlivá pole mohou nést informace různých typů – například text, čísla, kalendářní data apod.

Datové tabulky budeme vytvářet podle toho, jaké informace v nich chceme ukládat. Jednotlivé tabulky jsou v MySQL propojovány na základě společných jmenovatelů, tedy polí, která obsahují stejné informace. Představme si tedy tabulku, která obsahuje jména, adresy a identifikační čísla zákazníků. Druhá tabulka obsahuje identifikační čísla zákazníků a jejich objednávky. Společným jmenovatelem obou tabulek je tedy sloupec, v němž jsou uložena identifikační čísla zákazníků. Informace uložené v obou tabulkách lze pak spojit právě prostřednictvím tohoto sloupce. Stačí jen vyhledat řádky, v nichž se uložené informace (tedy identifikační čísla zákazníků) shodují.

Několik základních a zároveň neoblíbenějších funkcí MySQL:

- Možnost využití většího počtu procesorů prostřednictvím vláken jádra
- Možnost spouštění nezávisle na hostitelské platformě
- Četné typy sloupců pokrývající prakticky veškeré datové typy
- Skupiny funkcí pro matematické výpočty a třídění
- Příkazy, které umožňují snadný přístup správců k informacím o databázích
- Názvy funkcí, jež nekolidují s názvy tabulek a sloupců
- Systém ověřování uživatelů na základě hesla

2.4 HTML

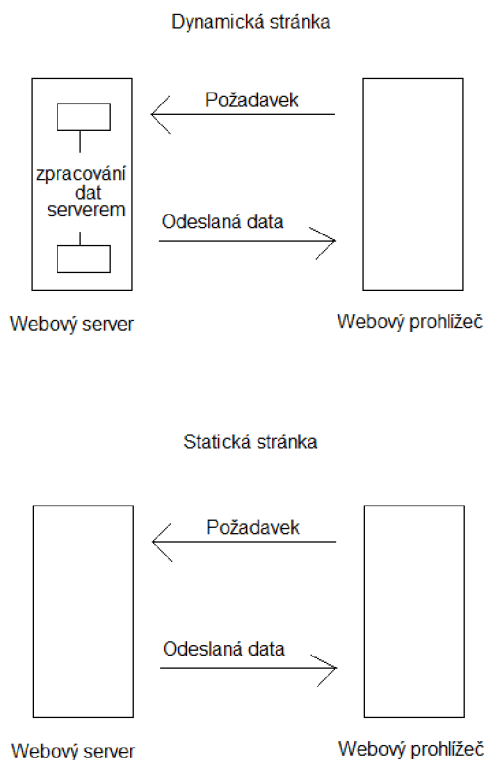
HTML je jednoduchý značkovací jazyk, který slouží k výrobě webových dokumentů. Jeho největší výhodou je univerzálnost. Je tedy jedno, kdo si stránky prohlíží, ať už používá Windows, Linux nebo mobilní telefon, stačí jen, aby měl webový prohlížeč. Postupem času vznikaly různé verze HTML. Ta nejnovější je HTML 4.01, která vznikla již v minulém tisíciletí. Od roku 2004 se už pracuje i na páté verzi, která je zatím stále ve vývoji.

Webové stránky lze vytvářet v libovolném textovém editoru jako je například poznámkový blok. Základní podmínkou je, aby ukládal soubory v čistém textu, tedy editory jako Word jsou nevhodné. Další důležitou vlastností je třeba použití fontu se stejnou šířkou jednotlivých písmen, což zpřehledňuje kód. Obě tyto podmínky poznámkový blok splňuje, proto lze vystačit při návrhu webových stránek pouze s ním. Pokročilé editory však nabízejí třeba barevné zvýraznění jednotlivých částí kódu či automatické doplňování tagů. Kvůli celkové úpravě psaného kódu a jeho přehlednosti je lepší používat editory k tomu určené, jako je například PSPad nebo Notepad.

Dokument HTML může mít několik přípon. Ty nejzákladnější jsou *.htm a *.html, které jsou zároveň nejpoužívanější. Je důležité si jednu variantu vybrat a pak se jí držet v celém projektu. Mezi další přípony patří třeba *.php, *.asp nebo *.xhtml podle toho, jaký programovací jazyk používáme. V zásadě se pod každou příponou skrývá obyčejný textový soubor, který můžeme editovat ve výše uvedených

programech. Pokud bychom používali pouze poznámkový blok, nesmíme zapomenout vytvořený soubor uložit s danou příponou, jinak se nebude otvírat v příslušném prohlížeči, ale opět v normálním textovém editoru. Dokumenty s příponou *.html se označují jako stránky statické a jak již sám název napovídá, nabídne uživateli vždy pouze to, co je jeho obsahem. Pokud tedy do statické stránky pomocí HTML napíšeme větu „Dnes je pondělí ráno a je devět hodin“, pak se tato věta na stránce zobrazí vždy, nehlédě na fakt zda je úterý, středa či čtvrtek, a nehlédě na to kolik je ve skutečnosti hodin. Naproti tomu stránka dynamická používá funkce programovacího jazyka ve kterém je napsána, a poskytuje uživateli údaje dle funkcí, které obsahuje. Takové dokumenty mají převážně příponu *.php. Do takové stránky tedy můžeme napsat např. „Dnes je <?php echo date(“d.m.Y H:i:s”); ?>“ a funkce programovacího jazyka PHP nám každý den a každou hodinu vypíše naprosto přesný datum, hodinu i den. Tyto funkce pak zpracovává webový server, který musí mít podporu pro dynamické stránky.

Pokud se podíváme na rozdíl mezi dynamickou a statickou stránkou pomocí obrázku, bude vypadat asi takto:



Obr.1 – Rozdíl mezi statickou a dynamickou stránkou

Jak je z obrázků patrné, tak u statických stránek odešle webový prohlížeč požadavek na stránku, server ji najde, a odešle ji prohlížeči tak, jak je. To znamená, že stránku odešle v naprosto stejném formátu a obsahu v jakém je uložena. Naproti tomu při požadavku na dynamickou stránku odešle prohlížeč požadavek, server jej zpracuje, vykoná, a teprve potom pošle údaje prohlížeči, který je zobrazí. Díky dynamické stránce se tak dostává uživateli mnohem širšího spektra informací a možností jak se stránkou pracovat. Je tak např. možné na stránku online vkládat komentáře, uploadovat obrázky, odesílat maily pomocí formuláře a mnohé jiné. Nejčastějším programovacím jazykem pro statické stránky je HTML, XHTML s

podporou CSS (Flash a Javu pak vykonávají prohlížeče, nikoliv server), pro dynamické pak PHP či ASP s podporou databází.

Celý HTML kód je jazyk, kterému rozumí jak uživatel, tak i prohlížeč. Tento kód se skládá z tzv. „tagů“ neboli značek, které se píšou v ostrých závorkách <>. Zmíněné tagy se dělí na párové a nepárové. Každý tag má svůj specifický význam, který ovlivňuje výslednou podobu stránky. Každý dokument HTML se musí skládat ze dvou částí: hlavičky a těla. O to se postarají tagy <head> a <body>. Oba to jsou párové tagy, tedy musí někde začínat a někde končit, tedy se skládají ze dvou částí, např. <head> a </head>. Ukončení tagu se dělá pouhým přidáním lomítka.

Celou stránku musíme ještě ohraničit tagem <html>, který říká, že se jedná o dokument HTML. Jednoduchý zdrojový kód prázdné webové stránky tedy vypadá takto:

```
<html>
  <head>
    <title>Toto je jednoduchá stránka</title>
  </head>
  <body>
    Diplomová práce.
  </body>
</html>
```

Do tagu <title></title> se vkládá název dokumentu, neboli to, co se objeví v záhlaví prohlížeče. Tento tag může být umístěn pouze v hlavičce, kde se obvykle nacházejí rozšiřující informace o dané stránce a různé nastavení. Tělo naopak obsahuje to, co se ve výsledku zobrazí na výstupu, tedy v prohlížeči. Stránka tedy bude obsahovat text „Diplomová práce“. Název index.html (či index.htm, index.php atd) se používá pro hlavní stránku v daném adresáři, obvykle hlavní stránku webu. Pokud bychom index.html nahráli na webhosting, stačí pak napsat jen <http://www.nase-adresa.cz/> a zobrazí se index.html. Další stránky se dají potom propojit pomocí (hypertextových) odkazů.

V případě párových tagů by se nikdy nemělo zapomínat na jejich uzavření, vyhneme se tak spoustě problémům. Na druhou stranu, prohlížeče jsou při čtení dosti tolerantní a spoustu věcí jsou si schopné domyslet. Pokud tedy na konci zapomeneme napsat </html>, pravděpodobně se nic nestane. Ovšem z hlediska zápisu je to nesprávné. Časem se kód může stát poměrně rozsáhlým a nepřehledným. Proto je potřeba naučit se dodržovat určitá pravidla, abychom zachovali zpětnou čitelnost. Je poměrně častým zvykem odsazovat každou novou úroveň tagů pomocí tabelátoru. Dále je možnost psát všechny tagy velkými písmeny, čímž také můžeme docílit lepší přehlednosti. Orientaci v kódu nám také usnadní barevné zvýraznění, které umožňují pokročilejší editory určené k vytváření HTML kódu.

Velmi důležitou položkou v HTML jazyku je práce s formuláři a vytváření tabulek. Webový formulář je část stránky složená z různých tlačítek, vstupních polí a prepínačů, která slouží pro získání nějakých dat od uživatele. Na základě těchto dat

se potom provádí skript (napsaný v PHP, Javě nebo v jakémkoli jiném jazyku) a jeho výsledek se jeho prostřednictvím předá uživateli. Obvykle skript formátuje text pomocí HTML značek a svůj výsledek předá prohlížeči, který ho zobrazí. Formulář bývá zajímavým zpestřením všech stránek, ale pokud chceme mít z formuláře nějaký výsledek, neobejdeme se bez skriptů, které data z formuláře zpracují.

Tabulky jsou v HTML velice důležité nejen proto, že se pomocí nich dají velice přehledně prezentovat různé informace, ale i proto, že si s nimi můžeme občas vypomoci i při designu stránek. Na druhou stranu lze místo tabulek používat podporu CSS, avšak nejlepší variantou se jeví kombinace tabulek i CSS. Tabulky musíte uzavřít do párového tagu `<table></table>`, použití ukončovacího tagu je zde vyžadováno. Vlastnosti tabulek můžeme měnit pomocí mnoha atributů. Můžeme tak určit, jak má být na stránce tabulka zarovnána, barvu výplně jednotlivých buněk nebo jejich šířku.

2.5 CSS (Cascading Style Sheets)

Cascading Style Sheets neboli kaskádové styly umožňují rozsáhle formátovat téměř všechny objekty na stránce. Nabízejí prakticky neomezené možnosti formátování a samotná údržba webových stránek je tak velice jednoduchá. Základním důvodem proč používat CSS je především omezená formátovací schopnost HTML. Nastavit u textu font, velikost, řez a barvu je opravdu málo a s některými objekty neuděláte pomocí HTML vůbec nic. CSS umožní upravovat každý prvek na webu, cokoliv může získat rámeček, barvu pozadí, texturu či speciální umístění vůči zbytku stránky. Pokud píšeme dokument v čistém HTML, musíme psát formátování rovnou do textu. To vede k nepřehlednosti kódu a zpětná úprava vzhledu je velmi pracná. Naopak pokud využijeme kaskádních stylů, tak se HTML kód omezí pouze na rozložení stránky ve smyslu: toto je hlavní nadpis, toto je podnadpis, toto je normální odstavec, tento text je důležitý, toto je postranní lišta. K takovému dokumentu pak připojíme externí soubor se styly, který určí, jak má vypadat hlavní nadpis, podnadpis, normální text a že postranní pruh se zobrazí jako postranní. Změny ve vzhledu se pak provádí pouze úpravou CSS souboru, do HTML kódu se vůbec nemusí zasahovat.

CSS můžeme rozdělit na interní a externí. Interní je ta jednodušší možnost, která nevyužívá všechny možnosti CSS. Zadává se přímo do daného objektu. Externí způsob spočívá v tom, že použijeme zvláštní místo (třeba i samotný soubor *.css), v kterém nadefinujete všechny potřebné vlastnosti a přiřadíte je tagům, a který potom zpětně vložíme do každé své HTML stránky. Tedy pokud si nastavíte jako hlavní písmo Verdanu, vyrobíte třeba 100 stránek, a rozhodnete se najednou třeba pro Tahomu, stačí změnit jediné slovo v jediném souboru. Pokud bychom použili tag ``, museli bychom měnit zase všech 100 souborů.

CSS má ovšem taky určité nevýhody. Tou základní je podpora prohlížečů. Tedy nejčastější prohlížeče jako Firefox, MSIE nebo Opera všechny základní parametry podporují, ovšem problém spočívá v tom jakým způsobem. Každý totiž reaguje na danou vlastnost trochu jinak a v celkovém designu se to nakonec projeví docela razantně. CSS se skládá ze selektoru a definice neboli deklarace nebo více

definicí. Definice je (v některých případech) uzavřena ve složených závorkách { } a mezi jednotlivými definicemi se píše středník. V praxi to vypadá takto:

```
<div style="font-family:verdana; color:red;"> </div>
```

V tomto případě je selektorem tag <div>, definice jsou font-family a color. Složené závorky zde použity nejsou, místo toho jsou zde uvozovky. Místo definice se spíše používá výraz vlastnost. Každá vlastnost má svou hodnotu. Bavit se o nějakém tagu nemá smysl, stejně je pokaždé jiný. Ovšem každý tag, ať už je jakýkoliv (tedy pomocí CSS můžeme formátovat téměř každý tag), obsahuje při tomto způsobu parametr style. V tomto parametru jsou jednotlivé vlastnosti. Každá vlastnost má za dvojtečkou svoji hodnotu. Co tyto dvě vlastnosti dělají, je zcela zjevné. Text v tagu div bude verdanou a bude červený. Jinak tyto dvě vlastnosti fungují stejně jako COLOR a FACE v tagu . Tedy vlastnost color může obsahovat barvu jak slovně (red, blue, black...) tak hexadecimálně. A za verdana můžeme dosadit jakékoliv písmo.

3. Tvorba databáze a její obsluha

Abychom mohli generovat testové otázky, musíme mít možnost je někde uschovat. K úschově dat nám poslouží systém databází MySQL. Obsahem kapitoly tedy bude vytváření databáze, která nám bude později sloužit k náhodnému generování jejího obsahu. Nejprve bude nutné databázi navrhnout a vytvořit. Po dokončení takové procedury můžeme začít databázi plnit záznamy, které později můžeme zobrazovat, upravovat nebo zpětně mazat. K výsledkům se můžeme dopracovat dvojím způsobem: Prvním způsobem je v příslušném editoru vytvořit pomocí PHP a MySQL příkazů příslušné funkce, které nám zajistí možnost přidávání, editování a mazání záznamů. Druhou a jednodušší variantou je použití webové aplikace phpMyAdmin, která je řešená graficky a veškeré příkazy k vykonávání příslušných funkcí můžeme generovat pouhým zmáčknutím správného tlačítka. V práci se seznámíme s oběma způsoby vytváření a úpravy databáze.

3.1 Návrh a vytvoření databáze

V prvním kroku se provádí návrh databáze. Nebo-li, vytvoříme vhodné tabulky, do kterých později data ukládáme. Poslouží k tomu primární tabulka, která bude obsahovat následující sloupce: Identifikační číslo (ID) testové otázky, zadání, možnosti odpovědí (a,b,c), označení správné odpovědi, kategorie dané otázky a okruh otázky. Identifikační číslo je nejdůležitější parametr v tabulce, bývá často označován za primární klíč. Pomocí něho budeme schopni později data v tabulce řadit, měnit, mazat atd. Tabulka by mohla vypadat následovně:

ID otázky	Zadání otázky	Možnost A	Možnost B	Možnost C	Správná odpověď	Kategorie otázky	Okruh otázky
1	Může člověk slyšet tón o frekvenci 35Hz?	Ano	Ne	Nelze určit	2	2	1
2	Hmotnost protonu je stejná jako hmotnost?	Elektronu	Fotonu	Neutronu	3	2	2
3	Popiš 6 rozdílů mezi živými a neživými systémy	-	-	-	-	1	3
...
...

Obr.2: Primární tabulka databáze

Protože obsah sloupců správná odpověď, kategorie otázek a okruh otázek se bude pro některé otázky neustále opakovat, jsou vyplněna pouze čísla, která se odkazují na další vytvořené tabulky. Číslo uložené v hlavní tabulce označuje ID odkazu v tabulkách vedlejších. Tabulka pro správnou odpověď obsahuje odkaz na korektní odpověď v zadání. Kategorie otázek značí, jestli bude zadání otázky použito jako testová otázka s možnostmi odpovědí nebo pouze jako zadání otázky na kterou je požadováno odpovědět. Okruh otázek určuje, na jaké téma je otázka vytvořena.

Tabulky budou vypadat následovně:

ID odpovědi	Název odpovědi
1	Odpoved A
2	Odpoved B
3	Odpoved C

ID kategorie	Název kategorie
1	Zadání
2	Test

ID okruhu	Název okruhu
1	Fyzika
2	Chemie
3	Biologie

Obr.3: Další tabulky databáze

Po návrhu uspořádání tabulek se můžeme pustit do vytváření databáze otázek a jejích tabulek. Toho můžeme buď dosáhnout pomocí sady SQL příkazů, kdy je nutné se k databázovému serveru nejprve připojit, vytvořit databázi, označit ji jako pracovní a teprve potom vytvářet tabulky odpovídajícími SQL příkazy. Jednodušší a přehlednější způsob je pro vytváření tabulek využít webové aplikace phpMyAdmin, které je níže věnována podkapitola.

3.2 Vkládání, získávání a zobrazování dat z databáze

K naplnění databáze požadovanými daty můžeme využít buď příkazů sepsaných v textovém editoru nebo webové aplikace phpMyAdmin. Pro ověření funkčnosti všech funkcí použitých v diplomové práci nám postačí databázi naplnit experimentálními údaji, které se budou skládat z otázek určené pro test a zadání, přičemž otázky budou v určitém poměru rozloženy do všech tří vytvořených okruhů. V případě otázek zadání jich bude celkem 15 a budou zastoupeny v okruzích v poměru 3(fyzika) : 7(chemie) : 5(biologie). U testových otázek bude situace řešena podobně, tedy celkový počet testových otázek bude 10 a budou zastoupeny v okruzích v poměru 5(fyzika) : 3(chemie) : 2(biologie).

Jádrum každé databázové aplikace je čtení dat z databáze. Data z databázového serveru MySQL čteme pomocí PHP kódu a přehledně zobrazujeme většinou pomocí tabulek HTML. V MySQL, ostatně jako v každé jiné databázi obsahující jazyk SQL, se data z databáze vybírají pomocí příkazů. PHP používá na zadávání příkazu pro výběr požadovaných dat svoje funkce, pomocí nichž databáze vrátí tzv. sadu záznamů, kterou budeme chtít dále zpracovávat. Pro zobrazení dat potom používáme funkce, pomocí kterých načteme požadovaný řádek do asociativního nebo číselného pole a v cyklu můžeme data řádek po řádku vypisovat. Poměrně časté je vypisovat sady záznamů do tabulek. Není to vůbec složité, protože kód PHP se s HTML může proplétat. Výsledek celé sady příkazů v kombinaci PHP a HTML kódu by byl pro seznam pouze testových otázek zobrazen na obr. 4. Obdobně by vypadal seznam pro otázky zadání (obr. 5) a seznam okruhů je zobrazen na obr. 6. Vybrali jsme tedy sadu záznamů, kterou jsme navíc seřadili podle kategorie otázek a poté jsme ji řádek po řádku vkládali do tabulky HTML.

Databáze testových otázek

Zadání	Možnost A	Možnost B	Možnost C	Okruh otázek
Litr vody a litr lihu. Vysledný objem bude (za předpokladu, že ani kapka nepadne vedle?)	přesně 2 litry	o něco méně než 2 litry	o něco více než 2 litry	Fyzika
Velikost poloměru horizontu událostí závisí na?	hmotnosti černé díry	na elektrickém náboji černé díry	na momentu hybnosti černé díry	Fyzika
Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i za rohem?	vlnová délka	frekvence	rychlost	Fyzika
Může průměrný člověk slyšet zvuk s frekvencí 35 kHz	Ano	Ne	Záleží na teplotě prostředí, v němž se člověk nalézá	Fyzika
Auto A se pohybuje rychlostí v . Auto B se pohybuje rychlostí $2v$. Kolikrát větší kinetickou energii má auto B než auto A?	dvakrát větší	čtyřikrát větší	osmkrát větší	Fyzika
Atom obsahuje?	atomové jádro (v něm protony) a elektronový obal (elektrony a neutrony)	atomové jádro (v něm protony a neutrony) a elektronový obal (v něm elektrony)	atomové jádro (v něm neutrony) a elektronový obal (v něm protony a elektrony)	Chemie
Hmotnost protonu je přibližně stejná jako hmotnost?	elektronu	fotonu	neutronu	Chemie
Nukleonové číslo udává počet?	nukleonů (= protonů a elektronů dohromady)	nukleonů (= protonů a neutronů dohromady)	nukleonů (= protonů, neutronů a elektronů dohromady)	Chemie
Živočich, který má samčí a samičí pohlaví se nazývá?	obojetník	oboupohlavník	homosexuál	Biologie
Červené krevní krvinky obsahují červené krevní barvivo?	fibrinogen	hemoglobin	globulín	Biologie
Celkem otázek: 10				

Obr. 4: Seznam testových otázek

Databáze písemných otázek

Zadání	Okruh otázek
Kolem výpravního stojáka na začátku nástupiště projel vlak o délce 250 m za 20 s. Za jak dlouho projede tento vlak podél nástupiště o délce 150 m?	Fyzika
Klec osobního výtahu ve čtrnáctiposchodovém obytném domě má hmotnost 150 kg. Uveze tři dospělé osoby o celkové hmotnosti 250 kg. Ze sklepa až do nejvyššího podlaží urazí klec dráhu 45 m za dobu 90 s. Někdy jede výtah prázdný, jindy se třemi osobami. Jakou silou je napínáno lano, je-li výtah v klidu nebo pohybuje-li se rovnoměrně?	Fyzika
O hydroelektrárně na řece Volha u města Volgograd víme, že má nejvyšší výkon 2540 MW, řeka má střední objemový tok 8 000 m ³ /s. Energie vodního toku lze využít na 60 %. Jak vysoko musí být hladina přehrady nad vstupem vody do turbín?	Fyzika
Napište vzorec pevné karboxylové kyseliny a kapalné karboxylové kyseliny. Nazvěte tyto kyseliny a porovnejte jejich sílu.	Chemie
Jak probíhá hydrolyza soli a karboxylových kyselin? Popište rovnici.	Chemie
Napište vzorec karboxylové kyseliny, která zapáchá a karboxylové kyseliny bez zápachu. Nazvěte tyto kyseliny a porovnejte jejich sílu.	Chemie
Popište rovnici důkaz aldehydů pomocí Fehlingova činidla.	Chemie
Jaké známe funkční deriváty karboxylových kyselin? Názvy a obecné vzorce!	Chemie
Co vzniká oxidací acetonu? Popište chemickou reakci.	Chemie
Co vzniká reakcí alkoholu a aldehydu? Popište konkrétní reakci.	Chemie
Popiš alespoň 6 rozdílů mezi živými a neživými systémy a vysvětli je.	Biologie
Popiš základní chemické složení buňky (atomy a rozdělení látky) a vysvětli slovo biogenita.	Biologie
Vyjmenuj a popiš význam energeticky bohatých nízkomolekulárních látek v buňce.	Biologie
Popiš stavbu a funkci struktur buňky, které mají zásadní význam pro energetické pochody v buňkách (stavba a význam mitochondrie a plastidu).	Biologie
Vysvětli princip mezozy, kde tento proces probíhá a co zajišťuje?	Biologie
Celkem otázek: 15	

Obr. 5: Seznam otázek zadání

Okruhy testových otázek

Fyzika
Chemie
Biologie
[Zpět do hlavního menu]

Obr. 6: Použité okruhy testových otázek

3.3 Úprava dat v databázi

Z databáze již umíme záznamy vybírat a umíme je do ní vkládat. To nám však nestačí, protože chceme mít také možnost záznamy editovat (aktualizovat) nebo mazat. Abychom však vytvořili prostředí jednoduché na obsluhu, je nezbytné, aby vstupní data zadával sám uživatel. Tato data jsou přitom shromažďována prostřednictvím formulářů. Webový formulář je část stránky složená z různých tlačítek, vstupních polí a přepínačů, která slouží pro získání nějakých dat od uživatele. Na základě těchto dat se potom provádí skript (napsaný v PHP, Javě nebo v jakémkoli jiném jazyku) a jeho výsledek se nějak předá uživateli. Obvykle skript formátuje text pomocí HTML značek a svůj výsledek předá prohlížeči, který ho zobrazí. Skript PHP přijímá data z formulářů a používá je k vykonání určité operace – například k přidání nového záznamu, k aktualizaci záznamů, mazání záznamů v databázi, atd.

Formulář je na stránce označen pomocí párového tagu `<form></form>`. Tímto způsobem určíme, které formulářové prvky (vstupní pole, tlačítka atd.) patří k sobě a budou se odesílat najednou - jednomu řídicímu skriptu. Jednotlivých formulářů může být na stránce libovolně mnoho, ale nesmí se navzájem nijak prolínat. Element `<form>` má několik atributů, které ovlivňují jeho chování. Nejdůležitějšími atributy, které budeme potřebovat vždy při použití formulářů, jsou `action=""` a `method=""`. Atributem `action=""` určíte, který skript se má použít pro zpracování zadaných údajů. Hodnota atributu `method=""` určuje metodu, jakou mají být data odeslána ke zpracování. Na výběr je ze dvou možností: GET nebo POST.

Metoda GET se používá spíše pro odesílání menších formulářů a tam kde příliš nezáleží na bezpečnosti, protože pokud odesíláme data metodou GET, je velice jednoduché je zfalšovat a podstrčit skriptu nějaké nekorektní údaje. To platí i u metody POST, ale tam je podstrčení špatných dat o malinko složitější. Metoda POST kóduje zadaná data stejně jako GET, ale posílá je zvlášť - mimo URL. Veškerá data se rovnou směřují na standardní vstup skriptu. Tato data se sice nikde nezobrazí, ale to neznamená, že je nejde změnit.

Příklad:

```
<form action="odeslat.php?action=pridat" method="post">
```

Předáme tak skriptu *odeslat.php* vstupní proměnnou *action* (s hodnotou „*pridat*“) odeslanou metodou POST.

3.3.1 Vkládání nových otázek do databáze

Pokud bychom chtěli vkládat nové záznamy do databáze, a přitom nechtěli využít webového prostředí phpMyAdmin musíme vytvořit vhodné jednoduché rozhraní, ve kterém si i nezkušený uživatel poradí s přidáváním nových záznamů do databáze. Prostředí vyhovující požadavkům je zobrazeno na obr. 7. Zobrazený formulář pro vyplnění nového záznamu slouží v tomto případě pro přidání nové testové otázky.

Přidat testovou otázku

Zadání	<input type="text"/>
Možnost A	<input type="text"/>
Možnost B	<input type="text"/>
Možnost C	<input type="text"/>
Správná odpověď	<input type="text" value="Vyberte správnou odpověď"/>
Okruh otázky	<input type="text" value="Vyberte okruh otázky"/>
<input type="button" value="Přidat"/>	

Obr. 7: Formulář pro přidání nové testové otázky

Pro přidání otázky zadání vypadá prostředí podobně, jen zde již není možnost vkládat možnosti odpovědí a vybírat správnou odpověď. Po vyplnění formuláře informacemi o nové testové otázce jej tlačítkem „Přidat“ odešleme ke zpracování skriptem, který prostřednictvím MySQL příkazů provede zápis nové otázky do databáze.

3.3.2 Editace otázek v databázi

Při editaci otázky už je situace trochu složitější, protože než vybereme otázku, je nutné zaznamenat identifikační číslo (ID) otázky, kterou se chystáme aktualizovat. Toho dosáhneme tak, že po označení požadované otázky, kterou chceme editovat, se načtení příslušného řádku v databázi provede pomocí informace o identifikačním čísle otázky, kterou současně odešleme stránce, na které se ji chystáme zobrazit. Okno pro editaci otázky s okruhu fyzika a se správnou odpovědí B pak vypadá následovně:

Úprava testové otázky

Zadání	Litř vody a litř lihu. Vysledný objem bude (za předpokladu, že ani kapka nepadne vedle?)	
Možnost A	přesně 2 litry	
Možnost B	o něco méně než 2 litry	
Možnost C	o něco více než 2 litry	
Správná odpověď	Odpověď B ▼	
Okruh otázky	Fyzika ▼	Změnit otázku
	Chemie Biologie	
Vymazat otázku		

Obr.8: Formulář pro úpravu testové otázky

Po úpravě formuláře jej tlačítkem „Změnit“ odešleme ke zpracování skriptem, který provedené změny uloží do databáze a nahradí tak původní hodnoty.

3.3.3 Mazání záznamů v databáze

Odstranění znamená vždy ztrátu dat. K odstranění záznamu musíme přesně určit, o který záznam v databázi jde. Jakmile pak jednou odstraňovací záznam spustíme, záznamy budou odstraněny bez jakékoli naděje na obnovu.

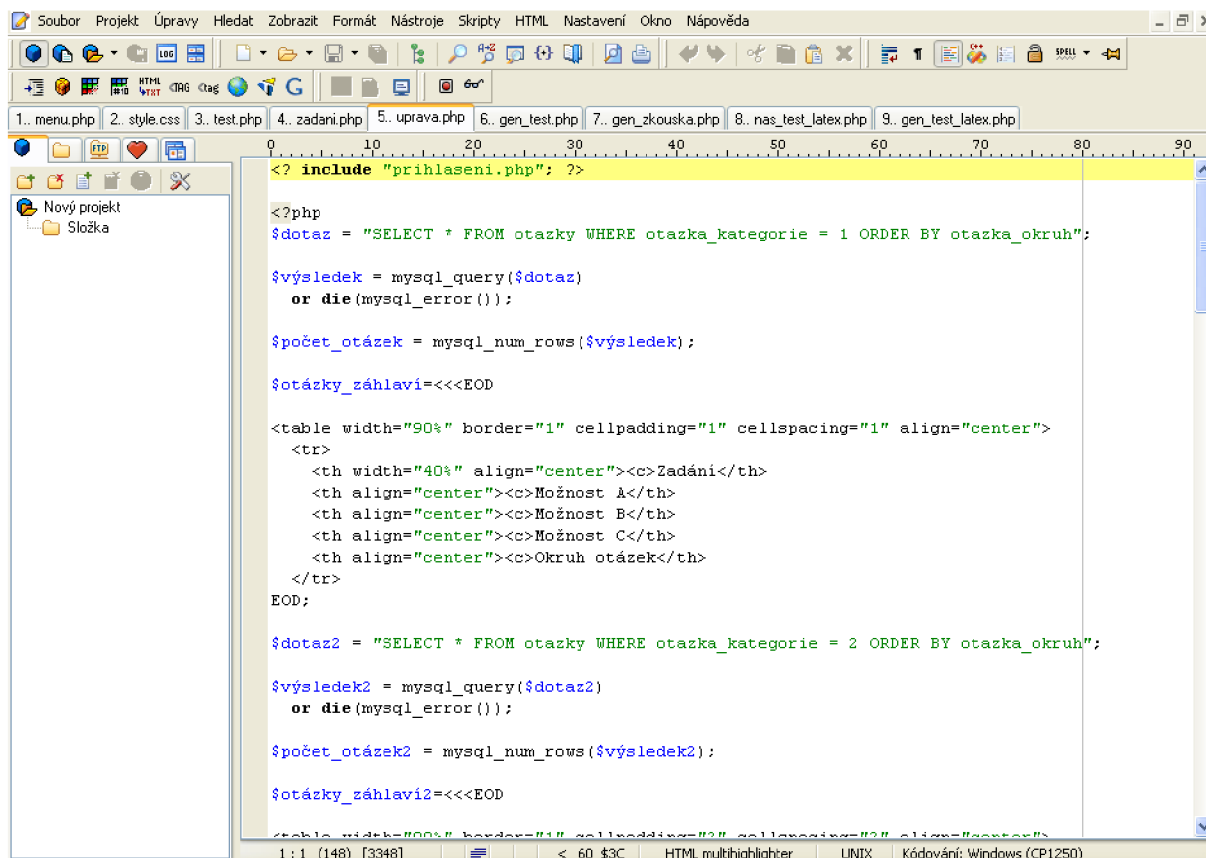
Stejně jako v případě úpravy záznamu, odesíláme vykonávacímu skriptu údaj o identifikačním čísle. Mazací procedura potom vybere řádek s odpovídajícím ID a řádek odstraní.

3.4 PSPad editor

Velmi užitečný editor určený programátorům nejrůznějších programovacích jazyků. Program neobsahuje nekonečnou řadu zbytečných funkcí, které většina z uživatelů stejně nikdy nevyužije a soustředí se spíše na jednoduchost, přehlednost a rychlost. Velmi zajímavá je funkce šablony, která umožňuje definovat určitou část programové kódu (či něčeho obdobného) pod definovaný příkaz a při programování tento kód rychle vyvolat. Pro započítí používání této funkce a nadefinování příslušných bloků kódu je nutno si pročíst nápovědu, která je velmi pěkně zpracována. V poslední verzi byla přidána možnost definice vlastních klávesových zkratk pro určité příkazy, což je velmi oblíbená funkce. Tento program není nutné instalovat a po stažení ho stačí pouze spustit. Program podporuje přes 30 různých prostředí (PHP, HTML, XML, ASP, SQL, Python, atd.), v kterých výrazně ulehčí práci.

Krátce k funkcím:

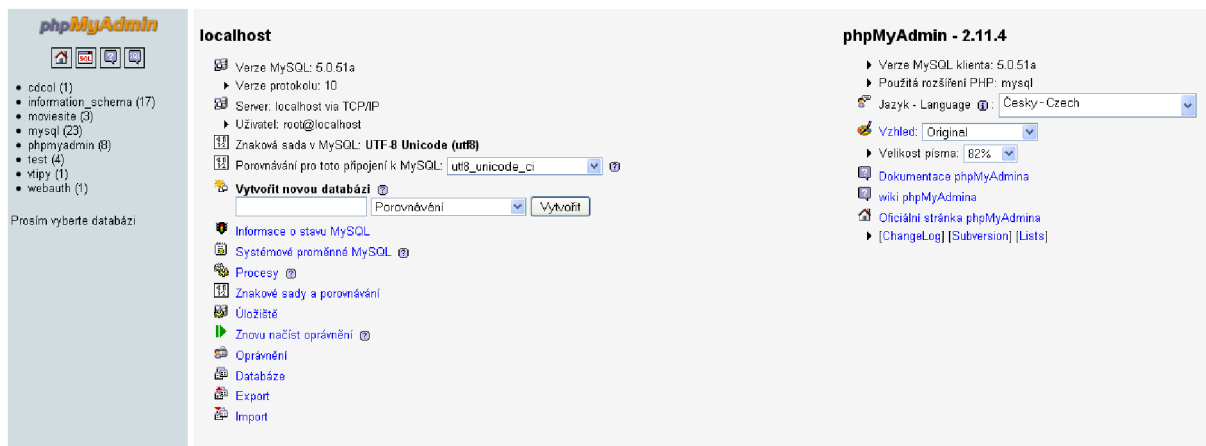
- editace v UNICODE,
- práce se dvěma soubory současně v rozděleném okně, včetně možnosti uzamknout posuv,
- neomezená délka editovaného textu,
- FTP client - umožňuje editovat soubory přímo z webu,
- HEXa editor s možností vkládání,
- umožňuje otevřít víc dokumentů současně,
- šablony (např. HTML tagy, skripty, kusy kódu atd.) jsou uživatelsky definovatelné,
- konverze češtiny (LatinII, Kamenických, Windows 1250, ISO 8859-2, UNICODE, UTF-8),
- kontrola pravopisu, doplnění diakritiky,
- konverze formátu DOS, MAC, UNIX,
- přímé uložení souboru na disketu, do obsahu e-mailu nebo jako příloha e-mailu,
- zvýraznění syntaxe s automatickým nastavením dle typu souboru, zvýraznění skriptů v rámci HTML,
- tisk zvýrazněné syntaxe s možností náhledu před tiskem,
- porovnání obsahu textových souborů,
- export včetně zvýraznění do RTF a HTML, souboru nebo přes schránku, import z RTF do HTML,
- definice externích programů, ve kterých je možné soubor otevřít, kompilace...,
- sloupcové a řádkové bloky, záložky v textu,
- jednoduchá integrace do kontextové nabídky systému Windows,
- přeformátování, komprese a kontrola HTML kódu,
- interní HTML prohlížeč,
- změny velikosti písmen, odstranění a přidání diakritiky, formátování kódu, kontrola pravopisu,
- zobrazení ASCII tabulky s možností tisku,



Obr.9: Okno textového editoru PSPad

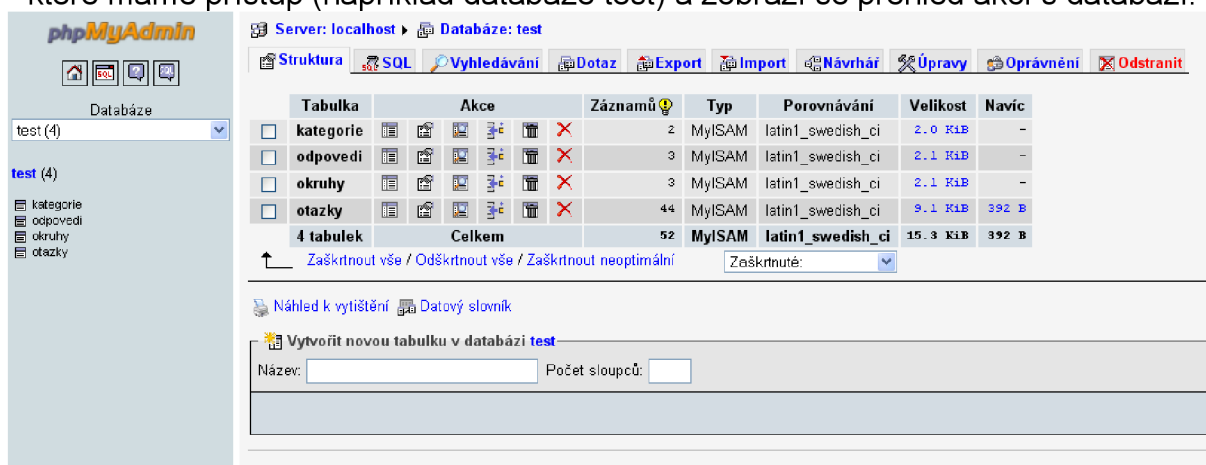
3.5 phpMyAdmin

PhpMyAdmin je webová aplikace umožňující základní správu s databází MySQL a jejich tabulkami. Umožňuje také administrátorům spravovat možné uživatele databáze a jejich přístupová práva. Rozhraní je opticky rozděleno na dvě části - levý sloupec, ve kterém je seznam databází a tabulek a některé ovládací prvky a zbývající pravá část, kde se odehrává veškerá manipulace s daty. Na první stránce můžeme provést některá nastavení, provádět administraci MySQL a také zde jsou odkazy na dokumentaci:



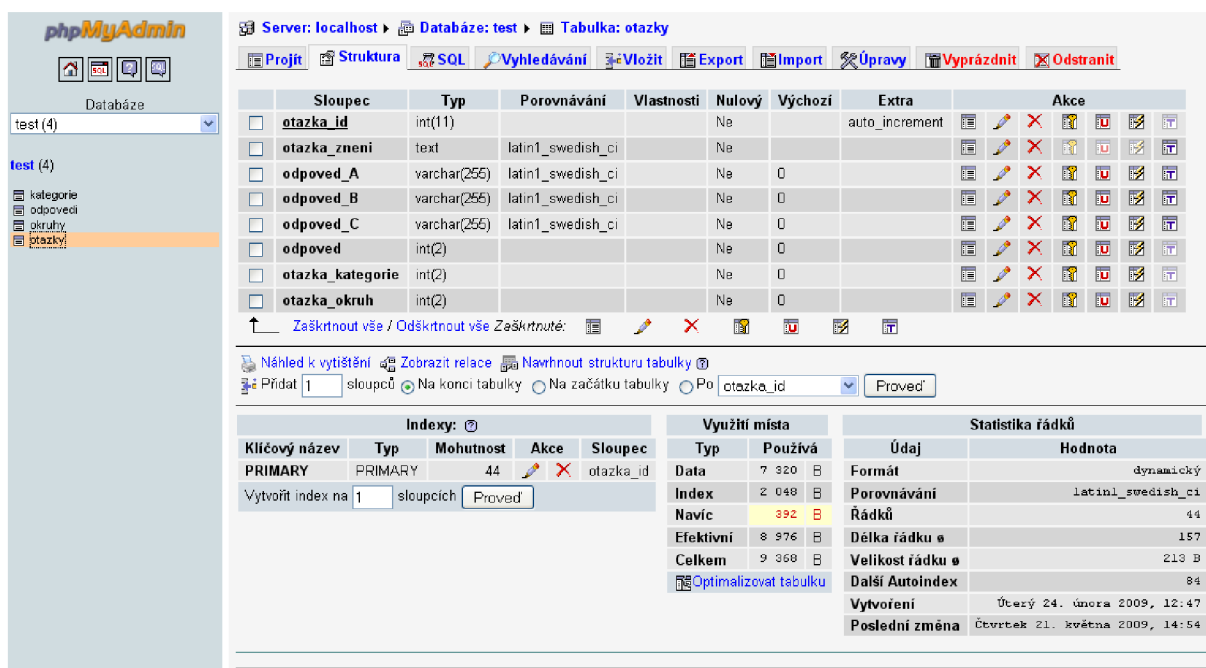
Obr.10: Hlavní okno webové aplikace phpMyAdmin

Můžeme zde vytvořit databázi, zobrazit informace o serveru a provést různá nastavení rozhraní. Pro základní prohlídku si v levém sloupci vybereme databázi, do které máme přístup (například databáze test) a zobrazí se přehled akcí s databází:

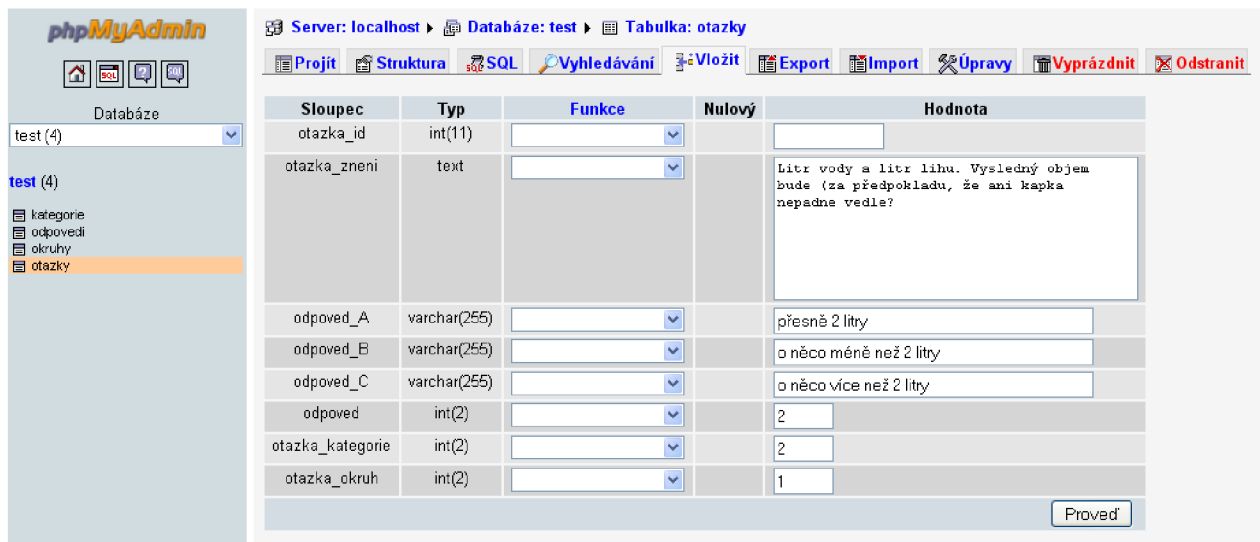


Obr.11: Vytvořená databáze „test“ a její tabulky

Na obr.11 je zobrazeno okno ze zmiňovaného prostředí, ve kterém se procedury vytváření databáze a tabulek vykonávaly. Byla tedy vytvořena databáze s názvem „test“ s odpovídajícími tabulkami. Databáze test obsahuje čtyři tabulky: *kategorie*, *odpovedi*, *okruhy* a *otazky*. Jednotlivé tabulky můžeme různě procházet a prohlížet si tak jejich sloupce. Například tabulka *otazky* se skládá ze sloupců: *otazka_id*, *otazka_zneni*, *odpoved_A*, *odpoved_B*, *odpoved_C*, *odpoved*, *otazka_kategorie* a *otazka_okruh*. Pokud je něco ve sloupcích tabulky již uloženo, můžeme jejich obsah pomocí phpMyAdmin prohlížet, měnit, mazat nebo přidávat záznamy nové.



Obr.12: Struktura sloupců v tabulce „otazky“



Obr. 13: Okno pro přidání nové testové otázky

phpMyAdmin umožňuje při práci s MySQL především:

- Vytvoření a odstranění databáze
- Vytvoření tabulky, její kopírování, přejmenování a odstranění
- Speciální funkce prováděné nad tabulkou (změna typu, optimalizace)
- Údržba struktury tabulky včetně indexů
- Vkládání, upravování a editaci dat
- Nahrávání binárních dat
- Vyhledávání dat v tabulce i v celé databázi
- Zobrazení dat

Pro administrátory jsou pak funkce jako:

- Správa uživatelů a jejich oprávnění
- Kontrola databázových oprávnění
- Stav serveru
- Export databází a to i více najednou

4. Generování otázek

Celá předchozí kapitola se věnovala obsluze navrhnuté databáze a prostředí pro vkládání, úpravu a mazání použitých dat v databázi. Dalším krokem bude využití informací z databáze k jejich náhodnému generování. V databázi jsou uloženy otázky, které se dají třídít podle kategorie a okruhu do kterého spadají. Nabízí se tedy možnost z databáze náhodně generovat sadu otázek z několika okruhů, případně podle typu generovaného testu. Aby bylo dosaženo takového výsledku, je nezbytné vytvořit celou řadu funkcí, která zajistí již zmiňovaný náhodný výběr otázek a také jejich odpovědí v případě generování testu nebo zkoušky.

4.1 Funkce pro náhodné generování

Před tím než začneme vybírat z databáze data, je třeba zjistit kolik otázek ve zvolené kategorii a okruhu je k dispozici. Protože otázky rozdělené podle okruhů jsou v databázi uloženy v různém počtu, při výběru musíme zajistit, aby celkový počet otázek ze všech možných okruhů byl v zadání generován podle četnosti rozložení otázek pod těmito okruhy uložených. Jinak řečeno poměr počtu generovaných otázek z jednotlivých okruhů by se měl shodovat s poměrem celkovému počtu otázek v databázi rozdělených podle okruhů. Další velmi důležitý úkol bude plnit právě funkce, která nám zajistí, aby otázky po výběru z databáze byly skutečně zobrazovány náhodně. Při generování testu nebo zkoušky je nutno navíc zajistit také náhodnou generaci možných odpovědí a dále vytvořit skript, který nám pro aktuální generovaný test nebo zkoušku zobrazí správné řešení otázek. Pro identifikaci jednotlivých zadání a správných odpovědí nám poslouží funkce pro generování náhodné posloupnosti znaků, která bude unikátní pro každé nově vytvořené zadání a bude stejná pro sadu správných odpovědí. Vyjádření funkce pro vytvoření unikátního identifikačního čísla generovaného zadání je vyřešeno v příloze. Hlavní podstatou skriptu je nadefinování proměnné, která obsahuje určitý počet znaků, které chceme při generaci náhodného čísla použít a potom pomocí počítadla generovat náhodně tyto znaky a ukládat si je do nové proměnné, která představuje vygenerovaný kód. Jeho délka je dána nastavením počítadla, a dokud není dosaženo požadované délky, cyklus se neustále opakuje.

Hlavní nabídka pro generování otázek je zobrazena na obr. 14. V levé části jsou odkazy na procedury týkající se obsluhy databáze zmiňované v předchozí kapitole a v prostřední části máme na výběr ze tří možností generace zadání. Další záložku tvoří Export zadání prostřednictvím LaTeXu, který je popsán později. Dále se tu ještě vyskytuje záložka Hlavní menu a Okruhy otázek.



Obr. 14: Menu aplikace pro generování otázek – záložka Hlavní menu

4.2 Generování zadání pro písemku

Pokud se zabýváme vytvářením náhodného zadání pro písemku, musíme databázi sdělit požadavky, které nám vyčlení pouze záznamy, které mohou být pro požadovanou akci použity. Nejprve tedy vybereme všechny záznamy, která jsou v databázi uložena v kategorii označená jako zadání. Po načtení jejich obsahu si spočítáme jejich celkový počet a také počet otázek odpovídající jednotlivým okruhům. Ten poslouží právě pro funkci zajišťující výpočet a náhodné generování zvoleného počtu otázek uživatelem z všech možných dostupných okruhů splňující danou kategorii zadání. Před samotným výpočtem si nejprve uživatel definuje počet otázek, který chce v písemce generovat a počet bodů za jednu otázku. Formulář pro zadávání zmíněných požadavků a výsledek generování je zobrazen na obr. 15 a obr. 16. Potvrzovací tlačítko nás přesměruje na skript, který zajistí pomocí příslušných funkcí vygenerování správného počtu otázek v závislosti na počtu otázek z jednotlivých okruhů.

Nastavení generování písemky

Počet generovaných otázek v písemce (1 - 15)	<input type="text" value="7"/>
Počet bodů za jednu otázku	<input type="text" value="5"/>
<input type="button" value="Generovat zadání"/>	

Obr. 15 – Nastavení generování písemky

Automatické generování testových otázek



ÚREL
ÚSTAV RADIOELEKTRONIKY

[Hlavní menu](#)[Úprava databáze](#)[Generování zadání](#)[Export zadání v LATEXU](#)[Okruhy otázek](#)

(Identifikační číslo: p70zch3n)

Zadání písemné práce

(Písemka se skládá celkem z 7 otázek. Každá otázka je hodnocena maximálně 5 body. Celkem tedy lze s písemky získat 35 bodů.)

1. Kolem výpravčího stojícího na začátku nástupiště projel vlak o délce 250 m za 20 s. Za jak dlouho projede tento vlak podél nástupiště o délce 150 m?
2. Napište vzorec karboxylové kyseliny, která zapáchá a karboxylové kyseliny bez zápachu. Nazvěte tyto kyseliny a porovnejte jejich sílu.
3. Jaké známe funkční deriváty karboxylových kyselin? Názvy a obecné vzorce!
4. Jak probíhá hydrolyza soli a karboxylových kyselin? Popište rovnici.
5. Popište rovnici důkaz aldehydů pomocí Fehlingova činidla.
6. Popište stavbu a funkci struktur buňky, které mají zásadní význam pro energetické pochody v buňkách (stavba a význam mitochondrie a plastidu).
7. Vyjmenuj a popište význam energeticky bohatých nízkomolekulárních látek v buňce.

Copyright © 2009 Designed by Pavel Čídl [Vysoké učení technické v Brně](#)

Obr. 16 – Generování zadání pro písemku

Zkušební databáze obsahuje celkem 15 otázek zadání v poměru 3:7:5 otázek na jednotlivé okruhy. V našem případě bylo požadováno generovat 7 otázek a byly vygenerovány otázky z okruhů v poměru 1:4:2. Výsledek tedy odpovídá četnosti rozložení otázek v databázi.

4.3 Generování zadání pro test

Při generování testu využíváme stejné funkce, jako tomu bylo v případě generování otázek pro zadání. Způsob výpočtu generovaných otázek je tedy shodný, avšak u testu je nutno také zajistit, aby i možnosti odpovědí byly generovány náhodně a nezáleželo tak na pořadí v jakém jsou uloženy v databázi. Další požadavek, který také nutno splnit je zobrazování správných odpovědí pro generované otázky. Pro náhodné zobrazování odpovědí vycházíme z původní funkce, která prošla řadou úprav. Funkce slouží pro generování náhodných čísel, které jsou od sebe navzájem různé. 1. parametr je počet potřebných čísel. 2. parametr je maximální hodnota generovaných čísel. Funkce defaultně vypisuje pod sebe na obrazovku, zároveň však přímo může vrátit pole s těmito hodnotami. Podrobněji je skript popsán v příloze. Nastavení generace testu a následné zobrazení zadání testu je zachyceno na obr. 17 a obr. 18:

Automatické generování testových otázek

UREL
ÚSTAV RADIOELEKTRONIKY

Hlavní menu Úprava databáze **Generování zadání** Export zadání v LATEXU Okazy na literaturu

- Generování testu
- Generování písemky
- Generování zkoušky

Nastavení generace testu

Počet generovaných otázek v testu (1 - 10)	<input type="text" value="8"/>
Počet bodů za jednu otázku	<input type="text" value="5"/>
<input type="button" value="Generovat test"/>	

Copyright © 2009 Designed by Pavel Cídl [Vysoké učení technické v Brně](#)

Obr. 17 – Nastavení generace testu

Zadání testu

(Test se skládá celkem z 8 otázek. Každá testová otázka je hodnocena 5 body. Celkem tedy lze s testu získat 40 bodů.)

1. Auto A se pohybuje rychlostí v . Auto B se pohybuje rychlostí $2v$. Kolikrát větší kinetickou energii má auto B než auto A?

- A) dvakrát větší
- B) čtyřikrát větší
- C) osmkrát větší

2. Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i za rohem?

- A) vlnová délka
- B) frekvence
- C) rychlost

3. Litř vody a litř lihu. Vysledný objem bude (za předpokladu, že ani kapka nepadne vedle)?

- A) o něco více než 2 litry
- B) přesně 2 litry
- C) o něco méně než 2 litry

4. Můžė průměrný člověk slyšet zvuk s frekvencí 35 kHz

- A) Záleží na teplotě prostředí, v němž se člověk nalézá
- B) Ne
- C) Ano

5. Nukleonové číslo udává počet?

- A) nukleonů (= protonů a neutronů dohromady)
- B) nukleonů (= protonů a elektronů dohromady)
- C) nukleonů (= protonů, neutronů a elektronů dohromady)

6. Hmotnost protonu je přibližně stejná jako hmotnost?

- A) elektronu
- B) fotonu
- C) neutronu

7. Atom obsahuje?

- A) atomové jádro (v něm protony) a elektronový obal (elektrony a neutrony)
- B) atomové jádro (v něm protony a neutrony) a elektronový obal (v něm elektrony)
- C) atomové jádro (v něm neutrony) a elektronový obal (v něm protony a elektrony)

8. Červené krevní krvinky obsahují červené krevní barvivo?

- A) fibrinogen
- B) globulín
- C) hemoglobin

Obr. 18 – Generování zadání pro test

I v případě testových otázek proběhla generace podle četnosti rozložení otázek v databázi. V posledním kroku zbývá ještě vyřešit zobrazování správných odpovědí. Toho lze dosáhnout definicí několika podmínek, které nám zajistí označení správné odpovědi i přesto, že z databáze jsou odpovědi generovány zcela náhodně. Ukázka řešení skriptu je opět zachycena v příloze.

SPRÁVNÉ VÝSLEDKY TESTU	
Identifikační číslo: 13895ydn	
Zadání	Správná odpověď
1. Auto A se pohybuje rychlostí v . Auto B se pohybuje rychlostí $2v$. Kolikrát větší kinetickou energii má auto B než auto A?	čtyřikrát větší
2. Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i za rohem?	vlnová délka
3. Litr vody a litr lihu. Výsledný objem bude (za předpokladu, že ani kapka nepadne vedle)?	o něco méně než 2 litry
4. Může průměrný člověk slyšet zvuk s frekvencí 35 kHz	Ne
5. Nukleonové číslo udává počet?	nukleonů (= protonů a neutronů dohromady)
6. Hmotnost protonu je přibližně stejná jako hmotnost?	neutronu
7. Atom obsahuje?	atomové jádro (v něm protony a neutrony) a elektronový obal (v něm elektrony)
8. Červené krevní krvinky obsahují červené krevní barvivo?	hemoglobin

Obr. 19 – Tabulka správných odpovědí

4.4 Generování zadání pro zkoušku

V případě požadavku generování zkoušky vybereme z databáze otázky pro zadání i pro test. Zkouška se tedy bude skládat z dvou částí: z písemné části a testové. Náhodné generování obsahu zkoušky probíhá podle předchozích poznatků, které byly použity při generování testu a písemky. Zbývá tedy odeslat požadavek podle obr. 20, ve kterém je nutno vyplnit počet otázek pro danou kategorii a bodové ohodnocení za jednu otázku. Při potvrzení tlačítka „Generovat zkoušku“ se vykonají funkce pro náhodnou generaci zadání a testu dohromady, stejně jako v předchozích podkapitolách. Výsledkem tedy bude zadání zkoušky (obr. 21), kde pro testovou část zkoušky je nutno také zobrazit jejich správné výsledky (obr. 22).

Nastavení generování Zkoušky	
Počet generovaných testových otázek ve zkoušce (1 - 10)	<input type="text" value="5"/>
Počet bodů za jednu testovou otázku	<input type="text" value="3"/>
Počet generovaných otázek zadání ve zkoušce (1 - 15)	<input type="text" value="5"/>
Počet bodů za jednu otázku zadání	<input type="text" value="2"/>
<input type="button" value="Generovat zkoušku"/>	

Obr. 20 – Nastavení generování zkoušky

Zadání zkoušky

(Zkouška se skládá celkem z 5 testových otázek a z 5 otázek zadání. Každá testová otázka je hodnocena 3 body a otázka zadání je hodnocena maximálně 2 body. Ze zkoušky je celkem možné získat maximálně 25 bodů.)

1. Klec osobního výtahu ve čtrnáctiposchodovém obytném domě má hmotnost 150 kg. Uveze tři dospělé osoby o celkové hmotnosti 250 kg. Ze sklepa až do nejvyššího podlaží urazí klec dráhu 45 m za dobu 90 s. Někdy jede výtah prázdný, jindy se třemi osobami. Jakou silou je napínáno lano, je-li výtah v klidu nebo pohybuje-li se rovnoměrně?

2. Popište rovnici důkaz aldehydů pomocí Fehlingova činidla.

3. Co vzniká reakcí alkoholu a aldehydu? Popište konkrétní reakci.

4. Popiš stavbu a funkci struktur buňky, které mají zásadní význam pro energetické pochody v buňkách (stavba a význam mitochondrie a plastidu).

5. Popiš alespon 6 rozdílů mezi živými a neživými systémy a vysvetli je.

6. Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i za rohem?

A) vlnová délka

B) rychlost

C) frekvence

7. Velikost poloměru horizontu událostí závisí na?

A) na elektrickém náboji černé díry

B) na momentu hybnosti černé díry

C) hmotnosti černé díry

8. Může průměrný člověk slyšet zvuk s frekvencí 35 kHz

A) Ano

B) Ne

C) Záleží na teplotě prostředí, v němž se člověk nalézá

9. Hmotnost protonu je přibližně stejná jako hmotnost?

A) fotonu

B) elektronu

C) neutronu

10. Červené krevní krvinky obsahují červené krevní barvivo?

A) hemoglobin

B) fibrinogen

C) globulín

Obr. 21 – Generování zadání pro zkoušku

SPRÁVNÉ VÝSLEDKY TESTOVÝCH OTÁZEK	
Identifikační číslo: dc8pkwvm	
Zadání	Správná odpověď
6. Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i za rohem?	vlnová délka
7. Velikost poloměru horizontu událostí závisí na?	hmotnosti černé díry
8. Může průměrný člověk slyšet zvuk s frekvencí 35 kHz	Ne
9. Hmotnost protonu je přibližně stejná jako hmotnost?	neutronu
10. Červené krevní krvinky obsahují červené krevní barvivo?	hemoglobin

Obr. 22 – Tabulka správných odpovědí pro zkoušku

4.5 Export zadání pomocí systému LaTeX

Program TeX představuje v podstatě pouze program (kompilátor) pro převod zdrojových příkazů uživatele do výstupního grafického souboru ve formátu DVI (DeVice Independent). LaTeX představuje nadstavbu TeXu o balíčky předdefinovaných maker, která usnadňují a zpřístupňují složitý jazyk TeXu pro sazbu dokumentů širšímu spektru uživatelů. Uživatel se tak může více soustředit na obsah, protože o formu se starají přichystaná makra. Pokud potřebujeme využívat českou sazbu, použijeme příkazu `usepackage[cp1250]{inputenc}`. Pro vytváření PDF potom `pdfLaTeX`. Při vytváření dokumentů do HTML a jiných formátů vhodných pro web se požívají nástroje `TeX4ht`, `LaTeX2HTML` nebo `tex2page`. Překlad pomocí `TeX4ht` a `pdfTeX` probíhá ve dvou fázích. Nejdříve je klasickým TeXem vygenerován výstup do DVI souboru, ovšem se speciálními poznámkami pro další zpracování. Na rozdíl od jiných nástrojů, které se snaží nějakým způsobem provádět expanzi TeXových příkazů vlastními prostředky, tak máme zaručeno, že expanze všech maker proběhne korektně.

Pro práci se systémem je třeba ještě prohlížeč souborů DVI, programy pro převod formátu DVI do formátu PS nebo PDF a další podpůrné utility. Ačkoliv zdrojový text je pouze ASCII soubor, který může být připraven v libovolném editoru, aby bylo možné systém efektivně používat, je vhodné mít k dispozici kvalitní editor pro tvorbu zdrojových souborů.

Co je TeX (LaTeX):

- Sázeací autorský systém, nejvhodnější pro sazbu matematických a technicky zaměřených knih, zpráv, protokolů...
- Programovatelný, t.j. s vlastním makrojazykem
- Volně šířitelný, otevřený, rozšiřitelný.
- Stabilní
- Dobře dokumentovaný
- S výstupem nezávislým na výstupním zařízení
- Optimalizovaný, vysoce efektivní, využívající nejrychlejší algoritmy své doby

K práci s LaTeX je potřeba:

- editor
- kompilátor(y)
- balíčky maker (grafika, barvy, hypertext, vícesloupcová sazba. . .)
- prohlížeče různých formátů výstupu (DVI, PostScript, PDF)
- konvertory formátů výstupu

Pokud tedy potřebujeme generovat zadání ve formátu vhodného pro další zpracování pomocí systému LaTeX je nutné původní skripty opatřit odpovídajícími změnami. Především výpis vygenerovaných otázek se již neuskutečňuje na obrazovce monitoru, ale zápis probíhá do předem vytvořeného souboru. Formát takového souboru už v sobě nezahrnuje HTML značky, ale naopak je doplněn značkami, které popisují, jak bude výstupní dokument po zpracování LaTeXem vypadat. Pro identifikaci jednotlivých zadání a správných odpovědí budou názvy generovaných testů ve svém názvu obsahovat náhodné a unikátní identifikační číslo. Výstupní soubor je opatřen příponou *.tex a uložen na místo definované přednastavenou cestou. Tento výchozí soubor už můžeme pomocí odpovídajících příkazů zpracovat podle požadovaného formátu. Stačí jen použít odpovídající příkaz a dále určit cestu k požadovanému souboru. Pro výstupní formát DVI, který můžeme prohlížet například pomocí prohlížeče YAP, vypadá zápis do příkazové řádky následovně:

```
latex jméno_souboru.tex
```

Překlad dokumentu ve formátu LaTeX do HTML můžeme provést příkazem:

```
htlatex jméno_souboru.tex
```

Celý dokument pak bude převeden do jediného HTML souboru. Překlad do DVI souboru se speciálními příkazy TeX4ht i následné převedení tohoto souboru do výstupního formátu probíhá automaticky, a to včetně případného několikanásobného překladu dokumentu za účelem vyřešení křížových odkazů.

Překlad dokumentu ve formátu LaTeX do PDF potom provádíme příkazem:

```
pdflatex jmeno_souboru.tex
```

ZADÁNÍ TESTU (Číslo: bz5m3x60)

Test se skládá celkem 6 z otázek. Každá testová otázka je hodnocena 5 body.

Celkem tedy lze s testu získat 30 bodů.

1. Auto A se pohybuje rychlostí v . Auto B se pohybuje rychlostí $2v$. Kolikrát větší kinetickou energii má auto B než auto A?

- A) dvakrát větší
- B) osmkrát větší
- C) čtyřikrát větší

2. Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i za rohem?

- A) rychlost
- B) vlnová délka
- C) frekvence

3. Litř vody a litř lihu. Vysledný objem bude (za předpokladu, že ani kapka nepadne vedle?)

- A) přesně 2 litry
- B) o něco méně než 2 litry
- C) o něco více než 2 litry

4. Hmotnost protonu je přibližně stejná jako hmotnost?

- A) elektronu
- B) neutronu
- C) fotonu

5. Nukleonové číslo udává počet?

- A) nukleonů (= protonů a elektronů dohromady)
- B) nukleonů (= protonů, neutronů a elektronů dohromady)
- C) nukleonů (= protonů a neutronů dohromady)

6. Červené krevní krvinky obsahují červené krevní barvivo?

- A) hemoglobin
- B) globulín
- C) fibrinogen

Obr.23: Vygenerované zadání v PDF souboru pomocí systému LaTeX

ZADÁNÍ TESTU (Číslo: bz5m3x60)

Test se skládá celkem 6 z otázek. Každá testová otázka je hodnocena 5 body.

Celkem tedy lze s testu získat 30 bodů.

1. Auto A se pohybuje rychlostí v . Auto B se pohybuje rychlostí $2v$. Kolikrát větší kinetickou energii má auto B než auto A?
 - A) dvakrát větší
 - B) osmkrát větší
 - C) čtyřikrát větší
2. Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i za rohem?
 - A) rychlost
 - B) vlnová délka
 - C) frekvence
3. Litř vody a litř lihu. Výsledný objem bude (za předpokladu, že ani kapka nepadne vedle)?
 - A) přesně 2 litry
 - B) o něco méně než 2 litry
 - C) o něco více než 2 litry
4. Hmotnost protonu je přibližně stejná jako hmotnost?
 - A) elektronu
 - B) neutronu
 - C) fotonu
5. Nukleonové číslo udává počet?
 - A) nukleonů (= protonů a elektronů dohromady)
 - B) nukleonů (= protonů, neutronů a elektronů dohromady)
 - C) nukleonů (= protonů a neutronů dohromady)
6. Červené krevní krvinky obsahují červené krevní barvivo?
 - A) hemoglobin
 - B) globulín
 - C) fibrinogen

Obr.24: Vygenerované zadání v HTML pomocí systému LaTeX

5. Závěr

Ze získaných poznatků byla v MySQL vytvořena jednoduchá databáze a pomocí obslužného jazyka PHP a HTML odzkoušeny základní operace s databázemi. Výsledkem je databáze otázek, která obsahuje tabulku pro zadání otázky, možnosti odpovědí v případě testových otázek, označení její správné odpovědi, kategorie dané otázky a okruh otázek pro který je otázka určena. Pro snadnější manipulaci s obsahem databáze bylo vytvořeno uživatelské prostředí pro obsluhu databáze a generování otázek do podoby, která je přehledná pro obsluhujícího uživatele. V prostředí můžeme otázky přidávat, editovat, měnit nebo mazat. Jednotlivé procedury byly odzkoušeny i přes webové prostředí phpMyAdmin.

Hlavním cílem diplomové práce bylo využít obsahu databáze k náhodnému generování testu, písemky nebo zkoušky při výběru otázek z několika okruhů. K tomuto účelu byla databáze naplněna experimentálními otázkami, které byly zastoupeny podle okruhů a to v různém poměru pro otázky zadání i pro otázky testové. V případě testových otázek je požadováno, aby se po generaci zadání testu náhodně zobrazovaly i možnosti s odpověďmi a nezobrazovali se tak, jak jsou uloženy v databázi. Dále bylo nutné zajistit, aby každé generované zadání mělo svůj unikátní identifikační klíč a v případě testových otázek nebo zkoušky se současně zobrazovali správné odpovědi na testové otázky. Všech zmiňovaných požadavků bylo dosaženo pomocí skriptů, které náhodně zajišťují výběr otázek z okruhů, náhodně zobrazují možnosti odpovědí a současně jsou schopny zobrazovat správné výsledky k aktuálnímu zadání.

Dalším úkolem práce bylo stávající systém doplnit funkcí pro generování zadání i správných odpovědí ve formátu PDF a HTML pomocí systému LaTeX. Toho bylo dosaženo ukládáním generovaného zadání do souboru s odpovídající příponou, který je opatřen značkami pracující se systémem LaTeX. Požadované formáty PDF a HTML byly následně vytvořeny pomocí speciálních příkazů.

6. Seznam použité literatury

[1] PHP 5, MySQL, Apache – vytváříme webové aplikace. Praha: Computer Press, 2006. 813 stran. ISBN 80-251-073-7

[2] Petr Zajíc, Seriál o PHP [online]. 2004 - [cit. 25.května 2009]. Dostupné na WWW: <http://www.linuxsoft.cz/article.php?id_article=171>

[3] Pavel Kácha, Seriál o HTML [online]. 2004 - [cit. 25.května 2009]. Dostupné na WWW: <http://www.linuxsoft.cz/article.php?id_article=185>

[4] Petr Zajíc, Seriál o MySQL [online]. 2005 - [cit. 25.května 2009]. Dostupné na WWW: <http://www.linuxsoft.cz/article.php?id_article=731>

[5] ing. Jan Fiala, PSPad editor [online]. 2007 - [cit. 25.května 2009]. Dostupné na WWW: <http://www.stahuj.centrum.cz/vyvojove_nastroje/www/tvorba/ostatni_editory/pspad>

[6] Michal Růžička, LaTeX pro web [online]. 2009 - [cit. 25.května 2009]. Dostupné na WWW: <<http://www.fi.muni.cz/~xruzick7/latex-pro-web/>>

[7] Tobiáš Potoček, CSS [online]. 2009 - [cit. 25.května 2009]. Dostupné na WWW: <<http://webtvorba.howto.cz/serialy/uvod-do-css/>>

[8] Tobiáš Potoček, HTML [online]. 2009 - [cit. 25.května 2009]. Dostupné na WWW: <<http://webtvorba.howto.cz/rubriky/html/>>

[9] W3C. Converting from HTML. [online]. 1999 – [cit. 25.května 2009]. Dostupné na WWW: <<http://www.w3.org/Tools/html2things.html>>

[9] Petr Sysel, Pavel Rajmic. LaTeX. [online]. 2007 – [cit. 25.května 2009]. Dostupné na WWW: <<http://latex.feec.vutbr.cz/>>

7. Přílohy

Příloha A – Ukázka skriptů pro práci s databází:

Vložení a úprava testové otázky prostřednictvím dat získaných z formulářů:

```
<?php
// pomocí přepínače nastavíme požadovanou akci
switch ($_GET['action']) {
// výběr akce úprava otázky
    case "edit":

// načtení příslušných sloupců v tabulce a pro jeden řádek provedeny změny
    $sql = "UPDATE otazky SET
        otazka_zneni = '" . $_POST['otazka_zneni'] . "',
        odpoved_A = '" . $_POST['odpoved_A'] . "',
        odpoved_B = '" . $_POST['odpoved_B'] . "',
        odpoved_C = '" . $_POST['odpoved_C'] . "',
        odpoved = '" . $_POST['odpoved'] . "',
        otazka_okruh = '" . $_POST['otazka_okruh'] . "'
        WHERE otazka_id = '" . $_GET['otazka_id'] . "'";

        break;
// vybrána akce přidat novou otázku
    case "pridat":
        switch ($_GET['type']) {
            case "otazka":
// vytvoření nové otázky a její uložení do databáze
                $sql = "INSERT INTO otazky
                    (otazka_zneni,
                    odpoved_A,
                    odpoved_B,
                    odpoved_C,
                    odpoved,
                    otazka_kategorie,
                    otazka_okruh)
                    VALUES
                    ('" . $_POST['otazka_zneni'] . "',
                    '" . $_POST['odpoved_A'] . "',
                    '" . $_POST['odpoved_B'] . "',
                    '" . $_POST['odpoved_C'] . "',
                    '" . $_POST['odpoved'] . "',
                    '1',
                    '" . $_POST['otazka_okruh'] . "')";

                break;
            }
        break;
    }
    if (isset($sql) && !empty($sql)) {
        echo "<!--" . $sql . "-->";
        $result = mysql_query($sql)
            or die("Invalid query: " . mysql_error());
    }
?>
```

Výmaz testové otázky:

```
<?php
// odstranění řádku v tabulce podle zjištěného identifikačního čísla
$sql = "DELETE FROM otazky
    WHERE otazka_id = '" . $_GET['otazka_id'] . "'";
mysql_query($sql);
?>
```

Příloha B – Ukázka skriptů pro generování otázek:

Skript pro generování náhodného řetězce znaků:

```
function generujKod ($delka = 8)
{
    // vytvoření prázdné proměnné $kod
    $kod = "";

    // definice možných znaků
    $znaky = "0123456789bcdfghjkmnpqrstvwxyz";

    // nastavení počítadla
    $i = 0;

    // vložení náhodného znaku do proměnné $kod dokud není dosaženo
    požadovaného počtu znaků podle $delka
    while ($i < $delka) {

        // vyzvednutí náhodného znaku z proměnné $znaky
        $char = substr($znaky, mt_rand(0, strlen($znaky)-1), 1);

        // dokud není dosaženo požadované délky cyklus se neustále opakuje
        if (!strstr($kod, $char)) {
            $kod .= $char;
            $i++;
        }
    }
    // Dokončen cyklus a je navracena posloupnost požadované délky
    return $kod;
}
```

Skript pro generování zadání písemné práce:

```
//načtení otázek pro generování písemky
$zadani = mysql_query("SELECT * FROM otazky WHERE otazka_kategorie = 2 order by otazka_id")
or die(mysql_error());

// počet všech otázek
$pocet_zadani = mysql_num_rows($zadani);

// výběr otázek z daného okruhu
$jedna = mysql_query("SELECT * FROM otazky WHERE otazka_okruh = 1 && otazka_kategorie = 2
order by otazka_id")
or die(mysql_error());

//počet otázek pro daný okruh
$pocet_1 = mysql_num_rows($jedna);

// výpočet počtu generovaných otázek z daného okruhu
$pocet_generovanych1 = ceil($pocet_1 * ($_POST['pocet']) / $pocet_zadani);

//ošetření proti zacyklení funkce
If ($pocet_generovanych1 > $pocet_1)
{return false;}

// nulování počítadla otázek v okruhu
$pocet_dosud_vygenerovanych = 0;

// definice že proměnná je pole
$pole_cisel = Array();

// dokud se pocet vygenerovanych nerovna poctu, který je potřeba provede cyklus while
($pocet_dosud_vygenerovanych != $pocet_generovanych1)
{
    $generovane_cislo = Rand (0,$pocet_1-1); // generovani
    //pokud je pravda,ze generovane cislo leži v poli..

    if ( in_array($generovane_cislo,$pole_cisel) == true )

        {} // pak nedelej nic

    else // pokud neni v poli..
        {
```



```

// generovane cislo ulozi do jine promene
$cislo_otazky = $generovane_cislo;

// take ho ulozi do pole
$pole_cisel[$pocet_dosud_vygenerovanych] = $generovane_cislo;

// inkrementace pocitadla
$pocet_dosud_vygenerovanych++;

//uložení výsledku generace
$vysedek = MySQL_Result($jedna,$generovane_cislo,otazka_zneni);
$otazky_podrobnosti1 .=<<<EOD

Skript pro generování možností odpovědí u testových otázek:
// funkce pro zobrazování možností s odpovědí v náhodném pořadí
function generovani_cisel ($kolik_potrebuju,$max_hodnota,$min_hodnota)
{
    If ($kolik_potrebuju > ($max_hodnota - $min_hodnota +1))
        {return false;}
    $pocet_dosud_vygenerovanych = 0;
    $pole_cisel = Array();

    while ($pocet_dosud_vygenerovanych != $kolik_potrebuju)
        {
            $generovane_cislo = Rand ($min_hodnota,$max_hodnota);

//pokud generovane cislo nelezi v poli..
            if ( !in_array($generovane_cislo,$pole_cisel) )
                {
                    $cislo = $generovane_cislo;
                    $pole_cisel[$pocet_dosud_vygenerovanych] = $generovane_cislo;
                    $pocet_dosud_vygenerovanych
                }
        }
    return $pole_cisel;
}

// odpovědi k náhodně vybrané otázce
$vysedek_1 = MySQL_Result($dotaz,$generovane_cislo,odpoved_A);
$vysedek_2 = MySQL_Result($dotaz,$generovane_cislo,odpoved_B);
$vysedek_3 = MySQL_Result($dotaz,$generovane_cislo,odpoved_C);

//zadání vstupních parametrů funkce
$pole_moznosti = generovani_cisel (3,3,1);

// cyklus pro zajištění náhodného zobrazení odpovědí
for ($i=0; $i<4; $i++)
{
    switch ($i)
    {
        case 0: $zobrazeni1 = "vysedek_".$pole_moznosti[$i]; break;
        case 1: $zobrazeni2 = "vysedek_".$pole_moznosti[$i]; break;
        case 2: $zobrazeni3 = "vysedek_".$pole_moznosti[$i]; break;
    }
}

//zobrazení odpovědí
echo $$zobrazeni1."<br>";
echo $$zobrazeni2."<br>";
echo $$zobrazeni3."<br>";

Skript pro zobrazování správných odpovědí na testové otázky:
// přepínač, který vrací podle zvoleného case proměnou "$vysedek_[$i]
for ($i=0; $i<4; $i++)
{
    switch ($i)
    {
        case 0: $zobrazeni1 = "vysedek_".$pole_moznosti[$i]; break;
        case 1: $zobrazeni2 = "vysedek_".$pole_moznosti[$i]; break;
        case 2: $zobrazeni3 = "vysedek_".$pole_moznosti[$i]; break;
    }
}

// definice podmínek k zjištění správné odpovědi
if ($odpoved == "Odpoved A")

```

```

$odpoved = "$vysledek_1";

if ($odpoved == "Odpoved B")
$odpoved = "$vysledek_2";

if ($odpoved == "Odpoved C")
$odpoved = "$vysledek_3";

// následovně už se pouze provede výpis proměnné $odpoved na obrazovku

```

Příloha C – Vygenerovaný soubor *.tex pro systém LaTeX:

```

\documentclass[12pt,a4paper,]{article}
\usepackage[cp1250]{inputenc}
\title{ZADÁNÍ TESTU (Číslo: bz5m3x60)}
\author{ Test se skládá celkem 6 z otázek. Každá testová otázka je hodnocena 5 body.}
\date{Celkem tedy lze s testu získat 30 bodů.}
\begin{document}
\maketitle
1. Auto A se pohybuje rychlostí v. Auto B se pohybuje rychlostí 2v. Kolikrát větší kinetickou
energii má auto B než auto A?
\begin{itemize}
\item A) dvakrát větší
\item B) osmkrát větší
\item C) čtyřikrát větší
\end{itemize} 2. Která charakteristika zvuku způsobuje to, že hluboké tóny můžeme slyšet i
za rohem?
\begin{itemize}
\item A) rychlost
\item B) vlnová délka
\item C) frekvence
\end{itemize} 3. Litř vody a litř lihu. Vysledný objem bude (za předpokladu, že ani kapka
nepadne vedle?
\begin{itemize}
\item A) přesně 2 litry
\item B) o něco méně než 2 litry
\item C) o něco více než 2 litry
\end{itemize} 4. Hmotnost protonu je přibližně stejná jako hmotnost?
\begin{itemize}
\item A) elektronu
\item B) neutronu
\item C) fotonu
\end{itemize} 5. Nukleonové číslo udává počet?
\begin{itemize}
\item A) nukleonů (= protonů a elektronů dohromady)
\item B) nukleonů (= protonů, neutronů a elektronů dohromady)
\item C) nukleonů (= protonů a neutronů dohromady)
\end{itemize} 6. Červené krevní krvinky obsahují červené krevní barvivo?
\begin{itemize}
\item A) hemoglobin
\item B) globulín
\item C) fibrinogen
\end{itemize}
\end{document}

```