

**UNIVERZITA PARDUBICE
ÚSTAV ELEKTROTECHNIKY A
INFORMATIKY**

**WWW APLIKACE PRO PREZENTACI
NAMĚŘENÝCH SPEKTER
V LABORATOŘI PRO VÝZKUM
NANOMATERIÁLŮ**

BAKALÁŘSKÁ PRÁCE

AUTOR PRÁCE: Jakub Král

VEDOUCÍ PRÁCE: RNDr. David Žák, Ph.D.

2007

**UNIVERSITY OF PARDUBICE
INSTITUTE OF ELECTRICAL
ENGINEERING AND INFORMATICS**

**WWW APPLICATION FOR
PRESENTATION MEASURING
SPECTRUMS IN LABORATORY FOR
RESEARCH NANOMATERIALS**

BACHELOR WORK

AUTHOR: Jakub Král

SUPERVISOR: RNDr. David Žák, Ph.D.

2007



Vysokoškolský ústav: Ústav elektrotechniky a informatiky
Katedra/Ústav: Ústav elektrotechniky a informatiky
Akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Pro: Král Jakub

Studijní program: Informační technologie

Studijní obor: Informační technologie

Název tématu: WWW aplikace pro prezentaci naměřených spekter v laboratoři pro výzkum nanomateriálů

Zásady pro zpracování:

Teoretická část se bude zabývat problematikou zobrazení naměřených dat (spekter, fotografií, vypočtených hodnot) na WWW stránkách, tvorbou a exportem grafů/obrázků (na základě naměřených hodnot) do protokolu o měření či pro prezentaci výsledků, konverzí grafických formátů.

Implementační část obsahuje návrh a tvorbu internetové aplikace s využitím relačních databází MySQL, v nichž jsou naměřená data určena ke zpracování, zobrazení a exportu uložena.

Aplikace musí umožnit vygenerovat z dat uložených v databázi:

- grafy a obrázky na základě naměřených hodnot,
- protokoly o měření ve formě rtf nebo pdf dokumentu obsahujícího texty, tabulky, grafy/obrázky,

- protokoly o měření ve formě html dokumentu,
- souhrnnou zprávu o výsledcích projektu ve formě rtf nebo pdf dokumentu,

- výstupní sestavy pro přehledy realizovaných projektů a měření ve formě rtf nebo pdf dokumentu,
- export dat do csv dokumentů,
- uložení grafu/obrázku spekter.

Student v práci navrhne vhodný způsob pro vytváření šablon rtf/pdf dokumentu.

Seznam odborné literatury:

Castagnetto, J. a kol. *Programujeme PHP profesionálně*. Computer Press, 2004.

Kout, P. *Praktický JavaScript*. Zoner Press, 2004.

Riordan, R.M. *Vytváříme relační databázové aplikace*. Computer Press, 2001.

Ullman, L. *PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek*. Computer Press, 2004

Rozsah: Přibližně 40 stran

Vedoucí práce: RNDr. David Žák, Ph.D.

Vedoucí katedry (ústavu): prof. Ing. Pavel Bezoušek, CSc.

Datum zadání práce: 30. 11. 2006

Termín odevzdání práce: 18. 5. 2007

PROHLAŠUJI:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 1. 5. 2007

Jakub Král
(vlastnoruční podpis)

ABSTRAKT

Tato práce se zabývá problematikou zobrazování výsledků naměřených spekter v laboratoři pro výzkum nanomateriálu na WWW stránkách, tvorbou a exportem grafů a obrázků na základě naměřených hodnot do protokolů o měření ve formátu PDF a HTML a pro exportem výsledků do CSV dokumentů. Práce vychází ze současného stavu poznatků, ovšem vzhledem k rychlému vývoji oblasti WWW aplikací lze očekávat další nové možnosti v tvorbě jednotlivých grafů a dokumentů PDF.

.

OBSAH

1. Úvod	12
2. Mössbauerovo spektrum	13
2.1. Rudolf Ludwig Mössbauer	13
2.2. Teoretické základy Mössbauerova jevu	13
2.2.1. Rezonanční absorpce	14
2.3. Mössbauerův spektrometr	15
3. Zobrazování naměřených dat	17
3.1. Technologie použité pro WWW aplikaci	17
3.1.1. XML.....	17
3.1.2. XHTML	19
3.1.3. CSS	21
3.1.4. PHP	22
3.1.5. MySQL	24
3.2. Tvorba obrázků pomocí knihovny GD	25
3.2.1. Připojení knihovny GD k distribuci PHP.....	25
3.2.2. Postup vytvoření obrázku.....	26
3.2.3. Funkční příklad užití GD.....	29
3.3. Tvorba grafů pomocí JpGraphu	32
3.3.1. Využití a instalace JpGraphu	32
3.3.2. Základní postup vytvoření grafu.....	33
3.4. Tvorba PDF dokumentů pomocí PHP	35
3.4.1. Knihovny PDFlib a FPDF.....	35

3.4.2.	Základní postup vytvoření PDF dokumentu	37
3.4.3.	Ukázka vytvoření PDF dokumentu	40
3.5.	Tvorba CSV dokumentů pomocí PHP.....	42
4.	Aplikace	44
4.1.	Požadavky	44
4.2.	Instalace PreSpe.....	45
4.2.1.	Konfigurace v internetovém prohlížeči	46
4.2.2.	Konfigurační soubor	47
4.3.	Databáze MySQL.....	48
4.4.	Zobrazení spekter	51
4.4.1.	Zobrazení spektra a subspekter v jednom grafu.....	51
4.4.2.	Jednotlivá spektra	55
4.5.	Protokoly o měření	56
4.5.1.	Protokoly o měření ve formátu PDF.....	56
4.6.	Export dat do CSV dokumentů	58
5.	Závěr.....	59

_Toc167794285

SEZNAM OBRÁZKŮ

Obrázek 1 - rozpadové schéma zářiče	15
Obrázek 2 - Mössbauerův spektrometr (nakreslen knihovnou GD)	15
Obrázek 3 - informace o nastavení GD pomocí funkce php_info()	26
Obrázek 4 - spojnicový graf vytvořený v příkladu	35
Obrázek 5 - konfigurace PreSpe v internetovém prohlížeči	46
Obrázek 6 - UML diagram databáze aplikace PreSpe	51
Obrázek 7 - formulář pro výběr zobrazovaných spekter	52
Obrázek 8 - Výsledný graf zobrazující navolená subspektra	54
Obrázek 9 - formulář pro změnu pořadí zobrazení subspekter	54
Obrázek 10 - export dat do CSV dokumentů	58

SEZNAM TABULEK

Tabulka 1 - základní funkce GD.....	27
Tabulka 2 - Přehled funkcí knihovny FPDF	38
Tabulka 3 - požadavky pro správný chod aplikace PreSpe.....	45
Tabulka 4 - Seznam tabulek databáze MySQL aplikace PreSpe	49
Tabulka 5 - PreSpe databáze - tabulka mereni.....	49
Tabulka 6 - PreSpe databáze - tabulka <i>data_x</i>	50
Tabulka 7 - PreSpe databáze - tabulka <i>spektra</i>	50
Tabulka 8 - PreSpe databáze - tabulka <i>data_y</i>	50

1. ÚVOD

Hlavní téma mé bakalářské práce se zabývá problematikou zobrazování naměřených spekter v laboratoři pro výzkum nanomateriálu na WWW stránkách. S tímto hlavním úkolem více či méně souvisí problematika tvorby a exportu již zmíněných dat do protokolů ve formátu PDF a HTML či dokumentů CSV.

Cílem práce je tedy WWW aplikace implementující výše zmíněné problémy. Data z výsledků měření bude aplikace získávat z databáze MySQL. Veškeré výstupy aplikace budou generovány v okamžik požadavku, což nám zaručí aktuálnost.

V této práci popíši veškeré postupy a metody, které sem pro tvorbu aplikace použil a také trochu nahlédnu do teorie. Postupně projdu alespoň okrajově všechny technologie, které jsem použil a více se zaměřím na problematiku tvorby obrázků pomocí knihovny GD, na tvorbu grafů pomocí souboru knihoven JpGraph a přiblížím tvorbu CSV a PDF dokumentů. V poslední části pak ukážu, jak jsem jednotlivé problémy implementoval do výsledné aplikace.

2. MÖSSBAUEROVO SPEKTRUM

Výsledná aplikace je navržena a otestována pro konkrétní typ naměřeného spektra. Jedná se o Mössbauerovo spektrum. V této kapitole tedy přiblížím problematiku Mössbauerova jevu a představím samotného vynálezce Rudolfa Ludwiga Mössbauera.

Texty kapitol 2.2, 2.2.1 a 2.3 jsou převzaty z části disertační práce RNDr. Davida Žáka, Ph.D. (1).

2.1. Rudolf Ludwig Mössbauer

Mössbauerův jev objevil v roce 1957 německý fyzik Rudolf Ludwig Mössbauer (narozený roku 1929). Za tento objev získal v roce 1961 Nobelovu cenu, o kterou se dělí společně s Robertem Hofstadterem (ten ji získal za výzkum struktury protonu a neutronu).

V roce 1958, tedy rok po objevení jevu, získal PhD na univerzitě v Mnichově. Později působil jako profesor fyziky. Od roku 1961 přednášel fyziku v Pasadeně na CalTechu (California Institute of Technology) a od roku 1964 ten samý předmět v Mnichově na Akademii techniky (Technical Academy).

2.2. Teoretické základy Mössbauerova jevu

Mössbauerův efekt je jev bezdrazové emise a rezonanční absorpce γ -záření určitého atomového jádra v pevných látkách. Mössbauerova

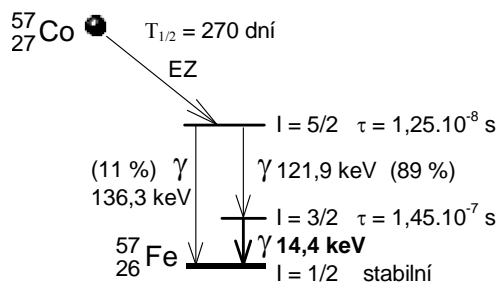
spektroskopie umožňuje studovat malé změny lokální elektronové konfigurace a elektrického a magnetického pole v pevných látkách. Z Mössbauerových spekter je možné rozlišit různé oxidační stavy, spinové stavy a strukturu prostředí.

Mössbauerovský přechod může být pozorován na přibližně 109 jaderných přechodech 70 izotopů 40 prvků. Největší zájem se soustředí na izotopy $^{57}_{26}\text{Fe}$, $^{73}_{32}\text{Ge}$, $^{119}_{50}\text{Sn}$, $^{121}_{51}\text{Sb}$, $^{125}_{52}\text{Te}$, $^{127}_{53}\text{I}$, $^{151}_{63}\text{Eu}$, $^{182}_{74}\text{W}$, $^{190}_{76}\text{Os}$, $^{191}_{77}\text{Ir}$ a $^{237}_{93}\text{Np}$.

2.2.1. Rezonanční absorpce

V Mössbauerově spektroskopii je energie fotonů záření γ emitovaných radioaktivním zářičem modulována pomocí Dopplerova jevu. Foton záření γ s energií odpovídající jadernému přechodu v jádře absorberu je tímto jádrem s určitou pravděpodobností rezonančně absorbován. Pro správné pochopení Mössbauerovy spektroskopie je nutné rozumět radioaktivnímu rozpadu, Dopplerově jevu a jaderným rezonančním jevům. V následujících řádcích se stručně seznámíme s klasickým výkladem Mössbauerova jevu (např. Frauenfelder, 1962) s ohledem na jádra $^{57}_{26}\text{Fe}$. Přesný výklad Mössbauerova jevu je možné provést na základě kvantové mechaniky (např. Lipkin, 1973).

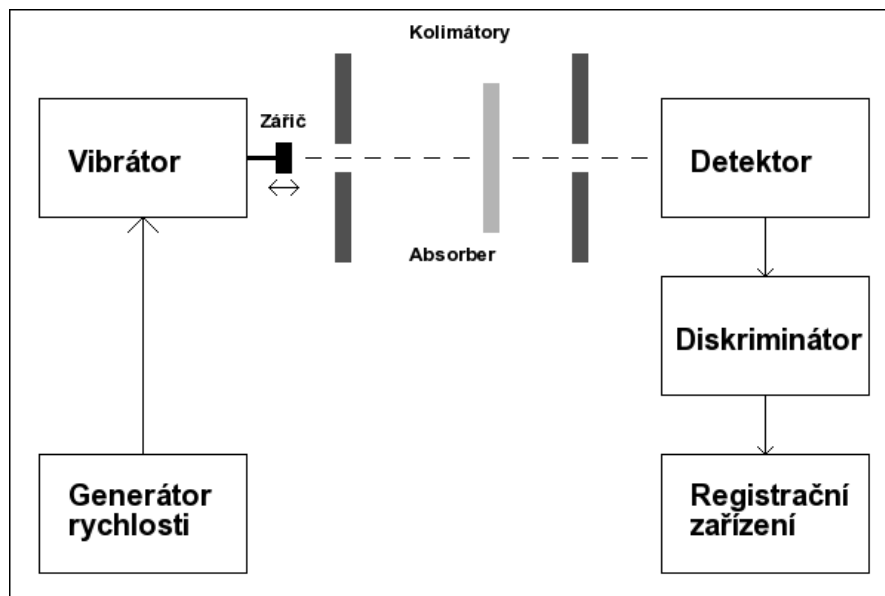
Pro Mössbauerovu spektroskopii na jádrech $^{57}_{26}\text{Fe}$ se používá zářič $^{57}_{27}\text{Co}$ (Irkajev a kol., 1970), který se záchytem elektronu rozpadá na $^{57}_{26}\text{Fe}$ ve druhém excitovaném stavu (Obrázek 1). $^{57}_{27}\text{Co}$ se získává v cyklotronu bombardováním jader $^{56}_{26}\text{Fe}$ deuterony.



Obrázek 1 - rozpadové schéma zářiče

2.3. Mössbauerův spektrometr

Obecné uspořádání Mössbauerova spektrometru ukazuje Obrázek 2. Zářič je pevně přichycen k pohyblivé části vibrátoru, jejíž pohyb je řízen generátorem rychlosti. Kolimovaný svazek γ -záření prochází absorberem a prošlé záření je registrováno detektorem. Pomocí diskriminátoru se provádí amplitudová selekce impulsů z výstupu detektoru. Impulsy z diskriminátoru jsou zaznamenány registračním zařízením do jednotlivých registračních kanálů odpovídajících různým rychlostem relativního pohybu zářiče vůči absorberu.



Obrázek 2 - Mössbauerův spektrometr (nakreslen knihovnou GD)

Energie záření dopadajícího na absorber je modulována na základě Dopplerova jevu a je rovna $E_{\gamma}^{\text{mod}} = E_{\gamma} \left(1 + \frac{v}{c} \cos \alpha \right)$, kde v je relativní rychlost pohybu zářiče vůči absorberu, α je úhel mezi vektorem rychlosti \vec{v} a směrem emise γ -záření. Výsledkem měření je závislost intenzity γ -záření registrovaného detektorem na relativní rychlosti pohybu v zvaná Mössbauerovo spektrum, v němž sledujeme křivky rezonanční absorpce, jejich amplitudu, polohu, šířku a štěpení způsobené kvadrupólovými nebo magnetickými interakcemi v absorberu či zářiči.

Výsledky měření se prezentují graficky ve formě $N=f(v)$, nebo $\varepsilon=f(v)$, kde N je počet registrovaných částic, ε efekt, dále se provede matematické zpracování naměřených hodnot, stanoví se parametry jednotlivých píků (rezonanční rychlost a šířka čáry) a závěrečnou a nejnáročnější fází je fyzikální rozbor naměřeného spektra a stanovení parametrů hyperjemných interakcí složek spektra.

Základními parametry charakterizujícími Mössbauerův spektrometr jsou: interval rychlosti, stabilita rychlosti, linearita rychlostní osy, kapacita registračního kanálu, počet registračních kanálů, mrtvá doba detektoru a jeho registrační účinnost.

3. ZOBRAZOVÁNÍ NAMĚŘENÝCH DAT

A nyní se již dostávám k hlavní problematice práce. Tato kapitola bude pojednávat o možnostech tvorby obrázků pomocí knihovny GD, tvorbě grafů s využitím souboru knihoven JpGraph a také generováním dokumentů PDF, které se v aplikaci využívají k tvorbě protokolů. Nezapomenu ani na problém exportu dat do CSV dokumentů.

3.1. Technologie použité pro WWW aplikaci

Nejprve však popíši veškeré technologie, které jsem při tvorbě WWW aplikace použil, ale které nejsou hlavní náplní této práce. Bez nich by však aplikace nemohla vzniknout. Jedná se o XML, XHTML, CSS, PHP a MySQL. Přiblížím jejich historické pozadí a základní vlastnosti.

3.1.1. XML

Ve své práci jsem využíval technologie XHTML. Abych mohl popsat jeho vývoj a jednotlivé vlastnosti a pravidla pro zápis, je nutné se nejprve seznámit s XML¹, ze kterého byl jazyk XHTML, ať už v jakékoliv verzi, odvozen.

XML je tedy značkovací jazyk, který si získává stále větší oblibu. Jeho předchůdcem byl, stejně jako pro jazyk HTML, jeden z prvních značkovacích jazyků vůbec – SGML². A právě z něho a z jazyka HTML vychází. Snahou bylo

¹ XML (eXtensible Markup Language) – rozšiřitelný značkovací jazyk

² Standard Generalized Markup Language – univerzální značkovací metajazyk

vzít si z těchto dvou jazyků jen to dobré a poučit se z chyb. Základním stavebním kamenem pro tento jazyk byla jednoduchost. Jak SGML, tak později HTML, byl jazyk nesmírně obsáhlý se spoustou tagů a tím pádem i složitý.

Jedno tedy bylo jasné – co nejjednodušší jazyk s minimem značek. Zároveň ale byla snaha o neomezený počet tagů pro přesnější označení konkrétních informací v dokumentu. Na první pohled nevyřešitelný problém. Konsorcium W3C však tento problém vyřešilo elegantně – vzniká jazyk XML, který neobsahuje žádné předem určené tagy (předem, tedy následně i neomezeně).

Při vývoji tohoto jazyka si konsorcium dalo několik cílů pro nový formát, mezi ty nejvýznamnější lze zařadit např. přímé použití v internetu, podpora širokého spektra aplikací (správu dokumentů, publikačního materiálu, databázových informací...), kompatibilita s SGML, snadná tvorba programů pro zpracování XML dokumentů, čitelnost a přehlednost dokumentů, rychlé dokončení návrhu³, formálnost a stručnost návrhu a hlavně striktní dodržování psaní značek v kompletní formě. V konečném důsledku to vypadá tak, že data dokumentu jsou oddělena od grafického zpracování. S velkou výhodou toho lze využívat pro zobrazení jednoho souboru dat do více podob. Nemusíte pracně přepisovat kód, stačí připojit jiný styl. A dokonce to funguje i naopak. Připojením jednoho stylu k libovolnému počtu dat dostáváme jeden grafický vzhled výstupu.

³ Vývoj jazyka začal v létě 1996, první verze byla vydána v únoru 1998

Po historickém vývoji jazyka XML a jeho cílech se ještě ve stručnosti zmíním o syntaktických pravidlech jazyka XML. Již výše bylo zmíněno striktní dodržování pravidel zápisu tagů. Další podmínky pro XML dokument jsou:

- XML dokument by měl začínat hlavičkou s verzí a kódováním
- XML dokument musí obsahovat alespoň jeden element a pouze jeden root⁴ dokument
- Názvy elementů musí odpovídat pravidlům
- Elementy se nesmí křížit
- XML dokument je case-sensitive
- XML dokument nesmí obsahovat zakázané znaky, nahrazují se entitami

3.1.2. XHTML

A nyní se již dostávám k XHTML. XHTML je zkratka z anglického *extensible hypertext markup language*, což v překladu znamená rozšiřitelný značkovací jazyk pro hypertext. Jedná se o značkovací jazyk vytvořený, nebo lépe řečeno vyvinutý, konsorciem W3C. Předchůdcem mu byl značkovací jazyk HTML, který ovšem již ukončil svůj vývoj.

XHTML je oproti HTML pomalejší, ale bezpečnější a navíc se více přizpůsobuje dnešní době. Nyní je ve vývoji verze XHTML 2.0, která má svůj boj o prosazení teprve před sebou.

První verze XHTML nebyla ničím jiným, než pouze přepsání poslední verze HTML 4.01 do pravidel pro XML. Tato verze existuje ve třech různých

⁴ Root – hlavní element, který obsahuje všechny ostatní (vyjma hlavičky)

variantách, a to *Transitional*, která je k programátorovi nejvíce tolerantní, *strict*, která se naopak velmi striktně drží pravidel a odmítá i sebemenší chybičku ve zdrojovém kódu a za chybné považuje i značky, které nedoporučuje používat, a poslední verzí je *frameset*, která je odvozena od Transitional.

Následující verze XHTML 1.1 se odvíjela od verze XHTML 1.0 strict. Navíc byla rozdělena do několika modulů a nabídla tedy mnohem větší škálu působnosti, např. tvorbu XHTML pro mobilní zařízení, tiskárny apod. Také zde existují různé varianty této verze, kromě běžné XHTML 1.1 ještě XHTML Basic, který odlehčuje základní verzi o několik modulů a také XHTML 1.1 s MathML 2.0, která navíc obsahuje jazyk MathML.

Jelikož jsem pospal základní pravidla pro XML, nyní to učiním i zde. Následuje výpis nejzákladnějších pravidel pro XHTML:

- XHTML požaduje vždy uvádět DOCTYPE
- značky musí být zapsány podle pravidel pro XML (viz výše)
- oproti HTML musí být tagy vždy uzavřeny (i nepárové)
- tagy musí být psány malými písmeny
- veškeré hodnoty parametrů musí být uzavřeny v uvozovkách a musí mít hodnotu

Kontrola. Jedna z nejdůležitějších (a nejobtížnějších) věcí na programování. Pokud vše naprogramujeme správně a aplikace (nejen pro www stránky) funguje bez problémů, máme vyhráno. Ale dovoluji si tvrdit, že těžší, než něco naprogramovat, je najít ve zdrojovém kódu po sobě, natož pak po někom jiném, chybu. Pro www stránky a konkrétně pro XHTML naštěstí konsorcium W3C vyvinulo validátor.

Sice existuje mnoho jiných validátorů, ať už off-line či on-line, ale žádný asi nebude tak přesný jako od samotného konsorcia W3C. Dostupný je na oficiálních stránkách W3C na adrese <http://validator.w3.org>.

Cest pro validaci je několik. Pokud jsou stránky dostupné z webu, lze zadat URL. Další možností je nahrát soubor z pevného disku, nebo zkopírovat zdrojový kód stránky a vložit do speciální formuláře validátoru.

Ve zprávě o validaci stránky se lze dozvědět vše potřebné od názvu souboru, který byl validován, přes kódování, doctype až samozřejmě po výsledek. Tedy zda je soubor validní či nikoliv. Při úspěšném pokusu validátor nabídne adresu ikony validní stránky, v opačném případě validátor vypíše veškeré chyby a při použití verze *Transitional* dokonce i varování pro chyby, které by byly verzí *Strict* odmítnuty.

3.1.3. CSS

Jelikož XHTML neobsahuje skoro žádné možnosti pro vzhled a formátování⁵, musel proto vzniknout jazyk pro popis způsobu zobrazování stránek napsaný tímto jazykem. A to je hlavním úkolem CSS⁶.

Jazyk CSS však vznikl dříve než jazyk XML a XHTML. Záměr však byl stejný. Pokud možno co nejvíce oddělit vzhled stránky od jejího obsahu. Programátorům se totiž nelíbily nejen omezené možnosti formátování, ale také zdlouhavé a nepřehledné formátování textu pomocí značky ``. Např. v tabulce bylo nutné nastavit písmo pro každou buňku.

⁵ Vzpomeňme na XML – ten neumožňuje již žádné

⁶ CSS- Cascading StyleSheets česky znamená tabulky kaskádových stylů, používá se však zkrácený název kaskádové styly

Opět zde nebudu rozbírat jednotlivé formy zápisu značek CSS a jejich významy, doplním jen, že stejně jako existuje validátor pro XHTML, existuje i validátor pro CSS, který je opět dostupný na oficiálních stránkách W3C, konkrétně na adrese <http://jigsaw.w3.org/css-validator/>. Možnosti zadávání jsou stejné jako u XHTML, tedy pomocí URL, souboru, nebo přímo zdrojového kódu.

3.1.4. PHP

Veškeré doposud popsané technologie (myšleno XML a XHTML) nedovolují vytvářet dynamické obsahy www stránek, ale pouze statické. A právě PHP⁷ je skriptovací jazyk, který je využíván především pro tvorbu dynamických webových stránek. Podívám se trochu na vývoj tohoto jazyka.

Autor původního nástroje PHP Rasmus Lerdorf chtěl vytvořit jazyk, který by zpracovával data z formulářů pro svou vlastní potřebu. Bylo to začátkem roku 1994 a ještě v tomto roce své skripty vydal. Následně skripty spojil se svým jiným programem a vznikl název PHP/FI⁸. Byl to však jen základ základu dnešního jazyka PHP, kterému chyběla řada funkcí.

V roce 1997 si projektu všimli Andi Gutmans a Zeev Suraski, kteří pracovali na univerzitním projektu ohledně elektronického obchodování. Vzhledem k nedostatkům se rozhodli kompletně jazyk přepracovat. Společně s Lerdorfem připravili nový jazyk označený jako PHP 3. Oproti předešlým verzím obsahoval mnohem více funkcí a byl rychlejší.

⁷ PHP – původně Personal Homepage, od verze PHP3 se nazývá Hypertext Preprocessor, neboli hypertextový preprocesor

⁸ PHP/FI – FI bylo přidáno společně s programem s názvem Form Interpreter

Přesto Andi Gutman se Zeevem Suraskim nebyli s výsledkem ještě spokojeni. Od roku 1998, tedy velmi krátce po vydání PHP 3, pracovali na novém jádru. Cílem bylo rychlé a jinak koncipované PHP. Zatímco předešlé verze zpracovávali skripty během čtení, nové PHP mělo skript zpracovávat až po načtení, respektive překladu. Na zpracování vytvořili Andi a Zeev nový stroj, který nazvali Zend⁹. Po čtyřech letech práce vychází v květnu roku 2000 nová verze PHP 4. Oproti předešlým verzím existuje v PHP 4 více verzí, které postupně reagovali na chyby a problémy verzí předešlých. Hlavní novinkou v PHP 4 bylo zavedení globálních a superglobálních proměnných. Právě zejména nastavení těchto proměnných bylo příčinou vydávání nových verzí. Důležitým milníkem pro PHP bylo vydání verze PHP 4.3.0 v prosinci roku 2002. Poprvé obsahovala rozhraní příkazového řádku, nově byly zavedeny proudy¹⁰ a také grafická knihovna GD, která by se s trochou nadsázky dala označit za základní stavební kámen mé bakalářské práce. Ještě nyní, když existuje PHP 5, je verze PHP 4 upravována a aktualizována a její zatím poslední verze¹¹ byla vydána 14. února 2007 s označením PHP 4.4.5.

A dostávám se k zatím poslední verzi PHP. Je to PHP 5. Základní změna – obecný, objektově orientovaný přístup. Nutností bylo přepsání části skriptovacího stroje Zend a tedy i nové označení Zend 2.0. Oproti předchozím verzím také obsahuje mnoho nových funkcí. První verze PHP 5 byla vydána po dlouhém vývoji a mnoha předverzích v červenci 2004.

Ted' se chvíli zastavím nad změnami v PHP 5. Hlavní novinky se samozřejmě týkají objektově orientovaných funkcí. Poprvé byly zavedeny modifikátory *public*, *private* a *protected* pro označení metod a vlastností

⁹ Zend je zkratka křestních jmen obou autorů, tedy **Z**eev a **A**ndi

¹⁰ Proudý, neboli Streams, vylepšovali vrstvu souborového a síťového rozhraní

¹¹ Stav k 22. dubnu 2007

jako veřejné, respektive soukromé a chráněné. Změna nastala i u konstruktorů a destruktorů. Deklarují se pomocí speciálních unifikovaných názvů, nejsou tedy shodné s názvem třídy. Následuje přehled dalších významných funkcí:

- rozhraní pro třídy
- operátor *instanceof*
- konečné metody a třídy
- explicitní klonování objektů
- konstanty tříd, statické metody a členy
- abstraktní třídy a metody
- iterátory
- funkce *_autoload()*

3.1.5. MySQL

Stěžejní technologií výsledné aplikace je multiplatformní databázový systém MySQL, kterou vytvořila švédská firma MySQL AB¹². Databáze MySQL existuje ve dvou verzích, v bezplatné a komerční.

Hlavní cíl při vytváření databáze MySQL byl kladen na rychlost. A to i za cenu některých nedostatků. Zejména programátorům WWW stránek začaly chybět funkce, které obsahovaly ostatní komerční verze. Postupem času tak byla (a stále je) databáze MySQL vylepšována. První verze uvolněná v roce 1995 obsahovala opravdu jen zárodky dnešní podoby. Větší průlom nastal od verze 3.23, která poprvé podporovala cizí klíče¹³ a transakce¹⁴. Verze 4.1

¹² Hlavními autory jsou Michael „Monty“ Widenius a David Axmark

¹³ Cizí klíč – hodnota tabulky odkazující na jinou hodnotu

¹⁴ Transakce – skupina příkazů převádějící databázi z jednoho konzistentního stavu do druhého

už obsahovala podporu pro různé znakové sady a časová pásma v datech a umožňovala vytváření vnořených dotazů. Poslední velký průlom nastal ve verzi 5.0. Větší komfort programátorovi poskytuje možnost ukládání procedur, vytváření triggerů¹⁵ a pohledů¹⁶. Aktuální verze v době psaní práce je 5.0.27 pro bezplatnou verzi a 5.0.37 pro komerční verzi.

Databáze MySQL je v dnešní době velice rozšířená. Hlavním důvodem obliby je samozřejmě možnost využívání bezplatné verze, dále pro svoji rychlost, výkon a neustálý vývoj.

3.2.Tvorba obrázků pomocí knihovny GD

Veškeré grafy, které lze v aplikaci vidět, jsou vytvářeny prostřednictvím souborů knihoven JpGraphu. Grafy však knihovny vytváří pomocí nadstavby PHP – grafické knihovně GD. V aplikační části práce tedy používání této knihovny není k vidění. Nicméně aplikace přilinkování této knihovny vyžaduje a tak si přiblížím i základní principy tvorby obrázků pomocí této grafické knihovny.

3.2.1. Připojení knihovny GD k distribuci PHP

Knihovna GD je v systémech Windows dodávána společně s distribucí PHP. Jediná nutnost je nastavit, nebo lépe řečeno povolit nebo dopsat, v konfiguračním souboru PHP `php.ini` řádek `extension=php_gd2.dll`. V unixu se pak musí distribuce přeložit s přepínačem `--with-gd`. Jestli se vše povedlo lze zjistit pomocí funkce `php_info()`, kde se dozvíme vše podstatné o této knihovně (Obrázek 3).

¹⁵ SQL trigger – činnost, která se provádí v případě volání definované události

¹⁶ SQL pohled –poskytuje uživateli data ve stejné podobě jako tabulka (může se však skládat z více tabulek)

gd

GD Support	enabled
GD Version	bundled (2.0.28 compatible)
FreeType Support	enabled
FreeType Linkage	with freetype
FreeType Version	2.1.9
T1Lib Support	enabled
GIF Read Support	enabled
GIF Create Support	enabled
JPG Support	enabled
PNG Support	enabled
WBMP Support	enabled
XBM Support	enabled

Obrázek 3 - informace o nastavení GD pomocí funkce `php_info()`

3.2.2. Postup vytvoření obrázku

Po úspěšném přilinkování grafické knihovny se můžu rovnou vrhnout na tvorbu obrázku. Detailně popíši základní funkce nutné pro vytvoření obrázku a na závěr uvedu souhrn nejpoužívanějších funkcí.

Abych mohl kreslit jakékoliv objekty do obrázku, musím nejprve vytvořit nové kreslicí plátno. Toho lze docílit pomocí funkce `imagecreate(x,y)`, která vytvoří kreslicí plátno o rozměrech x a y do kterého lze kreslit pomocí 256 odstínů barev. Alternativou je funkce `imagecreatetruecolor(x,y)`, která se od předchozí liší v rozsahu barev. Místo 256 odstínů lze kreslit v rozlišení *true color*. Implicitně obě funkce nastavují černou barvu pozadí.

Jednotlivým prvkům kresby lze nastavit barvu pomocí funkce `imagecolorallocate(image,R,G,B)`, kde argumentem `image` je obrázek vytvořený pomocí předchozí funkcí `imagecreate(x,y)`, respektive `imagecreatetruecolor(x,y)`. Další tři argumenty obsahují informace o

barevných složkách vyjádřených celým číslem v rozsahu 0-255. Barevné složky se uvádějí v pořadí červená (R), zelená (G) a modrá (B).

Získanou barvu lze aplikovat na veškeré prvky kresby. Například nakreslení a vyplnění obdélníku docílíme pomocí funkce `imagefilledrectangle(image,x1,y1,x2,y2,color)`. První argument `image` je opět obrázek vytvořený pomocí funkce `imagecreate(x,y)`, resp. `imagecreatetruecolor(x,y)`. Argumenty `x1` a `y1` se rozumí souřadnice levého horního rohu a argumenty `x2` a `y2` souřadnice pravého dolního rohu. Takovýmto způsobem se tedy barva aplikuje.

Po nakreslení celého obrázku se musí zadat funkce `header()` a předat informace o MIME typu - *Content-Type: image/png* (png lze zaměnit za jiný podporovaný typ obrázku). Posledním krokem vytvoření obrázku je zavolání funkce grafické knihovny `imagepng($image)`, která obrázek zobrazí v prohlížeči. Parametrem této funkce je již známý obrázek. Nyní je však nutné dát pozor. Před zavoláním této funkce nesmí být žádný kód, který odesílá jakákoliv data (stačí například obyčejné *echo*) na server. To by způsobilo chybu a obrázek by nebyl zobrazen.

A zde je již slíbený přehled nejpoužívanějších funkcí a jejich stručný popis. Detailní informace lze nalézt na oficiálních stránkách rozšíření PHP o GD - <http://cz2.php.net/manual/cs/ref.image.php>.

Tabulka 1 - základní funkce GD

Funkce	Popis
<code>Imagearc</code> <code>(im,cx,cy,w,h,sa,ea,c)</code>	Nakreslení oblouku do obrázku <i>im</i> , ze středu <code>[cx,cy]</code> , o šířce <i>w</i> a výšce <i>h</i> , s počátečním úhlem <i>sa</i> a koncovým úhlem <i>ea</i> , barvou <i>c</i>
<code>Imagefilledarc</code> <code>(im,cx,cy,w,h,sa,ea,c,s)</code>	Stejná funkce jako předchozí, navíc vyplněná stylem <i>s</i>
<code>Imagechar</code> <code>(im,font,x,y,ch,c)</code>	Nakreslí znak <i>ch</i> do obrázku <i>im</i> , od souřadnic <code>[x,y]</code> , písmem <i>font</i> a barvě <i>c</i>

Imagecharup (<i>im,font,x,y,ch,c</i>)	Stejná funkce jako předchozí, jen znak vykreslí vertikálně
Imagecolorallocate (<i>im,r,g,b</i>)	Vytvoří barvu pro obrázek <i>im</i> , se složkami červené <i>r</i> , zelené <i>g</i> a modré <i>b</i>
Imagecolorat (<i>im,x,y</i>)	Načte informaci o barvě z pozice <i>x,y</i>
Imagecolordeallocate (<i>im,c</i>)	Dealokuje barvu <i>c</i> z obrázku <i>im</i>
Imagecopy (<i>dim,sim,dx,dy,sx,sy,sw,sh</i>)	Kopíruje část obrázku <i>sim(z)</i> (z pozice [<i>sx,sy</i>]) o šířce <i>sw</i> a výšce <i>sh</i> do obrázku <i>dim</i> (do pozice [<i>dx,dy</i>])
Imagecopyresampled (<i>dim,sim,dx,dy,sx,sy,dx,dy,sw,sh</i>)	Stejná funkce jako předchozí, navíc mění rozměry zkopírované části
Imagecreate(w,h)	Vytvoření nového plátna o šířce <i>w</i> a výšce <i>h</i> a 256 barvách
Imagecreatetruecolor(w,h)	Vytvoření nového plátna o šířce <i>w</i> a výšce <i>h</i> a true color
Imagecreatefromgif(fn)	Vytvoření nového obrázku ze souboru nebo URL adresy, obrázek musí být formátu gif
Imagecreatefromjpeg(fn)	Stejná funkce, jen z obrázku jpeg
Imagecreatefrompng(fn)	Stejná funkce, jen z obrázku png
Imagecreatefromwbmp(fn)	Stejná funkce, jen z obrázku bmp
Imagedashedline (<i>im,sx,sy,ex,ey,c</i>)	Nakreslí čerchovanou čáru do obrázku <i>im</i> od bodu [<i>sx;sy</i>] do bodu [<i>ex;ey</i>] barvou <i>c</i>
Imagedestroy(im)	Zruší obrázek <i>im</i>
Imageellipse (<i>im,cx,cy,w,h,c</i>)	Nakreslí elipsu do obrázku <i>im</i> ze středu [<i>cx,cy</i>] o šířce <i>w</i> , výšce <i>h</i> a barvě <i>c</i>
Imagefilledellipse (<i>im,cx,cy,w,h,c</i>)	Stejná funkce jako předchozí, navíc vyplní obsah
Imagefill(im,x,y,c)	Záplavové vyplnění obrázku <i>im</i> ze souřadnic [<i>x;y</i>] a barvě <i>c</i>
Imagepolygon(im,p,n,c)	Nakreslí polygon určený body <i>p</i> (celkový počet bodů <i>n</i>) a barvě <i>c</i>
Imagefilledpolygon (<i>im,p,n,c</i>)	Stejná funkce jako předchozí, navíc vyplní obsah
Imagerectangle (<i>im,x1,y1,x2,y2,c</i>)	Nakreslí obdélník z levého horního bodu [<i>x1;y1</i>] do pr. spodního bodu [<i>x2;y2</i>] barvou <i>c</i>
Imagefilledrectangle (<i>im,x1,y1,x2,y2,c</i>)	Stejná funkce jako předchozí, navíc vyplní obsah
Imagefilltoborder (<i>im,x,y,b,c</i>)	Záplavové vyplňování obrázku <i>im</i> barvou <i>c</i> od bodu [<i>x;y</i>] do hranice určené barvou <i>b</i>
Imagegif(im[,fn])	Odeslání obrázku <i>im</i> gif do prohlížeče, nebo v případě vyplnění parametru <i>fn</i> do souboru
Imagejpeg(im[,fn])	Odeslání obrázku <i>im</i> jpeg do prohlížeče, nebo v případě vyplnění parametru <i>fn</i> do souboru
Imageline(im,x1,y1,x2,y2,c)	Nakreslí čáru do obrázku <i>im</i> z bodu [<i>x1,y2</i>] do bodu [<i>x2;y2</i>] barvou <i>c</i>

3.2.3. Funkční příklad užití GD

Abych tuto teorii předvedl také na některém praktickém příkladě, ukážu nyní nakreslení základního uspořádání Mössbauerova spektrometru uvedený v kapitole Mössbauerův spektrometr (Obrázek 2). Tento obrázek je právě výsledkem tohoto zdrojového kódu:

```
<?
//moss_spektometr.php

//nastavení rozměrů obrázku
$im_width=600;
$im_height=400;

//Vytvoření plátna plátna
$image=imagecreatetruecolor($im_width,$im_height);

//Nastavení barev
$border=imagecolorallocate($image, 0, 0, 0);
$background=imagecolorallocate($image, 255, 255, 255);
$black=imagecolorallocate($image, 0, 0, 0);

//vykreslení čtverce s plným pozadím (představuje pozadí)
imagefilledrectangle($image,0,0,$im_width,$im_height,$background);

//vykreslení čtverce jako rámu
imagerectangle($image,0,0,$im_width-1,$im_height-1,$border);

//vykreslení rámečku pro generátoru rychlosti
imagerectangle($image,20,$im_height-100,160,$im_height-20,$black);

//text generátoru rychlosti
imagefttext($image,15,0,40,$im_height-65,$black,'arialbd.ttf',"Generátor");
imagefttext($image,15,0,40,$im_height-45,$black,'arialbd.ttf',"rychlosti");

//vykreslení rámečku pro vibrátor
imagerectangle($image,20,60,160,140,$black);

//text vibrátor
imagefttext($image,15,0,40,110,$black,'arialbd.ttf',"Vibrátor");

//vykreslení rámečku pro dektor
imagerectangle($image,$im_width-160,60,$im_width-20,140,$black);

//text detektor
```

```

imagefttext($image,15,0,$sim_width-140,110,$black,'arialbd.ttf',"Detektor");

//vykreslení rámečku pro registrační zařízení
imagerectangle($image,$sim_width-160,$sim_height-100,$sim_width-
20,$sim_height-20,$black);

//text registrační zařízení
$text=iconv("windows-1250","UTF-8","Registrační");
imagefttext($image,15,0,$sim_width-140,$sim_height-65,$black,'arialbd.ttf',$text);
$text=iconv("windows-1250","UTF-8","zařízení");
imagefttext($image,15,0,$sim_width-140,$sim_height-45,$black,'arialbd.ttf',$text);

//vykreslení rámečku pro diskriminátor
imagerectangle($image,$sim_width-160,round($sim_height/2)-20,$sim_width-
20,round($sim_height/2)+60,$black);

//text diskriminátor
$text=iconv("windows-1250","UTF-8","Diskriminátor");
imagefttext($image,15,0,$sim_width-
150,round($sim_height/2)+30,$black,'arialbd.ttf',$text);

//vykreslení zářiče (3x čára = tlustá čára a vyplněný obdélník)
imageline($image,160,99,180,99,$black);
imageline($image,160,100,180,100,$black);
imageline($image,160,101,180,101,$black);
imagefilledrectangle($image,180,90,190,110,$black);

//text zářič
$text=iconv("windows-1250","UTF-8","Zářič");
imagefttext($image,10,0,170,80,$black,'arialbd.ttf',$text);

//vykreslení čárkované čáry od zářiče k deektoru
for ($i=180;$i<=420;$i=$i+20)
imageline($image,$i,100,$i+10,100,$black);

//alokace barvy a vykreslení jednotlivých kolimátorů
$c_kolim=imagecolorallocate($image, 80, 80, 80);
imagefilledrectangle($image,220,30,230,90,$c_kolim);
imagefilledrectangle($image,220,110,230,170,$c_kolim);
imagefilledrectangle($image,380,30,390,90,$c_kolim);
imagefilledrectangle($image,380,110,390,170,$c_kolim);

//text kolimátory
$text=iconv("windows-1250","UTF-8","Kolimátory");
imagefttext($image,10,0,270,20,$black,'arialbd.ttf',$text);

//alokace barvy a vykreslení absorberu
$c_abs=imagecolorallocate($image, 180, 180, 180);
imagefilledrectangle($image,320,50,330,150,$c_abs);

```

```

//text absorber
imagefttext($image,10,0,270,170,$black,'arialbd.ttf','Absorber');

//vykreslení šipky od generátoru rychlosti k vibrátoru
imageline($image,90,$im_height-100,90,140,$black);
imageline($image,80,150,90,140,$black);
imageline($image,100,150,90,140,$black);

//vykreslení šipky u zářiče
imageline($image,175,120,195,120,$black);
imageline($image,190,125,195,120,$black);
imageline($image,190,115,195,120,$black);
imageline($image,175,120,180,125,$black);
imageline($image,175,120,180,115,$black);

//vykreslení šipky od detektoru k diskriminátoru
imageline($image,510,140,510,180,$black);
imageline($image,505,175,510,180,$black);
imageline($image,515,175,510,180,$black);

//vykreslení šipky od diskriminátoru k registračnímu zařízení
imageline($image,510,260,510,300,$black);
imageline($image,505,295,510,300,$black);
imageline($image,515,295,510,300,$black);

//Vlastní vytvoření obrázku
header('Content-Type: image/png' );
imagepng( $image );

?>

```

Příklad 1 - Vytvoření obrázku Mössbauerova spektrometru pomocí knihovny GD

Mám tedy zdrojový kód pro vytvoření obrázku. A jakým způsobem ho můžu zobrazit v prohlížeči? Existují dva způsoby. Buďto ho zobrazím zadáním přímé cesty ke zdrojovému kódu a na stránce se zobrazí pouze obrázek, nebo ho zavolám z jiného zdrojového kódu klasickým způsobem volání obrázku, tedy tagem *img* (Příklad 2), kdy je parametrem *src* zdrojový kód pro vytvoření obrázku.

```



```

Příklad 2 - volání obrázku nakresleného pomocí knihovny GD

3.3.Tvorba grafů pomocí JpGraphu

A nyní se již dostávám k samotnému souboru knihoven JpGraph, který slouží k tvorbě grafů pomocí grafické knihovny GD. Tu jsem popsal v předchozí kapitole.

JpGraph vytvořila soukromá švédská firma Aditus Consulting. Jedná se o objektově orientovaný soubor knihoven pro PHP. Kompletně je napsán v PHP a může se použít v jakýchkoliv PHP skriptech. Jediná podmínka pro funkčnost, pomínu-li samozřejmě PHP, je již zmíněné rozšíření o grafickou knihovnu GD.

Vydání první verze JpGraphu 1.0 se datuje na 5. února roku 2001. Od té doby prošel velkým vývojem. JpGraph existuje ve dvou základních sériích, a to série 1.x, která je určena pro PHP verze 4 (ve verzi PHP 5 nefunguje) a série 2.x určená právě pro PHP verze 5. Ve své práci využívám aktuální poslední verzi JpGraph 2.2, která vyšla 9. dubna 2007.

Výsledné grafy jsou prezentovány jako obrázky. Tyto obrázky lze buďto zobrazit na stránkách, nebo je rovnou uložit do souboru. Uživatelé JpGraphu si mohou vybrat typ výsledného obrázku, podporované typy jsou *TTF*, *PNG* a *JPEG*.

3.3.1. Využití a instalace JpGraphu

Spektrum využití JpGraphu je velmi široké, základní činností a vůbec důvodem vzniku JpGraphu je však tvorba grafů. Jednoduše tak lze vytvořit grafy sloupcové, spojnicové, plošné, bodové, koláčové, 3D nebo pavučinové. Samozřejmě je možné vytvořit i graf kombinovaný, který se skládá ze dvou a více základních grafů.

Existuje ovšem i profesionální placená verze JpGraphu. Ta obsahuje i moduly pro vytvoření čárových kódů, tachometrů, větrných růžic a grafů s tabulkami podobných tabulkovému editoru Excel.

K instalaci je nutné stáhnout instalační balíček z oficiálních stránek JpGraphu, které jsou dostupné na URL adrese <http://www.aditus.nu/jpgraph/jpdownload.php>.

Balíček se musí rozbalit a umístit ke zdrojovému kódu. Tím je instalace prakticky hotová. Ovšem pro vytváření jednotlivých grafů je potřeba ke zdrojovému kódu přilinkovat potřebné knihovny s obslužnými funkcemi. Pro každý typ grafu existuje vlastní knihovna. Základní (a vždy nutná) knihovna se nachází v souboru *jpgraph.php*, která obsahuje základní funkce pro vytvoření grafu a nachází se v adresáři *src*. Soubory knihoven se přilinkují klasickým způsobem (Příklad 3).

```
include ("cesta_k_distribuci_jpgraphu/src/jpgraph.php");
```

Příklad 3 - přilinkování knihovny *jpgraph.php*

3.3.2. Základní postup vytvoření grafu

A nyní již nastal čas na malou ukázkou vytvoření grafu pomocí souboru knihoven JpGraphu. Ukážu postup vytvoření spojnicového grafu. Tedy grafu, který je vytvářen v aplikaci.

Jako první krok musíme přilinkovat knihovny pro tvorbu grafu (obecného, bez něj by se graf nevytvořil) a pro graf spojnicový. Obě dvě knihovny se nacházejí v adresáři *src*, viz Příklad 4.

```
include ("cesta_od_zdrojového_kódu/src/jpgraph.php");  
include ("cesta_od_zdrojového_kódu/src/jpgraph
```

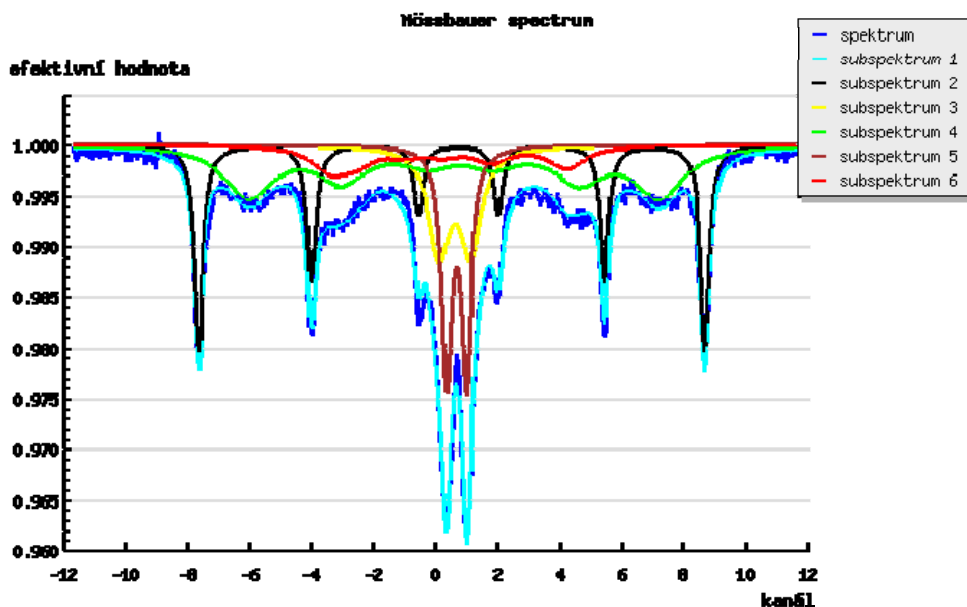
Příklad 4 - načtení knihoven pro tvorbu spojnicového grafu_line.php");

Dalším krokem je získání souřadnic osy x a osy y. Jako nejlepší varianta se zde nabízí využití pole (v příkladu jsou hodnoty osy x zapsány v poli *XData* a hodnoty osy y v poli *YData*). A nyní nám již nic nebrání ve vytvoření plátna pro výsledný obrázek grafu. Následuje nastavení titulku grafu a popsání os a jeho formátování. Na vytvořené plátno pak lze nakreslit *LinePlot*, který zobrazuje spojení jednotlivých zadaných souřadnic. Další možností je vytvoření legendy. Ta asi nemá smysl v případě grafu s jedním subspektrém. Poslední akcí je pak odeslání grafu na výstup, tedy vytvoření konečného obrázku. Požité funkce a jejich popis ukazuje Příklad 5 - vytvoření spojnicového grafu.

```
//vytvoření plátna pro graf o rozměrech 650x400px
$graph = new Graph(650,400,'auto');
//nastavení titulku
$graph->title->Set ("MÖSSBAUER SPECTRUM");
//nastavení popisků os – použití textu místo title->yaxis a xaxis
//vytvoření textu na pozici 500x385px
$txt = new Text('kanál',500, 385);
//nastavení fontu
$txt->SetFont(FF_FONT1,FS_BOLD);
//přidání textu do grafu/obrázku
$graph->AddText($txt);
//to samé s popiskem osy x
$txt = new Text('efektivní hodnota',5, 35);
$txt->SetFont(FF_FONT1,FS_BOLD);
$graph->AddText($txt);
//Vytvoření nového LinePlotu s hodnotami Xdata a Ydata (načtena mimo tento kód)
$lineplot=new LinePlot($Ydata, $Xdata);
//Přidání LinePlotu do grafu
$graph->Add($lineplot);
//vytvoření legendy grafu – s parametrem $leg (název datové řady)
$lineplot ->SetLegend($leg);
// zobrazení grafu (odeslání na výstup)
$graph->Stroke();
```

Příklad 5 - vytvoření spojnicového grafu

Výsledek tohoto skriptu po dosazení spektra a 6 subspekter (tedy sedmkrát vytvořen nový *LinePlot* a dosazen do grafu a legendy) o názvech zobrazuje Obrázek 4.



Obrázek 4 - spojnicový graf vytvořený v příkladu

3.4.Tvorba PDF dokumentů pomocí PHP

Aplikace umožňuje mimo zobrazování výsledků měření také tvorbu protokolů. Aplikace sice umožňuje výběr dvou formátů protokolů – HTML a PDF, ale stěžejní je právě formát PDF. V této kapitole se zaměřím ani ne tak na tvorbu konkrétních protokolů, ale na tvorbu obecných dokumentů PDF.

3.4.1. Knihovny PDFlib a FPDF

Existuje více cest k vytvoření PDF dokumentů. Za zmínku stojí například knihovna PDFlib, která je podobně jako knihovna GD nadstavbou PHP. Pochopitelně se tedy jeví jako nejlepší řešení aplikovat ji ve výsledné

aplikaci. Proč tedy není využita? Odpověď je prostá. Problémy se starými verzemi PDFlib. Přímo na oficiálních stránkách PHP se píše toto:

“ Žádná verze PHP 4 od data 9. března 2000 nepodporuje pdflib starší než 3.0. PDFlib 3.0 a vyšší je podporováno PHP 3.0.19 a vyšší. “ (2)

Naneštěstí ještě k tomu každá verze PDFlib obsahuje na první pohled stejné, přesto jiné funkce. Některé funkce (myšleno jejich názvy) zůstaly zachovány, změnil se však jejich parametry a u některých i významy, a některé funkce se zcela přejmenovaly. Navíc v PHP 5 pro mě byl neřešitelný problém přidání obrázku do dokumentu PDF. Ať už sem se vydal cestou absolutní, nebo relativní, nepodařilo se mi do dokumentu vložit obrázek existující, natož pak vygenerovaný. V některých případech dokonce došlo k „zamrznutí“ internetového prohlížeče.

A tak jsem se vydal cestou knihovny FPDF. Knihovna FPDF je objektově orientovaná knihovna určená pro generování PDF dokumentů. F před slovem PDF znamená Free – lze tedy knihovnu použít kdekoliv v kódu a případně ji i modifikovat.

Knihovna FPDF obsahuje mnoho funkcí (namátkou třeba jednoduché definování jednotek, velikosti stránek a jejich okrajů, vytváření záhlaví a zápatí s automatickým číslováním stran, komprese stránek apod.). Hlavní výhodou této knihovny spočívá v tom, že není nutné kompilovat podporu tvorby PDF pro PHP, vystačíme si pouze s knihovnou samotnou¹⁷.

Avšak i tato knihovna má jednu nepříjemnou vlastnost. Její základ nepodporuje české znaky a české kódování. Musí se tak české písmo a

¹⁷ Instalační balíček a manuál lze zdarma stáhnout na oficiální stránkách knihovny <http://www.fpdf.org/>.

kódování přidávat „ručně“. Nevýhodou je samozřejmě narůstající velikost dokumentu s každým dalším přidaným písmem.

3.4.2. Základní postup vytvoření PDF dokumentu

A opět nastal čas na popsání základního postupu. Tentokrát postupu při vytváření PDF dokumentu pomocí knihovny FPDF. Ukážu, jaký postup jsem zvolil při generování PDF protokolu v aplikaci. Dokument PDF bude obsahovat hlavičku a patu, a nějaký vlastní obsah. Konkrétní obsah protokolu vytvářeného v aplikaci představím až v kapitole 4.5 Protokoly o měření.

Jak už to bývá v případě použití pomocné knihovny, musí se knihovna nejdříve přilinkovat. Zde k dokumentu přilinkuji knihovnu FPDF. Pak již můžu vytvořit vlastní objekt knihovny FPDF. Dále nastavím alias pro čísla stránek (pomocí něj pak knihovna FPDF generuje čísla stránek) a vytvořím první stránku.

Hlavičku a patu vytvořím pomocí funkcí `Header()` a `Footer()`. Funkce se musí jmenovat přesně takhle, knihovna tyto názvy automaticky spojuje s hlavou a patou a obsah funkce `Header()` a `Footer()` vypíše jako záhlaví, resp. zápatí, každé stránce. Tyto funkce ale musíme zapouzdřit do třídy PDF.

Po vygenerování obsahu stránky už stačí odeslat objekt PDF pomocí funkce `Output()`. Projdu teď základní funkce knihovny PDF, pomocí nichž lze nastavit a vypsát text, vložit obrázek apod. Nejdůležitější funkce ukazuje Tabulka 2.

Tabulka 2 - Přehled funkcí knihovny FPDF

Funkce	Popis
AddFont (family [,style , file])	Importuje font True Type nebo Type1 family – rodina písem style – styl (řez) písma (B,I,BI – tučné, kurzíva) file – soubor písma
AddPage ([orientation])	Vytvoření nové stránky orientation – orientace stránky (P,L – na výšku, na šířku)
AliasNbPages ([alias])	Definování aliasu pro čísla stránek alias – alias, defaultně {nb}
Cell (w [,h , t , b, ln, al, fi , li])	Vytvoření buňky w – šířka (0 = celou šířku) h – výška t – text pro vypsání b – rámeček (0, 1, L, R, T, D – žádný, celý, vlevo, vpravo, nahoře, dole) al – zarovnání textu (L,R,C - vlevo, vpravo, na střed) fi – vykreslení pozadí (0, 1 – nevykreslit, vykreslit) li – odkaz na URL
Close()	Uzavře dokument , není však potřeba, funkce Output() uzavírá dokument automaticky
FPDF ([or [, unit, format]])	Konstruktor pro vytvoření PDF dokumentu or – orientace (viz fce Cell()) unit – jednotky dokumentu (pt, mm, cm, in – body, milimetry, centimetry, palce) – defaultní jsou mm format – formát stránky (A3, A4, A5, Letter, Legal, nebo pole o dvou hodnotách – šířka a výška)
GetStringWidth (s)	Vrací délku řetězce (v definovaných jednotkách) s – řetězec u kterého chceme zjistit délku
GetX()	Vrací souřadnici osy x aktuální pozice
GetY()	Vrací souřadnici osy y aktuální pozice
Image(file, x, y [,w , h, type, link])	Vloží do dokumentu obrázek file – cesta a název souboru s obrázkem x,y – pozice pro umístění levého horního rohu obrázku w – šířka obrázku h – výška obrázku type – typ obrázku (JPG, JPEG, PNG) link - odkaz na URL
Line (x1, y1, x2, y2)	Nakreslení čáry x1,y1 – souřadnice bodu začátku čáry x2,y2 – souřadnice bodu konce čáry

Output ([name])	Vytvoří PDF dokument name – uloží dokument do souboru, pokud není zadán, PDF je odesláno do prohlížeče
PageNo()	Vrátí číslo aktuální stránky
Rect (x, y, w, h [, style])	Nakreslí kružnici x,y – střed kružnice w,h – šířka (výška) style – styl (D, F, DF – vykreslení obrysu, vyplnění, obrys + vyplnění)
SetAuthor (author)	Nastaví autora dokumentu author – jméno autora
SetAutoPageBreak (auto [,margin])	Nastavení automatického zalamování stránky auto – true nebo false margin – vzdálenost od spodku stránky
SetCreator (creator)	Nastaví tvůrce dokumentu creator – jméno tvůrce
SetDisplayMode (zoom [, layout])	Nastavení vzhledu stránky v prohlížeči zoom – nastavení velikosti (fullpage, fullwidth, real, default – celá obrazovka, roztáhnout do šířky okna, reálná velikost, defaultní velikost podle nastavení prohlížeče) layout – uspořádání stránek (single, continuous, two, default – vždy pouze jedna stránka, navazující stránky, dvě stránky, podle defaultního nastavení prohlížeče)
SetDrawColor (r [, g, b])	Nastaví barvu pro obrysy kreslicích funkcí r, g, b – číselné hodnoty barev (červená, zelená, modrá)
SetFillColor (r [, g, b])	Nastaví barvu pro výplně kreslicích funkcí r, g, b – číselné hodnoty barev (červená, zelená, modrá)
SetFont (family [,style , size])	Nastavení formátu písma family – rodina písem style – styl (řez) písma (B,I,BI – tučné, kurzíva) size – velikost v bodech (defaultní 12)
SetFontSize (size)	Nastavení velikosti písma size – velikost písma v bodech
SetLeftMargin (margin)	Nastavení levého okraje stránky margin – velikost okraje
SetLineWidth (width)	Nastavení tloušťky čáry width – tloušťka čáry
SetMargins (left, top [,right])	Nastavení okrajů stránky left – levý okraj top – horní okraj right – pravý okraj (pokud není vyplněno, je stejný jako levý)
SetRightMargin (margin)	Nastavení pravého okraje stránky margin – velikost okraje
SetSubject	Nastavení předmětu dokumentu

(subject)	subject – předmět dokumentu
SetTextColor (r [,g, b])	Nastaví barvy pro text r, g, b – číselné hodnoty barev (červená, zelená, modrá)
SetTitle (title)	Nastavení titulku dokumentu title – titulek dokumentu
SetTopMargin (margin)	Nastavení horního okraje stránky margin – velikost okraje
SetX (x)	Nastavení aktuální polohy souřadnice osy x
SetXY(x, y)	Nastavení aktuální pozice kurzoru x,y – hodnoty souřadnice osy x a y
SetY(y)	Nastavení aktuální polohy souřadnice osy y
Text (x, y, txt)	Vypsání textu txt na pozici x,y
Write (h, txt [, link])	Vypsání textu od aktuální pozice kurzoru h – výška řádku txt – text pro vypsání link - odkaz na URL

3.4.3. Ukázka vytvoření PDF dokumentu

Celou teorii vytváření PDF dokumentů pomocí knihovny FPDF si teď ukážeme. Bude se jednat o první stránku protokolu z měření. Stránka bude obsahovat záhlaví s nápisem Protokol měření a zápatí s číslem aktuální stránky a celkovým počtem stránek. Vlastní stránka pak bude obsahovat textový popis měření a obrázek grafu – Příklad 6.

```
//načtení knihovny FPDF
require('fpdf.php');

//definování zápatí a záhlaví
class PDF extends FPDF
{
    //Hlavička
    function Header()
    {
        //Nastavení písma Arial, tučného, velikosti 15
        $this->SetFont('Arial','B',15);

        //Posun buňky na pozici 80
        $this->Cell(80);
    }
}
```



```

//Vypsání textu (titulku) na pozici 30x10
$this->Cell(30,10,"Protokol",0,0,'C');

//Line break
$this->Ln(20);
}

//zápatí
function Footer()
{
    //Nastavení pozice 1.5 cm od spodu
    $this->SetY(-15);

    //Nastavení písma Arial, kurzívy, velikosti 8
    $this->SetFont('Arial','I',8);

    //Vypsání čísla stránek
    $this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C'); }
}

//vytvoření nového objektu PDF
$pdf=new PDF();

//nastavení aliasu pro čísla stránek
$pdf->AliasNbPages();

//přidání stránky do dokumentu
$pdf->AddPage();

//přidání písma Times (s českými znaky)
$pdf->AddFont('Times','','times.php');

//nastavení písma Times o velikosti 20
$pdf->SetFont('Times','',20);

//vypsání nadpisu INFORMACE O MĚŘENÍ
$pdf->Cell(0,20,'INFORMACE O MĚŘENÍ',0,1,'C');

//nastavení písma Times o velikosti 12
$pdf->SetFont('Times','',12);

//vypsání názvu, datumu a popisu měření z proměnných $nazev, $datum a $popis
$pdf->Cell(0,8,'Název měření: ' . $nazev,0,1);
$pdf->Cell(0,8,'Datum měření: ' . $datum,0,1);
$pdf->MultiCell(0,8,'Popis měření: ' . $popis);

//Nastavení fontu Times o velikosti 20 bodů
$pdf->SetFont('Times','',20);

//Vypsání nadpisu
$pdf->Cell(0,30,'Zobrazení všech spekter',0,1,'C');
//vložením obrázku grafu ze souboru ,tmp/all.png' na pozici [80;60] o šířce 140
(výška se přepočítá automaticky)
$pdf->Image('tmp/all.png',80,60,140);

```

```
//odeslání objektu PDF do prohlížeče (vytvoření dokumentu PDF)
$pdf->Output();
```

Příklad 6 - Vytvoření dokumentu PDF pomocí knihovny FPDF

3.5.Tvorba CSV dokumentů pomocí PHP

Poslední problematice, které se budu v teoretické části věnovat, je tvorba CSV dokumentů. Ze všech doposud popsaných částí se jedná o nejjednodušší postup, jelikož sám dokument CSV je velice jednoduchý. Obsahuje pouze data oddělená čárkami, nebo středníky. Využívá se zejména pro předávání tabulkových dat.

V aplikaci budu do dokumentu CSV importovat data výsledků jednotlivých měření. Do prvního řádku uložím informace o názvu měření. V druhém se již vyskytují data osy x a v dalších následují jednotlivé hodnoty spektra a subspekter. Vše opět ukážu na příkladu.

```
<?
//otevřeme soubor (případně vytvoříme) protokol.csv pro zápis
$soubor= fopen('protokol.csv','w');

//z databáze načteme názav měření
$dotaz="SELECT id,nazev FROM mereni;";
$vypis=mysql_query($dotaz, $link) or die ("SQL dotaz nešlo provést");
while ($rada=mysql_fetch_array($vypis)){
    //zapišeme do souboru název měření – důležitý ENTER na konci řádku
    fwrite($soubor, "NÁZEV MĚŘENÍ;" . $rada['nazev'] . '
');

//načteme z databáze hodnoty osy x
$dotaz="SELECT hodnota FROM data_x WHERE mereni=" . $cislo_mereni . ";";
$vypis=mysql_query($dotaz, $link) or die ("SQL dotaz nešlo provést"); $i=0;
while ($rada2=mysql_fetch_array($vypis)){
    $data_x[$i]=$rada2['hodnota'];
    $i++;
}

//procházíme jednotlivé subspektra
$dotaz="SELECT id,nazev FROM kanaly WHERE mereni=" . $cislo_mereni . ";";
```

```

$vypris=mysql_query($dotaz, $link) or die ("SQL dotaz nešlo provést");
while ($rada=mysql_fetch_array($vypris)){

    //ulozime jmena subspekter
    $nazvy[$i]=$rada3['nazev'];

    //procházíme data jednotlivých subspekter
    $dotaz2="SELECT hodnota FROM data_y WHERE kanal=" . $rada['id'] . ";";
    $vypris2=mysql_query($dotaz2, $link) or die ("SQL dotaz2 nešlo provést");
    $j=0;
    while ($rada2=mysql_fetch_array($vypris2)){
        //uložíme všechny hodnoty
        $data_y[$i][$j]=$rada4['hodnota'];
        $j++;
    }

    //do souboru zapíšeme názvy
    $nazvy_pro_zapsani='x;';
    for ($i=0; $i<count($nazvy);$i++)
        $nazvy_pro_zapsani.=$nazvy[$i] . ',';
    fwrite($soubor, $nazvy_pro_zapsani . '
');

    //zapíšeme do souboru veškerá data
    for ($i=0; $i<count($data_x);$i++){
        $radek=$data_x[$i] . ',';
        for ($j=0; $j<count($nazvy); $j++){
            $radek.=$data_y[$j][$i] . ',';
        }
        fwrite($soubor, $radek . '
');
    }

    //uzavřeme soubor – dokument CSV je hotov
    fclose($soubor);
?>

```

Příklad 7 - vytvoření CSV dokumentu

4. APLIKACE

V rámci bakalářské práce jsem vytvořil WWW aplikaci pro prezentaci naměřených spekter v laboratoři pro výzkum nanomateriálu. Aplikaci jsem pojmenoval PreSpe¹⁸.

WWW aplikace umožňuje zobrazení výsledků měření v podobě grafu. Spektra a subspektra lze zobrazit v jednom výsledném grafu, nebo každé zvlášť. Při zobrazení jednoho výsledného grafu si může uživatel navolit, která subspektra se mají v grafu zobrazit. Další zajímavou vlastností je možnost ovlivnit pořadí subspekter v zobrazení. Pro vytvoření grafu aplikace využívá soubor knihoven JpGraph švédské firmy Aditus Consulting.

Aplikace PreSpe také umožňuje tvorbu protokolů o měření. Protokoly lze zobrazit (a uložit) ve formě PDF nebo HTML. Jednotlivé protokoly obsahují informace o měření a výsledky zpracované ve formě grafů.

Veškerá data o proběhlém měření aplikace čerpá z tabulek uložených na MySQL serveru. Data je možné exportovat do cvs dokumentů, které lze velmi jednoduše otevřít např. v aplikaci MS Office Excel.

4.1.Požadavky

Pro správný chod aplikace je třeba mít na serveru nainstalované PHP s grafickou knihovnou GD a databázový server MySQL na kterém jsou uložena veškerá data pro prezentaci (výsledky měření v laboratoři). Soubory

¹⁸ PreSpe je zkratka Prezentace Spekter

knihoven JpGraphu (pro tvorbu grafů) a knihovnu FPDF (pro generování PDF dokumentů) není třeba instalovat, jsou již obsaženy v balíčku PreSpe.

Aplikace musí mít práva pro zápis do adresářů „zpravy“ (na toto místo se ukládají dokumenty csv) a „vystupy_soubory/tmp“. Tento adresář je využíván při generování pdf protokolů.

Tabulka 3 - požadavky pro správný chod aplikace PreSpe

Požadavek	Použitá verze	Minimální verze
PHP	5.1.1	5.0
GD	2.0.28	2.0
FPDF	1.53	Je obsažena v aplikaci
JpGraph	2.2	Je obsažena v aplikaci

4.2. Instalace PreSpe

Aplikaci lze distribuovat jako balíček. Instalace není třeba, stačí jen balíček rozbalit na serveru a spustit v prohlížeči. Jediné, co je třeba pro bezproblémové fungování PreSpe je konfigurace aplikace. Aplikaci lze konfigurovat dvěma způsoby – úpravou konfiguračního souboru *config.php* nebo pomocí internetového prohlížeče (Obrázek 5). Doporučuji zvolit druhý způsob, jelikož se pak nemůže přihodit nechtěné smazání části skriptu a následná nefunkčnost aplikace.

Pokud se aplikace PreSpe spouští poprvé, je uživatel automaticky přeměřován do konfigurace. To se stane i v případě, že konfigurační soubor není nalezen, je poškozen, či konfigurace neproběhla v pořádku¹⁹. Konfiguraci navíc lze kdykoliv modifikovat, stačí se do ní přepnout odkazem *konfigurace* ve spodní části každé stránky internetové aplikace. V následující podkapitole je popsána konfigurace pomocí internetového prohlížeče.

¹⁹ Před zahájením jakékoliv akce aplikace PreSpe je testována konfigurace z důvodu možnosti (byť třeba nechtěného) smazání konfiguračního souboru, nebo zásahu do konfiguračního souboru či nedosažitelnosti nastavené databáze.

Věnujte prosím pozornost tomuto nastavení

Nastavení

Konfigurace

Nastavení aplikace

Titulní název:

Nastavení databáze

Server:

Název databáze:

Uživatel:

Heslo databáze:

Nastavení měření

Měření:

ULOŽENÍ KONFIGURACE

Jakub Král © 2007 konfigurace

Obrázek 5 - konfigurace PreSpe v internetovém prohlížeči

4.2.1. Konfigurace v internetovém prohlížeči

Postup konfigurace:

1. Nastavení aplikace
2. Nastavení spojení s databází
3. Nastavení měření

První je tedy na řadě nastavení aplikace. Zde uživatel zadá titulní název, který se zobrazuje jako hlavní nadpis na každé stránce internetové aplikace PreSpe.

Následuje nastavení databáze. Bez správného nastavení a spojení s databází program neumožní pokračování. Povinně se musí zadat server a název databáze. Nepovinnými, ale rozhodně doporučenými údaji vzhledem

k bezpečnosti, je nastavení uživatelského jména a hesla pro připojení do databáze. Volnost tohoto nastavení však nezávisí na volbě uživatele, ale na nastavení konkrétní databáze.

Spojení s databází lze kdykoliv otestovat tlačítkem *Otestuj spojení*. V tomto případě se provede znovunačtení konfigurace se zadanými údaji a vypíše se úspěšnost pokusu připojení na server a databázi. Pro správnou funkčnost se musí podařit spojení se serverem i databází. Jaké tabulky musí databáze obsahovat popisuje kapitola 4.3 Databáze MySQL.

Posledním krokem v instalaci je nastavení měření. Zde se zadává, které konkrétní měření má být prezentováno. Pokud však není navázáno spojení s databází, nelze měření vybrat. Je tedy nutné před tímto výběrem minimálně otestovat spojení s úspěšným koncem, případně konfiguraci uložit a poté výběr provést.

V případě úspěšného nastavení a uložení konfigurace nabídne přechod na hlavní stránku PreSpe. Jak již bylo zmíněno, do konfigurace se lze kdykoliv vrátit pomocí odkazu *Konfigurace* ve spodní liště WWW aplikace.

A jakým způsobem se konfigurace ukládá do konfiguračního souboru? Po každém stisknutí tlačítka *Ulož konfiguraci* je celý konfigurační soubor buď přepsán novými hodnotami, nebo v případě, že konfigurační soubor neexistuje, vytvořen zcela nový.

4.2.2. Konfigurační soubor

A jak tedy vypadá konfigurační soubor? Je umístěn v kořenové složce aplikace a je pojmenován *config.php*. Jelikož se ve skutečnosti jedná o php kód, soubor obsahuje právě php proměnné. Celkem jich je šest a každá

musí bezpodmínečně existovat. Je potřeba dodržovat jejich přesné názvy. Chybějící (nebo nesprávně pojmenované) proměnné nezpůsobí pád programu, ale přesměrování do konfigurace. Veškeré nutné proměnné a jejich význam popisuje Příklad 8 - Ukázka konfiguračního souboru aplikace PreSpe.

```
<?php
////////////////////////////////////
//NASTAVENÍ DATABÁZE//
////////////////////////////////////

//Databázový server
$databaze_server="localhost";

//Název databáze
$databaze_name="bakalarka";

//Uživatelské jméno
$databaze_user_name="admin";

//Heslo pro databázi
$databaze_pass="";

//ID právě zorbazovaného měření
$databaze_id_mereni="1";

////////////////////////////////////
//NASTAVENÍ ZOBRAZENÍ//
////////////////////////////////////

//Název na stránce
$page_title="Mössbauer spectrum";
?>
```

Příklad 8 - Ukázka konfiguračního souboru aplikace PreSpe

4.3.Databáze MySQL

Databáze MySQL je tedy hlavním úložištěm dat. Už jsem popsal, jak se k databázi připojit a jak vybrat konkrétní měření. Zbývá však ještě říci, jaké má databáze obsahovat tabulky a jakou strukturu mají mít.

Již z konfigurace vyplívá benevolentnost aplikace vůči názvu databáze (může se tedy jmenovat teoreticky libovolně), avšak vůči tabulkám a jejich struktuře již je velice striktní. Databáze musí obsahovat přesně definované tabulky, které ukazuje Tabulka 4.

Tabulka 4 - Seznam tabulek databáze MySQL aplikace PreSpe

Název tabulky	Kódování
mereni	utf8_general_ci
data_x	utf8_general_ci
spektra	utf8_general_ci
data_y	utf8_general_ci

V tuto chvíli je nutné se pozastavit nad jednotlivými parametry vytvořených tabulek. Tabulka *mereni* (Tabulka 5) obsahuje obecné informace o konkrétním měření, tedy jeho název, datum, kdy se měření uskutečnilo, a jeho popis. Název měření²⁰ a datum se zobrazuje v konfiguraci při výběru měření, tedy v posledním kroku konfigurace. Popis je pak důležitý pro vytvoření protokolů, avšak není, oproti názvu a datu měření, nezbytností.

Tabulka 5 - PreSpe databáze - tabulka mereni

Sloupec	typ	Kódování
id	int	
nazev	varchar(30)	utf8_general_ci
datum	date	
popis	text	utf8_general_ci

Tabulka *data_x* (Tabulka 6) už obsahuje data měření. Již podle názvu je patrné, že se jedná o hodnoty osy *x*. Krom vlastních hodnot má tabulka ještě sloupec nazvaný *mereni*, která určuje vazbu mezi hodnotou a měřením tabulky *mereni*.

²⁰ Maximální délka názvu měření je 30 znaků

Tabulka 6 - PreSpe databáze - tabulka *data_x*

Sloupec	typ	Kódování
id	int	
mereni	int	
hodnota	float	

Každé spektrum se dá rozložit na subspektra. Na to pamatuje i databáze aplikace PreSpe. Právě pro tento účel zahrnuje databáze tabulku *spektra* (Tabulka 7), která obsahuje naměřené spektrum a jednotlivá subspektra. Každému subspektru lze přiřadit název a vazbu na tabulku měření.

Tabulka 7 - PreSpe databáze - tabulka *spektra*

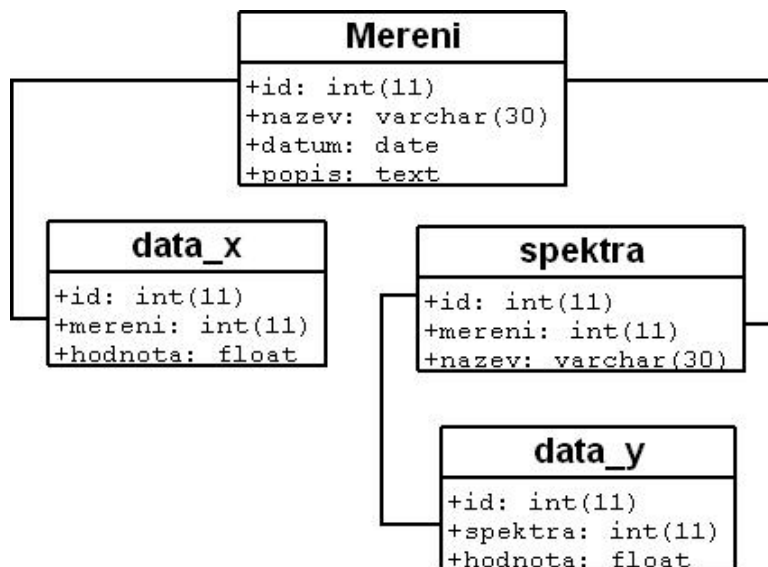
Sloupec	typ	Kódování
id	int	
mereni	int	
nazev	varchar(30)	utf8_general_c

Poslední povinnou tabulkou je tabulka *data_y* (Tabulka 8). Její struktura je podobná tabulce *data_x*, liší se pouze ve vazbě. Hodnoty se nevážou přímo na měření, ale na jednotlivé spektra či subspektra.

Tabulka 8 - PreSpe databáze - tabulka *data_y*

Sloupec	typ	Kódování
id	int	
spektra	int	
hodnota	float	

Přesné názvy tabulek a sloupců a jejich atributů jsou důležité pro čtení dat při vytváření grafů. Seběmenší odchylka může způsobit, že aplikace nezobrazí výsledky správně, případně vůbec. Veškeré vazby mezi jednotlivými tabulkami můžeme vyjádřit pomocí UML diagramu (Obrázek 6).



Obrázek 6 - UML diagram databáze aplikace PreSpe

4.4. Zobrazení spekter

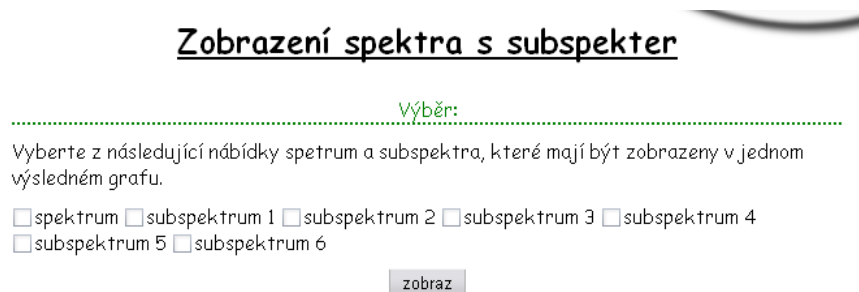
Po úspěšné konfiguraci PreSpe mohou přistoupit k jednotlivým funkcím aplikace. Tou hlavní je zobrazování spektra a subspekter. Ty můžeme zobrazit všechna (nebo jen vybraná) v jednom grafu, nebo jednotlivě.

V této části se již nebudu zabývat principem vytváření jednotlivých grafů (rozebral jsem v kapitole 3.3 – Tvorba grafů pomocí JpGraphu), ale postupem, který jsem zvolil při konkrétním úkonu.

4.4.1. Zobrazení spektra a subspekter v jednom grafu

Jak již sám název napovídá, v této části bude popsán postup vytvoření grafu, který obsahuje spektrum a subspektra. Aplikace navíc umožňuje uživateli zadat, která subspektra se mají v grafu zobrazit. Zajímavá je i možnost ovlivnění pořadí jednotlivých subspekter v zobrazení.

Výběr spektra a subspekter je v aplikaci realizován pomocí zaškrtnutých políček (Obrázek 7). Jejich počet a název je načítán z databáze konkrétního měření. Uživatel tak přesně může specifikovat výsledný graf a označit, která subspektra²¹ se mají zobrazit.



Zobrazení spektra s subspekter

Výběr:

Vyberte z následující nabídky spektrum a subspektra, které mají být zobrazeny v jednom výsledném grafu.

spektrum subspektrum 1 subspektrum 2 subspektrum 3 subspektrum 4
 subspektrum 5 subspektrum 6

Obrázek 7 - formulář pro výběr zobrazovaných spekter

Postup je tedy jasný – načíst subspektra, které uživatel zvolil pro zobrazení a zobrazit je v grafu. A jak je to konkrétně řešené v aplikaci PreSpe? Jak již zmiňovala kapitola o vytváření obrázků, v kódu stránky zavoláme obrázek (výsledný graf) použitím klasického tagu ``. Jako parametr `src` je uveden soubor `graph.php`, který se nachází v adresáři `jpgraph`.

Nyní musím malinko odbočit a blíže seznámit se souborem `graph.php`. Tento soubor je jakousi „hlavou“ pro vytváření grafů. Veškeré grafy, které je možno v aplikaci PreSpe spatřit, jsou zobrazovány či ukládány právě tímto souborem. Aby však věděl, jaký graf má zobrazit, je nutné předat parametr s názvem `mn`. Tento soubor však sám o sobě graf nevytvoří. Tvorbu grafů obstarávají funkce v souboru `graph_gen.php`.

Jestli-že byl soubor `graph.php` „hlavou“, pak soubor `graph_gen.php` je jejím „mozkem“. Právě ona vytvoří pomocí svých funkcí se zadanými

²¹ Dále se budu zmiňovat jen o subspektrech, ve skutečnosti však lze určit, zda se má zobrazovat i spektrum

parametry graf a vrátí ho souboru *graph.php* k dalšímu použití. Základní funkcí tohoto souboru je funkce *vytvor_graf(\$mn, \$kanaly, \$nacist_zdroje, \$color)*, kde parametr *mn* funkci informuje o typu grafu, jaký má vytvořit. *Kanaly* předávají informace o tom, jaké subspektra se mají zobrazit. Parametr *nacist_zdroje* sděluje funkci, zda je potřeba načíst zdroje dokumentů. Tento parametr je důležitý v případě, že funkci voláme několikrát za sebou a PHP verze 5 odmítá předeklarovat parametry, které již byly deklarovány při předchozím volání funkce (navíc je to i zbytečné, jelikož bychom vlastně parametry předeklarovali na ty samé typy). Poslední parametr udává barvu grafu. Předává se zde celé kladné číslo, které odkazuje na barvu uloženou v poli funkce *vytvor_graf()*.

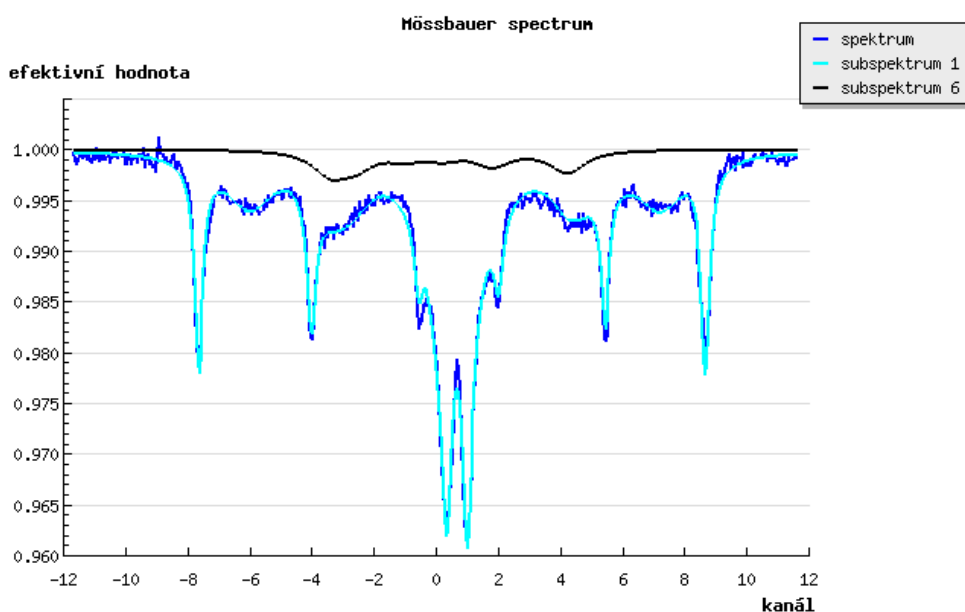
A zpět k řešenému problému zobrazení vybraných subspekter. Nyní už víme, komu předávat informace, ale zatím nevíme jaké. V tomto případě se pro zobrazení vybraných subspekter předává souboru *graph.php* parametr *mn* s hodnotou *choose*. Navíc k němu připojíme soupis subspekter, které mají být zobrazeny, oddělené znakem *x*.

Tento soubor následně zavolá funkci pro vytvoření grafu *vytvor_graf('choose', \$subspektra, 1, 0)*, která se nachází v onom „mozku“. Zde funkce ví, že byl požadavek vytvořit graf, který zobrazí pouze vybraná subspektra uvedené v parametru *\$subspektra*. Zdroje bude načítat (předána jednička), jelikož se jedná o první volání funkce. Barva je nastavena na 0, což odpovídá první položce v poli barev, konkrétně barvě modré. Nutno podotknout, že modrou barvou se však vykreslí pouze první subspektrum, všechny ostatní inkrementují číslo barvy, takže co subspektrum, to jiná barva.

Jelikož jsme funkci *vytvor_graf()* předali seznam subspekter, kde jednotlivé položky byly oddělené znakem *x*, musí funkce obsahovat

dekodér, který získá původní seznam subspekter. Z databáze jsou pak načteny jen tyto subspektra.

Funkce `vytvor_graf('choose', $subspektra,1,0)` tedy vytvoří graf a vrátí ho souboru `graph.php`, který ho následně zobrazí v prohlížeči. Uživatel aplikace si obrázek může uložit použitím pravého tlačítka myši a volby „Uložit obrázek...“²². Možný výsledný obrázek grafu se navolenými subspektry ukazuje Obrázek 8.



Obrázek 8 - Výsledný graf zobrazující navolená subspektra

V zobrazeném grafu pak uživatel může ještě ovlivnit pořadí zobrazovaných subspekter. Postupně může přesouvat jednotlivá subspektra směrem dopředu pomocí formuláře pro změnu pořadí (Obrázek 9).

Změna pořadí:

Posun vpřed:

subspektrum 1
 spektrum

Výsledné zobrazení

Obrázek 9 - formulář pro změnu pořadí zobrazení subspekter

²² V jednotlivých prohlížečích může být název této volby trochu odlišný

Princip posunutí spočívá ve vyhledání pozice subspektra (které má být přesunuto vpřed) v seznamu zobrazovaných spekter. Toto subspektrum je pak posunuto o jednu pozici směrem ke konci seznamu, čímž docílíme pozdějšího načtení z databáze a tím pádem vystoupení subspektra do popředí.

4.4.2. Jednotlivá spektra

Druhou možností je zobrazení jednotlivých subspekter pod sebou v jenom obrázku. Je vlastně vytvořen multigraf, který obsahuje grafy jednotlivých subspekter zvlášť.

Jak už je zvykem, i v tomto případě uvedu jako parametr *src* v obrázku soubor *graph.php*, tentokrát ovšem s hodnotu *in* parametru *mn*. Soubor *graph.php* pak zavolá funkci *vytvor_graf('in',1,0)*. Zde už asi není potřeba parametry komentovat, odlišnost zde nastává v parametru *in*. Ten funkci říká, aby vytvořila jeden graf obsahující grafy subspekter.

Funkce *vytvor_graf* nejprve vytvoří multigraf. Následně prochází databázi a zjistí maximální a minimální hodnotu subspekter. Ty jsou důležité pro nastavení rozsahu osy *y*. Následně se každému subspektru vytvoří graf o stejném rozsahu osy *y*, avšak jiné barvy. Vytvořené grafy jsou postupně přidávány do multigraf a po posledním vloženém grafu je multigraf vrácen souboru *graph.php*. Ten graf zobrazí v prohlížeči. Výsledný graf si opět uživatel může uložit použitím pravého tlačítka myši.

4.5. Protokoly o měření

Aplikace PreSpe vytváří také protokoly o měření, které poskytují základní informace o výsledcích v podobě grafů. Aplikace tyto protokoly generuje ve dvou formátech – ve formátu PDF a HTML. Data jsou opět čerpána z databáze a grafy jsou vytvářeny na základě těchto dat, což zaručuje aktuálnost protokolu v době jeho vzniku.

Každý protokol (nezávisle na formátu) obsahuje informace o měření²³ a grafické zobrazení výsledků. Grafické zobrazení se skládá z obrázku, který představuje spektrum a subspektra v jednom grafu, a grafů, která zobrazují jednotlivá subspektra zvlášť.

A jakým způsobem jsou jednotlivé formáty protokolů vytvářeny? Jen jednou poznámkou se zmíním o jednodušším formátu – formátu HTML. Pro protokol je vygenerována vlastní stránka, která poskládá obsah protokolu podle popsaného obsahu v předchozím odstavci. Obrázky v protokolu jsou vytvářeny stejně jako v aplikaci pomocí volání funkcí souboru *graph_gen.php*, takže nenarazíme na žádný problém²⁴.

4.5.1. Protokoly o měření ve formátu PDF

V případě protokolu ve formátu PDF je to již trochu komplikovanější. Díky kapitole 3.4 Tvorba PDF dokumentů pomocí PHP by již neměl být problém vytvořit dokument PDF. Neměl by být problém ani načíst

²³ Informacemi o měření je myšleno název a datum měření a jeho stručný popis

²⁴ Zde narazíme však na ten důvod, proč je ve funkci *vytvor_graf()* zaveden parametr pro načtení zdroje. Při generování grafů pro jednotlivá subspektra už načítat zdroje nebudeme, protože jsou již načteny při vytváření grafu pro všechna subspektra.

informace o měření z databáze a zobrazit je v tomto dokumentu. Problém ale nastává ve chvíli, kdy chci do protokolu vložit vygenerovaný obrázek.

Knihovna FPDF totiž v případě, že je jako adresa obrázku uveden odkaz na soubor *graph.php*, ohlásí chybu. Ta informuje o tom, že formát obrázku není podporovaný touto knihovnou, přestože ve funkci pro vložení obrázku *Image()* je uveden jako předposlední parametr správný formát. Jako adresu je nutné vždy uvést skutečný soubor obrázku, tedy ne soubor obsahující kódy, které vedou k jeho vytvoření.

Je tedy potřebné vydat se jinou cestou. V aplikaci jsem to vyřešil vytvořením speciálního adresáře *tmp*, do kterého jsou nejprve vygenerovány obrázky, a pak je skript přesměrován na soubor, který vygeneruje PDF protokol. Do něj jsou vloženy obrázky (grafy) z tohoto adresáře.

Abych nemusel vytvářet ještě jeden skript pro smazání obrázků z adresáře *tmp*, přidal jsem funkci pro smazání celého obsahu adresáře ještě před vytvořením nových grafů (tzn., že všechny obrázky v adresáři zůstanou až do příštího generování protokolu ve formátu PDF). Po požadavku na vygenerování je tedy nejprve smazán celý adresář a teprve potom vytvářeny nové grafy. Postup celé tvorby protokolů PDF lze shrnout asi takto:

- Požadavek na vygenerování protokolu ve formátu PDF
- Smazání adresáře *tmp*
- Na základě dat uložených v databázi jsou vygenerovány obrázky a uloženy do adresáře *tmp*
- Přesměrování na soubor generující PDF
- Vygenerování PDF dokumentu s obrázky z adresáře *tmp*
- Zobrazení dokumentu PDF

4.6.Export dat do CSV dokumentů

A dostávám se k poslední funkci aplikace a tou je export dat do CSV dokumentů. Celý postup je vlastně vylíčen již v kapitole 3.5 Tvorba CSV dokumentů pomocí PHP. Jedná se o přesný postup vytvoření dokumentu CSV, který obsahuje data o měření vygenerovaná z databáze.

Zde alespoň popíšu uživatelskou část, tedy to, kam se vlastně vytvořený dokument uloží a jakým způsobem může uživatel zadat jméno souboru.

Uživatel si může vybrat, zda si chce vytvořený dokument uložit pod vlastním názvem, který zadá do formuláře (Obrázek 10), nebo využít defaultní název *protokol.csv*.

Export dat do CVS dokumentu

Export dat do csv

Data budou uložena do adresáře *zpravy*. Pokud nebude název zadán, bude použit implicitní název *protokol.csv*

Název souboru (bez koncovky *csv*):

Obrázek 10 - export dat do CSV dokumentů

Po zadání názvu a kliknutí na tlačítko *ULOŽ* je stránka přesměrována na soubor *vystupy_soubory/csv.php*. Ten obsahuje vlastní kód pro vytvoření csv dokumentu, který je uložen na server. Jelikož je soubor uložen na serveru a k němu nemusí mít (většinou nemají) přístup všichni uživatelé, nabízí Aplikace PreSpe odkaz pro stažení vytvořeného dokumentu.

5. ZÁVĚR

Vytvořená aplikace není komplexním řešením systému pro správu naměřených spekter v laboratořích, mohlo by se však jedna o její část. Je to WWW aplikace a jejím hlavním úkolem je prezentace již naměřených dat, která jsou uložena v databázi. Neřeší tedy ukládání dat do databází, ani neumožňuje jejich správu či modifikaci.

Aplikace umožňuje vygenerování grafů podle požadavku uživatele. Lze vygenerovat graf, který obsahuje uživatelem zadaná subspektra, či multigraf který obsahuje jednotlivá subspektra zvlášť, tedy pro každé subspektrum je vytvořen vlastní graf, který je součástí výsledného multigrafu.

Další možností aplikace je generování protokolů o měření ve formě PDF nebo HTML, které obsahují textové informace o provedeném měření a grafické výsledky v podobě grafů všech a jednotlivých subspekter. Oba protokoly jsou generovány v momentě požadavku, tím je zaručena aktuálnost protokolů.

V neposlední řadě aplikace umožňuje také export dat provedeného měření do CSV dokumentů, které jsou vhodné pro přenos takovýchto dat.

Všechny zobrazované grafy, generované protokoly a exporty jsou vytvářeny na základě dat uložených v databázi. Veškeré výstupy si lze uložit pomocí pravého tlačítka myši a volby „Uložit jako...“.

Hlavní přínos práce vidím v možnosti začlenění této aplikace do komplexnějšího systému týkajícího se měření spekter v laboratoři pro výzkumu nanomateriálu. Díky použití kaskádových stylů lze celkem rychle přizpůsobit i vzhled aplikace systému, do kterého by měla být aplikace začleněna.

SEZNAM POUŽITÉ LITERATURY A CITACÍ

1. **Žák, David.** *Mössbauerovský spektrometr – analytický přístroj pro laboratoře i průmyslové použití.* Disertační práce : Přírodovědecká fakulta Univerzity Palackého Olomouc, 1998.
2. PHP: Hypertext Preprocessor. *PHP: Hypertext Preprocessor.* [Online] 2007. <http://cz.php.net/>.
3. Mössbauer, Rudolf. *Slavní lidé.* [Online] 11. červen 2003. http://www.aldebaran.cz/famous/people/Mossbauer_Rudolf.html.
4. XHTML. *Wikipedie, otevřená encyklopedie.* [Online] 15. duben 2007. <http://cs.wikipedia.org/wiki/XHTML>.
5. World Wide Web Consortium. *Wikipedie, otevřená encyklopedie.* [Online] 7. duben 2007. http://cs.wikipedia.org/wiki/World_Wide_Web_Consortium.
6. **Grausová, Lucie.** *XML pro úplné začátečníky.* 1. vyd. Praha : Computer Press, a.s., 2002. str. 196. ISBN 80-7226-697-7.
7. **Gutmans, A., Bakken, S. S. a Rethans, D.** *Mistrovství v PHP 5.* Brno : Computer Press, a.s., 1. vyd. 2005. str. 656. ISBN 80-251-0799-X.
8. **Hauser, M., Hauser, T. a Wenz, Ch.** *HTML a CSS.* 1. vyd. Brno : Computer Press, a.s., 2006. str. 912. ISBN 80-251-1117-2.
9. PHP: Image Functions. *PHP: Hypertext Preprocessor.* [Online] 2007. <http://cz2.php.net/manual/cs/ref.image.php>.

10. JpGraph. *JpGraph - PHP Graph Creating Library*. [Online] Aditus Consulting, 2007. <http://www.aditus.nu/jpgraph/about.php>.

11. MySQL - Wikipedie, otevřená encyklopedie. *Wikipedie, otevřená encyklopedie*. [Online] 2007. <http://cs.wikipedia.org/wiki/MySQL>.

12. **Matuška, Jan**. Waldova homepage. *Waldova homepage*. [Online] 2007. <http://walda.starhill.org/index.php?kategorie=5&clanek=28>.

13. FPDF. *FPDF Library PDF Generator*. [Online] 2005. <http://www.fpdf.org/>.

ÚDAJE PRO KNIHOVNICKOU DATABÁZI

Název práce	WWW aplikace pro prezentaci naměřených spekter v laboratoři pro výzkum nanomateriálu
Autor práce	Jakub Král
Obor	Informační technologie
Rok obhajoby	2007
Vedoucí práce	RNDr. David Žák, Ph.D.
Anotace	Problematika zobrazování výsledků naměřených spekter v laboratoři pro výzkum nanomateriálu na WWW stránkách. Tvorba a export grafů a obrázků na základě naměřených hodnot do protokolů o měření ve formátu PDF a HTML. Export výsledků do CSV dokumentů.
Klíčová slova	Mössbauerovo spektrum WWW aplikace Export grafů Export obrázků PDF HTML CSV