

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VYŠŠÍ VRSTVY A ROZHRANÍ PRO KNIHOVNU
GENERUJÍCÍ GRAFY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

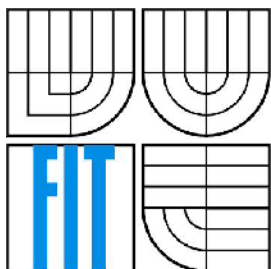
AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ STUDENÝ

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VYŠŠÍ VRSTVY A ROZHRANÍ PRO KNIHOVNU GENERUJÍCÍ GRAFY

HIGH LAYERS AND INTERFACE FOR GRAPH LIBRARY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ STUDENÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROMÍR MARUŠINEC, Ph.D.

BRNO 2009

Vysoké učení technické v Brně - Fakulta informačních technologií

Centrum výpočetních a informačních služeb

Akademický rok 2008/2009

Zadání diplomové práce

Řešitel: **Studený Tomáš, Bc.**

Obor: Informační systémy

Téma: **Vyšší vrstvy a rozhraní pro knihovnu generující grafy**

Kategorie: Uživatelská rozhraní

Pokyny:

1. Seznamte se s knihovnami pro generování grafů na webu a problematikou ukládání obrázků do cache paměti.
2. Navrhněte, implementujte a otestujte rozšíření některé z těchto knihoven o plně a polo automatické generování grafů.
3. Rozšíření testujte pomocí ukázkových příkladů a použijte je pro generování grafů v příslušných modulech na Portálu VUT.
4. Vytvořte modulu pro software MediaWiki (mediawiki.org), který bude toto rozšíření používat.
5. Zhodnoťte výsledky vaší práce a navrhněte možnosti dalšího vylepšení.

Literatura:

- Mistrovství v PHP 5, Andi Gutmans a kol.
- AJAX a PHP, Cristian Darie, Bogdan Brinzarea, Filip Chereches-Tosa, Mihai Bucica
- <http://www.mediawiki.org>
- Přehled statistických metod zpracování, Hendl Jan, Portál, s.r.o. 2006

Při obhajobě semestrální části diplomového projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

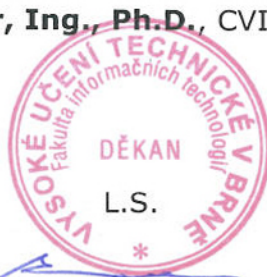
Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Marušinec Jaromír, Ing., Ph.D., CVIS VUT**

Datum zadání: 22. září 2008

Datum odevzdání: 26. května 2009



doc. Ing. Jaroslav Zendulka, CSc.
děkan FIT

Abstrakt

Tato práce se zabývá problematikou tvorby grafů pro použití na webových stránkách. Cílem projektu je vytvoření jednoduchého a přehledného uživatelského prostředí, které zjednoduší práci s knihovnou pChart pro tvorbu grafů. K vytvoření grafu pak není třeba znát syntaxi knihovny pChart, ale vše je možné upravovat přes formulář ve www prohlížeči pomocí myši.

Klíčová slova

web, www, grafy, sloupcový graf, spojnicový graf, výsečový graf, pChart, generování grafů, php, ajax, modul pro mediawiki

Abstract

This work deals with problems of easy web graph generate. Project implement user-friendly interface for Object-Oriented Graph creating library – pChart.

Keywords

web pages, web, web presentation, www, graph, bar graph, line graph, pChart, php, ajax, plug-in for mediawiki

Citace

Bc. Studený Tomáš: Vyšší vrstvy a rozhraní pro knihovnu generující grafy. Brno, 2009, diplomová práce, FIT VUT v Brně.

Vyšší vrstvy a rozhraní pro knihovnu generující grafy

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana inženýra Michala Jurosze. Uvedl jsem všechny literární prameny, publikace a internetové zdroje, ze kterých jsem čerpal.

.....
Tomáš Studený
12.05.2009

© Tomáš Studený, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod.....	3
2 Použité nástroje.....	4
3 Nástroje pro generování grafů.....	5
3.1 GraPHPite.....	5
3.2 Chart.....	5
3.3 ChartDirector.....	6
3.4 PostGraph.....	6
3.5 PHPlot	7
3.6 JGraph	7
3.6.1 Instalace	9
3.6.2 Použití knihovny JGraph	11
4 Knihovna pChart.....	13
4.1 Instalace knihovny pChart.....	14
4.2 První graf pomocí pChart.....	14
4.3 pCache – vyrovnávací paměť.....	16
5 Grafické rozhraní ChartUI	19
5.1 Instalace ChartUI.....	20
5.2 Vytváření grafů pomocí ChartUI	20
5.2.1 Bodový graf	22
5.2.2 Spojnicový graf (úsečkový a křivkový).....	23
5.2.3 Sloupcový graf.....	25
5.2.4 Grafy limit (sloupce a oblasti)	27
5.2.5 XY, Scatter.....	28
5.2.6 Koláčový graf (2D a 3D)	29
5.2.7 Radar graf	30
5.2.8 Kombinace několika typů grafu.....	32
5.3 Automatika v ChartUI.....	34
5.4 Použití ChartUI v jiných projektech.....	34
6 Zásuvný modul pro MediaWiki	36
6.1 MediaWiki.....	36
6.2 Zásuvné moduly do MediaWiki.....	37
6.3 Vložení grafu do WikiTextu	37
7 Fonty a čeština	43

7.1	TrueType fonty.....	43
7.1.1	Aplikace TrueFontEditor	43
7.2	OpenType fonty.....	45
8	Závěr	46
	Příloha A.....	50
	Seznam příloh	51

1 Úvod

V dnešní době jsou grafy nedílnou součástí nejen firemních intranetů a webových aplikací, ale můžeme je nalézt i na jednoduchých www prezentacích.

Graf je velice užitečný nástroj, jak jednoduše, přehledně a výstižně zobrazit číselné údaje. Pro zobrazení různých vstupních dat se hodí různé typy grafů. Asi nepoužívanějšími jsou sloupcový, spojnicový, plošný a v neposlední řadě výsečový graf (někdy nazýván i jako koláčový).

Existují různé knihovny pro generování grafů na webu. Liší se především ve kvalitě výstupních obrazů a v možnostech nastavení. Jedno však mají společné: pro vytvoření grafu je třeba ve většině z nich „naprogramovat“ výsledný obrázek pomocí nepřehledného množství funkcí a parametrů.

Tato diplomová práce se snaží vytvořit přívětivé a srozumitelné rozhraní mezi běžným uživatelem a generující knihovnou, bez potřeby znalostí jednotlivých funkcí a schopnosti napsat výsledný skript, který graf vygeneruje. Zmíněné rozhraní je vytvořeno formou xHTML formulářů, pomocí kterých uživatel, za použití roletových menu, textových a zaškrťovacích políček, dotvoří automaticky navržený graf.

Na začátku práce bude uvedeno několik projektů, zabývajících se generováním grafů. Jedná se o více či méně rozsáhle projekty napsané ve skriptovacím jazyce PHP. U každého z nich budou popsány klady a zápory dané knihovny.

Samotné rozhraní bylo založeno na knihovně projektu pChart, na kterou bude zaměřena větší část této práce. Budou ukázány použité nástroje, uvedeny užitečné funkce knihovny a předvedeno vytvořené rozhraní pro generování grafů.

Závěr práce obsahuje shrnutí dosažených výsledků, zhodnocení výsledné aplikace a nástin dalších možných úprav a funkcí, které by z vytvořeného rozhraní udělalo mocnější nástroj přidáním dalších funkcí.

2 Použité nástroje

Samotná aplikace je napsána za použití server-side jazyka PHP¹, doplněného jazykem JavaScript². Formuláře, pomocí kterých se grafy upravují, jsou vytvořeny validním xhtml kódem. Pro design byly použity kaskádové styly. Díky tomu je možné poměrně jednoduše změnit vzhled celé aplikace (pouhou výměnou souboru s definicí CSS³).

Odesílání dat z formulářů na server se provádí pomocí JavaScriptu, resp. AJAX. Pro některé grafické prvky uživatelského rozhraní bylo s výhodou použito knihovny jQuery [jquery] (jedná se například o barevnou paletu určenou pro volbu barvy namísto zadávání hodnoty do textového políčka v hodnotách rgb).

System ke svému běhu vyžaduje nainstalovanou a funkční knihovnu GD2 určenou pro práci s pokročilejší grafikou v PHP. Je možné pracovat i s GD1, ale některé funkce pak nejsou dostupné. Samozřejmě je i nainstalovaná a zprovozněná knihovna pChart.

Samotná aplikace je pak přístupná přes běžný www prohlížeč, ve kterém uživatel zadá vstupní data grafu a poté upravuje design grafu za pomoci myši.

Pro provoz nebo pouze testování na lokálním počítači, je možné s výhodou využít různé předpřipravené balíčky obsahující instalaci Apache, PHP. Příkladem může být populární triáda Apache Friends – XAMPP [xampp], která byla využita při vývoji aplikace. Vytvořené grafické rozhraní knihovny pChart, však samozřejmě funguje na kterémkoliv www serveru, který podporuje PHP skripty.

¹ **PHP** (*Hypertext Preprocessor*) – Open Source skriptovací jazyk pro tvorbu webu, interpretace kódu se provádí na straně serveru. Zdrojový kód se ke klientovi, při bezchybném chodu www serveru, nedostane.

² **JavaScript** – programovací jazyk interpretovaný na straně klienta (v prohlížeči). Kód programu se zapisuje přímo do webové stránky a je přímo viditelný.

³ **CSS** (*Cascading Style Sheets*) – Nástroj pro centrální správu designu stránek z roku 1997, vyvinutý konsorciem W3C (*World Wide Web Consortium*), které se stará o podporu standardizace služeb v oblasti www stránek.

3 Nástroje pro generování grafů

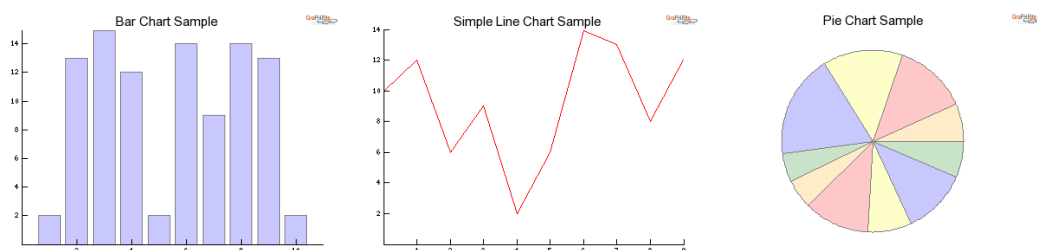
V dnešní době je možné nalézt nepřeberné množství různých nástrojů a rozšiřujících knihoven pro vytváření velice profesionálně vypadajících grafů k použití v oblasti webu.

Jako nejlepší byl nakonec vybrán projekt pChart. Nicméně v následujících podkapitolách budou stručně představeny některé další.

3.1 GraHPite

Jedná se o projekt napsaný v PHP pomocí objektově orientovaného modelu. Podporuje jak verzi 4 tak i verzi 5 a využívá grafické knihovny GD2. Podporována je i GD1, ale obecně je lepší využít GD2 pro její širší možnosti a lépe vypadající výstup. GraHPite je volně šiřitelný software pod licencí GNU GPL. Jeho předností je velké množství různých typů grafů (včetně základních jako sloupcový, spojnicový apod.). Zajímavostí je i použití „pojmenovaných barev“, kdy je možné barvy zadávat pomocí více jak 150 předdefinovaných názvů. Celá knihovna je díky zvolenému OO modelu přehledná a logicky rozčleněna na jednotlivé části. [graphpите]

Následují ukázky základních typů grafů.

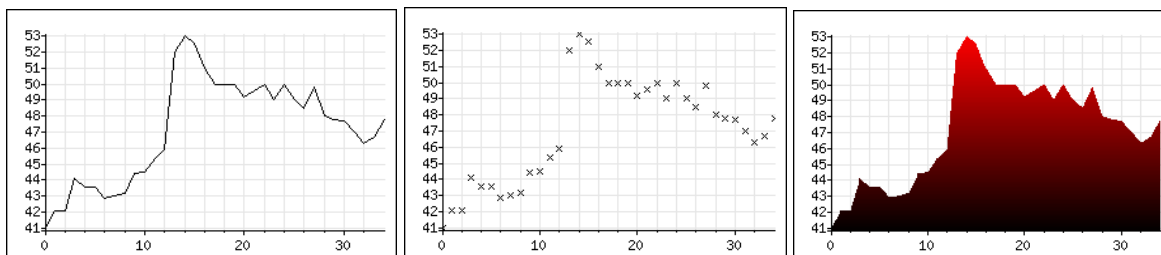


Obrázek č. 1 - Ukázky sloupcového, spojnicového a koláčového grafu

3.2 Chart

Další projekt zaměřený na generování grafů pomocí PHP tentokrát podporující pouze PHP verze 4. šířený, obdobně jako předchozí pod licencí GNU GPL. Je vhodný pro velice jednoduché vytvoření grafů (graf je možné vytvořit již pomocí 3 řádků kódu). Výsledný vzhled grafu je možné měnit a upravovat. Jedná se však o velice jednoduché grafy se základním vzhledem. [chart]

Více naznačí ukázky.

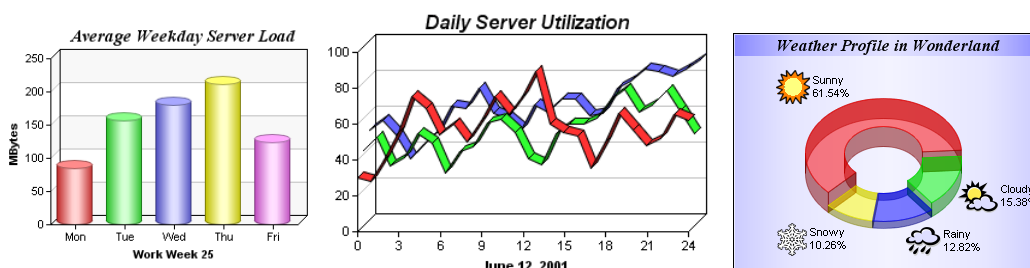


Obrázek č. 2 – Ukázky spojnicového, bodového a plošného grafu

3.3 ChartDirector

Komerční produkt dostupný pro více jak 8 různých programovacích jazyků včetně PHP verze 4 a 5. Jedná se o samostatné dll knihovny. ChartDirector dovoluje vytvářet velice zajímavé a profesionálně vypadající grafy. Díky použité technologii AJAX umí výsledné grafy interaktivně reagovat na podněty uživatele (např. zobrazení hodnoty pod kurzorem myši, zobrazení poznámky při kliknutí na část grafu apod.) [chartdirector]

Opět následuje několik ukázek.

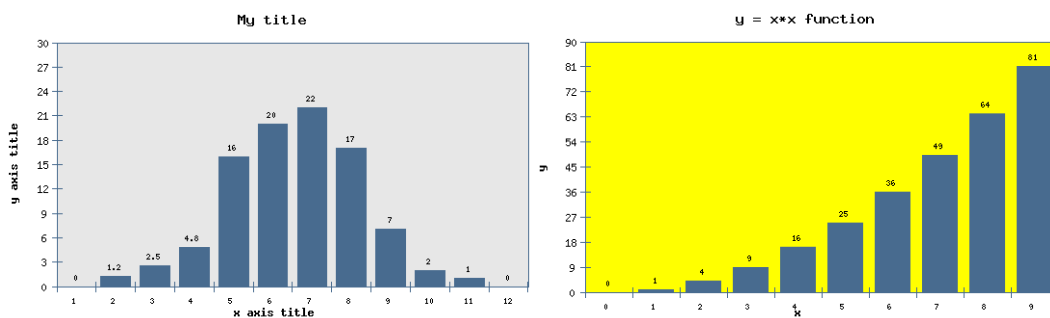


Obrázek č. 3 – Ukázky sloupcového, spojnicového a koláčového grafu

3.4 PostGraph

Velice jednoduchá, volně šiřitelná (licence GNU GPL) PHP knihovna pro tvorbu sloupcových grafů. PostGraph funguje s PHP verzemi 4 i 5 a využívá standardní rozšiřující GD knihovny. [postgraph]

Příklad naznačuje, jak může vypadat výsledný graf.



Obrázek č. 4 – Ukázky sloupcového grafu

3.5 PHPlot

Poslední příklad grafické knihovny pro PHP verze 3 nebo 4 používající rozšíření GD. PHPlot podporuje TTF fonty, vložení obrázku na pozadí grafu, vytisknutí více grafů do jednoho výstupu apod. Výsledné formáty jsou pak gif, wbmp, png a jpeg. Mezi podporované typy grafů patří například sloupcový, spojnicový, koláčový nebo plošný graf. [phplot]

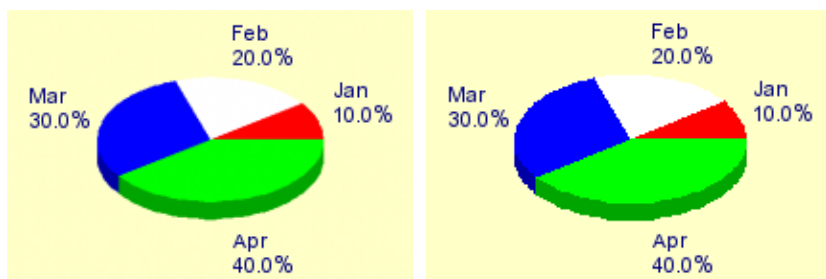
3.6 JPGraph

Na tuto knihovnu byla zaměřena semestrální práce. Důvodů přechodu na pChart bylo hned několik. Prvním bylo složité nastavení licenční politiky, které omezovalo širší použití této knihovny v jiných projektech a omezení na nekomerční projekty. Dalším důvodem bylo finální zobrazení obrázků grafu. Při srovnání grafického výstupu knihovny pChart a JPGraph, je zřejmé, že knihovna pChart mnohem lépe pracuje s anti-aliasingem, tj. výsledné grafy mají hladké hrany a okraje.

Celá knihovna je napsána za pomoci objektově orientované programovací techniky a je možné ji provozovat na PHP verze 4 i 5. Obsahuje více než 100 tříd s celkem něco málo přes 1060 metod.

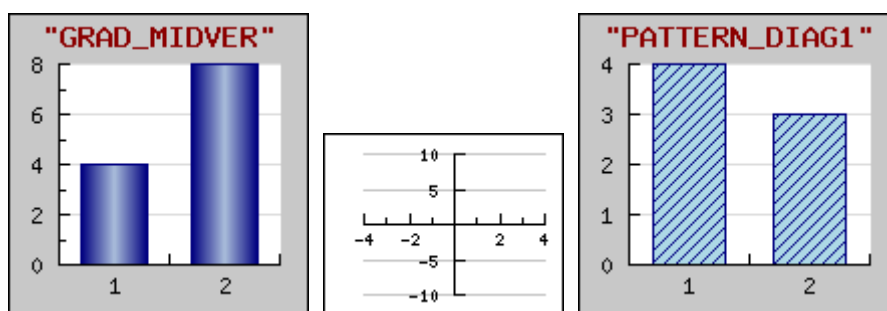
K dispozici je nepřehledné množství 2D i 3D grafů. Mimo jiné knihovna zvládá sloupcový, spojnicový, koláčový, polární a jiné grafy. Navíc je možné generovat obrázky s tabulkami, bezpečnostní kódy proti spamu, čárové kódy, digitální LED zobrazovače. Implementovány jsou i vlajky přes 200 zemí, které je možné použít při vytváření grafů.

Podporován je i tzv. anti-aliasing, který je patrný zvláště u koláčových grafů. Graf s vypnutým anti-aliasingem má viditelně více „zubaté“ hrany.



Obrázek č. 5 – Koláčový graf s a bez použitého anti-aliasingu

Plně nastavitelná je i pozice jednotlivých os grafu. Oblasti grafu mohou být vyšrafovány nebo na ně může být aplikován přechod dvou barev s možností výběru směru světla



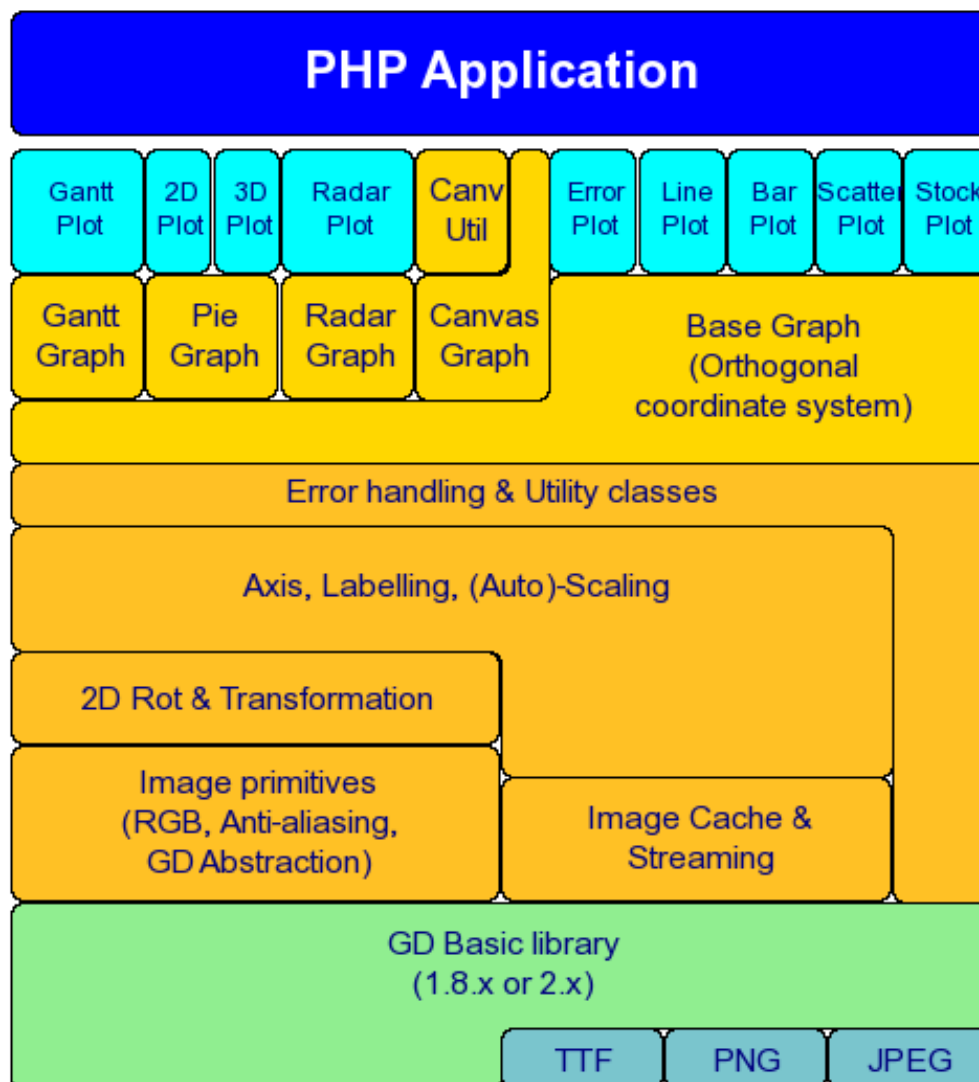
Obrázek č. 6 – Ukázky možností výplně a umístění os

JPGraph nabízí dva licenční modely. První – standard edition je určený pro nekomerční použití, tj. osobní použití, školství a všude tam, kde není s použitím JPGraph spojen finanční zisk. Druhý – professional version je určený pro komerční využití. Pro verze přináší i několik přídatných funkcí. Příkladem může být generování čárových kódů, jak lineárních, tak i 2D.



Obrázek č. 7 – Příklady čárových kódů PRO verze JPGraph

Architektura celé aplikace JpGraph je patrná z následujícího schémata.



3.6.1 Instalace

Instalace samotné knihovny JpGraph není složitá. Stačí přesunout obsah adresáře src z distribučního balíčku na webový server podporující skriptovací jazyk PHP. Nezbytnou podmínkou pro funkci JpGraph knihovny je aktivní, funkční knihovna GD. To, zda webový server toto rozšíření obsahuje, je možno otestovat jednoduchým skriptem:

```

$im = @ ImageCreate ( 150, 100)
    or die ( "Obrazek nelze vytvorit. Knihovna GD neni funkcní!");
$background_color = ImageColorAllocate ($im, 255, 255, 255);
$text_color = ImageColorAllocate ($im, 233, 14, 91);
ImageString ($im, 1, 5, 5, "Testovací text ...", $text_color );
header ("Content-type: image/png");
ImagePng ($im);

```

Jestliže je výsledkem tohoto skriptu obrázek s černým a bílým obdélníkem, je vše v pořádku. V opačném případě je třeba přezkontrolovat nastavení knihoven v konfiguračním souboru php.ini.

Veškeré pokročilé funkce JpGraph je však možné využít při použití GD knihovny verze 2. To, zda ji aktuální instalace PHP podporuje, je opět možné zjistit jednoduchým skriptem:

```

$im = ImageCreateTrueColor ( 300, 200);
$black = ImageColorAllocate ($im, 0, 0, 0 );
$white = ImageColorAllocate ($im, 255, 255, 255 );

ImageFilledRectangle ($im,0, 0,399,99 ,$white);
ImageRectangle ($im,20, 20,250,190 ,$black);

header ("Content-type: image/png" );
imagepng ($im);

```

Nastavení chování JpGraph knihovny se provádí pomocí konfiguračního souboru jpg-config.inc.php v kořenovém adresáři JpGraph. Přednastavené hodnoty jsou ve většině případů pro běžné používání dostatečné, takže je není třeba měnit.

3.6.2 Použití knihovny JGraph

Samotný skript, který generuje graf, je vlastně posloupností funkcí (metod), kterými definujeme jednotlivé vlastnosti grafu.

Na začátku skriptu je třeba vždy připojit soubor *jgraph.php* pomocí PHP funkce *include*. Stejně tak je třeba připojit další soubor, a to v závislosti na typu grafu. V níže uvedeném příkladu na vytvoření sloupcového grafu je to soubor *jgraph_bar.php*.

Inicializace nového grafu se provádí metodou *graph()*, které se jako parametr předává výsledná velikost obrázku s grafem.

```
$graph = new Graph(300,200);
```

Dalším krokem je nastavení okrajů a automatická velikost obsahu grafu. Také je možné zakázat rámeček kolem grafu, resp. výsledného obrázku.

```
$graph->SetScale("textlin");  
$graph->SetMarginColor('white');  
$graph->SetMargin(30,1,20,20);  
$graph->SetBox();  
$graph->SetFrame(false);
```

Nastavení a zobrazení mřížky grafu (pozadí) v rovině X (*xgrid*) a v rovině Y (*ygrid*) může vypadat například následovně.

```
$graph->ygrid->SetFill(true,'#DDDDDD@0.5','#BBBBBB@0.5');  
$graph->ygrid->SetLineStyle('dashed');  
$graph->ygrid->SetColor('gray');  
$graph->xgrid->Show();  
$graph->xgrid->SetLineStyle('dashed');  
$graph->xgrid->SetColor('gray');
```

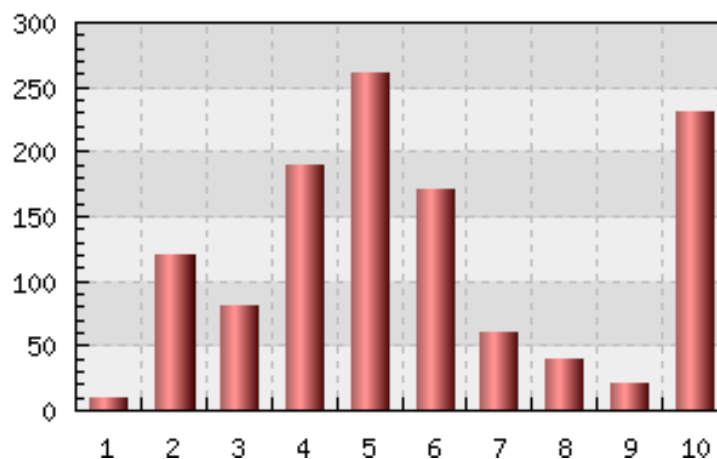

Teď je už možné přidat samotné sloupce. Těch je možné do grafu přidat několik (různé datové řady). Hodnoty pro vynesení do grafu se předpokládají v proměnné *data1*. Pomocí metody `add()` se pak sloupce přidávají do výsledného grafu.

```
$bplot = new BarPlot($data1);  
$bplot->SetWidth(0.6);  
$bplot->SetFillGradient('#440000', '#FF9090', GRAD_LEFT_REFLECTION);  
$bplot->SetWeight(0);  
$graph->Add($bplot);
```

Nakonec je vše odesláno zpět do `www` prohlížeče, ve kterém je zobrazen finální graf.

```
$graph->Stroke();
```

Výsledkem tohoto krátkého příkladu je pak následující sloupcový graf.



Obrázek č. 8 – Výsledek ukázkového grafu

4 Knihovna pChart

Jedná se o knihovnu napsanou v jazyce PHP pomocí objektového programování, která je určena pro jednoduché generování velice efektně vypadajících grafů. Licenční politika tvůrců se řídí licenci GNU General Public Licence, což umožňuje tento mocný nástroj implementovat zdarma v jiných projektech.

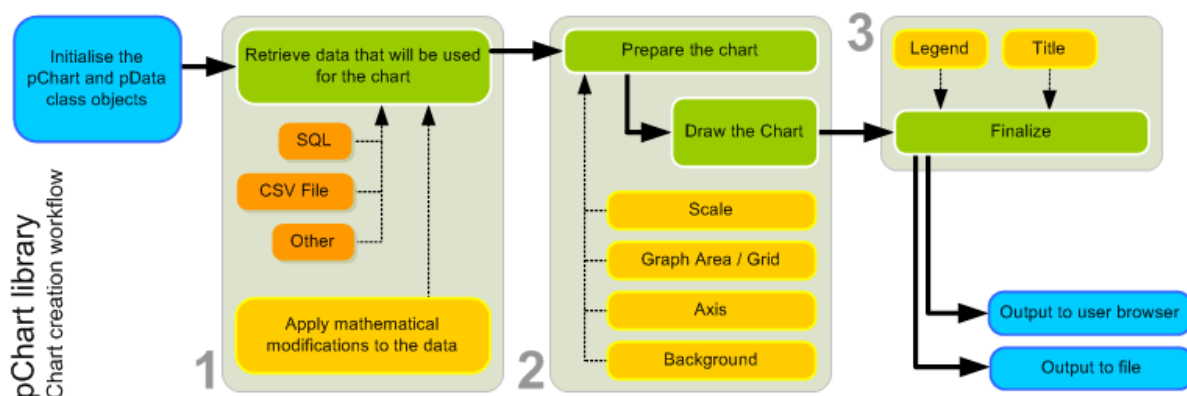
Na výstupní obrázek knihovny je aplikován antialiasing, což zaručuje vyhlazenou grafiku a odstraňuje „zubaté“ hrany okrajů grafů a jejich dalších součástí.

Jako vstupní data je možné použít několik zdrojů. Tím prvním je libovolná databáze (např. MySQL / Microsoft SQL apod.). Další možností je dodat data knihovně pomocí CSV souboru (textový soubor s hodnotami oddělenými středníkem). V neposlední řadě je tu možnost zapsat vstupní data přímo do skriptu, který vrací výsledný obrázek.

Stejně tak jako ostatní projekty i pChart potřebuje ke svému běhu na straně serveru nainstalovaný modul GD, který se v PHP stará o vykreslování grafiky.

Neocenitelnou vlastností pChart je implementace funkce pCache, která umožňuje ukládání výsledných obrázků s grafy do speciálního adresáře – vyrovnávací paměti a jejich použití v případě, že knihovna obdrží požadavek na graf, jehož vstupní data nebyla změněna. Tím se šetří výkon serveru, který nemusí graf znovu vytvářet, ale také se snižuje doba odezvy potřebné k odeslání grafu do prohlížeče klienta. Vyrovnávací paměti pCache je věnován jeden z následujících pododstavců.

Knihovna pChart je živý projekt a autoři uvádějí, že úpravy a vylepšení jsou prováděny každý týden. S každou novou verzí dochází i k optimalizaci rychlosti vykreslování.



Obrázek č. 9 – Workflow vytváření grafů knihovnou pChart

4.1 Instalace knihovny pChart

Instalace knihovny není složitá. Jako u většiny PHP knihoven, stačí zkopírovat zdrojové PHP skripty do složky, která je přístupná pro www server, který je poté interpretuje.

Do skriptů, kde se má graf zobrazit, se pouze nalinkují dva soubory, a to *pData.class* a *pChart.class*. K tomu je možné využít např. funkci `include()`.

```
include("pChart/pData.class");  
include("pChart/pChart.class");
```

Dále je třeba povolit používání knihovny GD v nastavení PHP a knihovna pChart je připravena k použití.

4.2 První graf pomocí pChart

Poté co jsou metody pChart přístupné, je možné začít vytvářet graf. Zdrojový kód grafu je posloupnost metod třídy pChart, které postupně přidávají jednotlivé části grafu do obrázku. Tato posloupnost je pak zakončena metodou *render()*, která odešle obrázek grafu do prohlížeče.

Na začátku je třeba nastavit datové série grafu.

```
$DataSet = new pData;  
$DataSet->AddPoint(array(3,5,3,1,2,4,8), "S1");  
$DataSet->AddPoint(array(5,1,2,4,5,2,1), "S2");  
$DataSet->AddAllSeries();  
$DataSet->SetAbsciseLabelSerie();  
$DataSet->SetSerieName("Leden", "S1");  
$DataSet->SetSerieName("Duben", "S2");
```

Metodou *pData* zinicilizujeme vkládání datových hodnot. Další metoda *AddPoint* přidá jednotlivé série datových bodů a následující metoda *AddAllSeries* všechny série vloží do grafu. Poté dojde k přidání popisků pro nápovědu přes metody *SetSerieName*.

Následuje vykreslení pozadí grafu a vložení os x a y.

```
$obrazek = new pChart(700,230);  
$obrazek->setFontProperties("Fonts/tahoma.ttf",8);  
$obrazek->setGraphArea(50,30,585,200);
```

```

$obrazek->drawScale($DataSet->GetData(), $DataSet->GetDataDescription(),
SCALE_NORMAL, 150, 150, 150, TRUE, 0, 2);
$obrazek->drawGrid(4, TRUE, 230, 230, 230, 50);

```

V tomto kroku vytvoříme obrázek o rozměru 700x230 pomocí metody pChart. Nastavíme font a jeho velikost a do obrázku vykreslíme osy x a y s popisky, které je možné upravovat dle velikosti vstupních dat. Je možné také zapnout mřížku na pozadí grafu pomocí metody drawGrid.

V navazujícím kroku zvolíme samotný typ grafu.

```

$obrazek->drawBarGraph($DataSet->GetData(),
$DataSet->GetDataDescription());

```

V tomto případě byl typ grafu nastaven na sloupcový graf a došlo k vykreslení dat do obrázku.

V posledním kroku bude graf dokončen a vyexportován do PNG souboru.

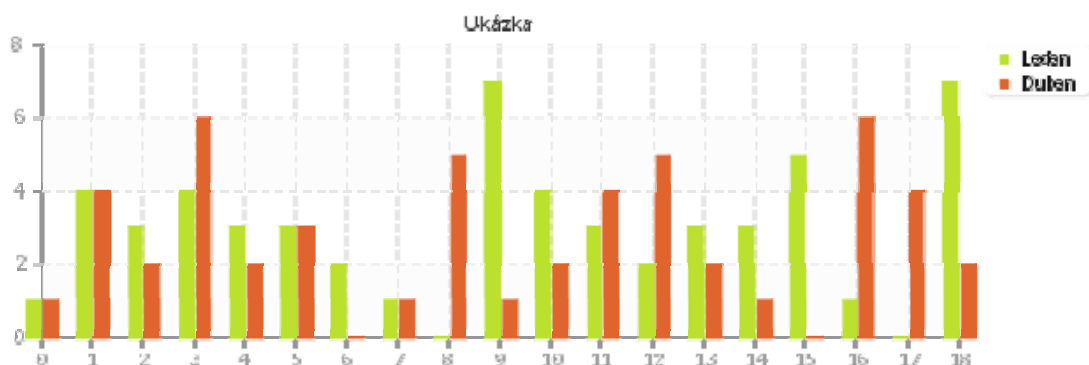
```

$obrazek->setFontProperties("Fonts/tahoma.ttf", 8);
$obrazek->drawLegend(600, 30, $DataSet->GetDataDescription(), 255, 255, 255);
$obrazek->setFontProperties("Fonts/tahoma.ttf", 10);
$obrazek->drawTitle(50, 22, "Ukázka", 50, 50, 50, 585);
$obrazek->Render("graf.png");

```

Nastavení fontu a velikosti písma pro legendu, vykreslení legendy. Dále pak nastavení fontu a velikosti písma pro titulek, vykreslení titulku. Nakonec se pomocí metody tender uloží výsledný obrázek na disk do obrázkového souboru ve formátu PNG.

Výsledkem tohoto stručného, komentovaného kódu je obrázek, zobrazující vstupní hodnoty pomocí sloupcového grafu.



Obrázek č. 10 – výsledný graf

Tento příklad ukazuje pouze základní operace s knihovnou pChart. Knihovna obsahuje desítky metod pro práci a úpravy grafu.

Výše uvedená ukázka znázorňuje jednoduchost vytváření grafů pomocí knihovny pChart. Zároveň však také demonstruje, že pokud bychom chtěli využít veškerých vlastností této knihovny, je třeba nastudovat podrobně dokumentaci této knihovny a umět minimálně základy programování v jazyce PHP.

4.3 pCache – vyrovnávací paměť

Ukládání obrázků do vyrovnávacích pamětí je důležité zejména u webových serverů s velkou návštěvností. Více požadavků na generování složitého obrázku by mohlo mít za následek nedostupnost celého serveru.

Toto se snaží řešit právě vyrovnávací paměti. Obrázek se generuje pouze jednou, při prvním požadavku a zároveň se uloží na disk. Další požadavky jsou již odbavovány z disku, z vyrovnávací paměti.

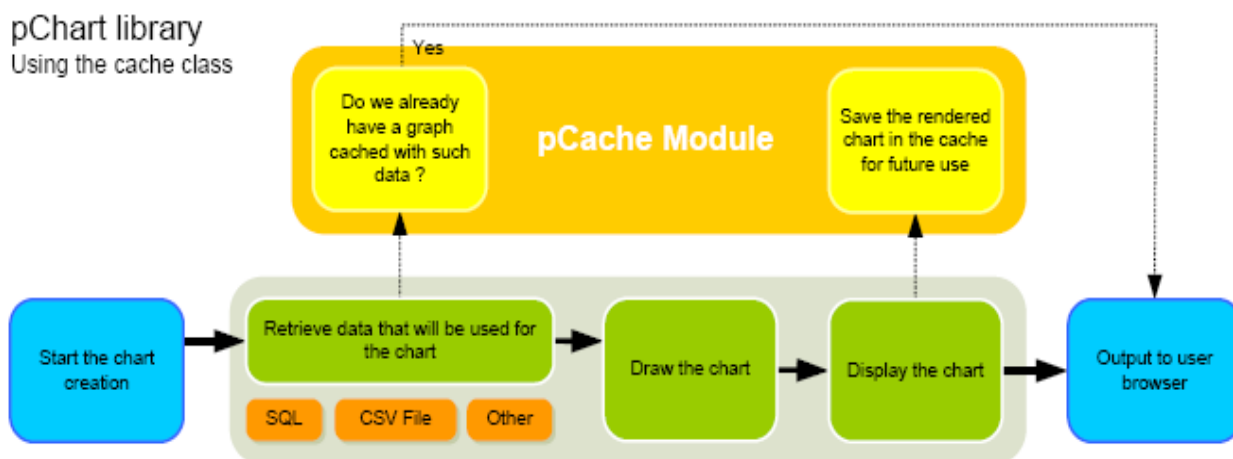
Důležité je správně odhadnout a nastavit životnost obrázku, tj. dobu, za kterou se nezmění, a po kterou je možné jej distribuovat z vyrovnávací paměti. V opačném případě by hrozilo, že si někteří klienti stáhnou již neaktuální data.

Samotný pChart již má implementován systém na ukládání vytvořených grafů do vyrovnávací paměti. Pro zapnutí tohoto systému je třeba nalinkovat knihovnu *pCache*. Tj. k již výše uvedeným dvou knihovnam *pData.class* a *pChart.class* přibude knihovna *pCache.class*. Na začátku skriptu, který graf vytváří je třeba uvést následující:

```
include("pChart/pData.class");  
include("pChart/pChart.class");  
include("pChart/pCache.class");
```

Adresář použitý pro vyrovnávací paměť musí být pro procesy webového serveru přístupný pro zápis. Cesta do adresáře, kam se ukládají obrázky se nastavuje s knihovně *pCache.class*.

Funkce vyrovnávací paměti je patrná z následujícího obrázku (obrázek č. 10).



Obrázek č. 10a – diagram práce s cache v pChart

Zpřístupnění funkce vyrovnávací paměti se provádí metodou `pCache`. Volitelný parametr této metody je cesta do adresáře vyrovnávací paměti. Pokud není zadána, použije se přednastavená hodnota v knihovně `pCache.class` (většinou je to složky `pCache` v rootu `pChart`).

```
$Cache = new pCache();
```

Do skriptu, který generuje graf, je potřeba umístit metodu `WriteToCache`. Tato metoda musí být zavolána ještě před odesláním obrázku s grafem pomocí `stroke()` nebo `render()`.

```
$Cache->WriteToCache("Graph_unique_id", $DataSet->GetData(), $obrazek);
```

Poslední částí systému vyrovnávací paměti je metoda `GetFromCache`, která se volá ještě před inicializací grafu pomocí „`$obrazek = new pChart(700,230)`“.

```
$Cache->GetFromCache("Graph_unique_id", $DataSet->GetData());
```

Při prvním vygenerování obrázku se uloží jeho kopie do adresáře pro vyrovnávací paměť pomocí metody `WriteToCache`. Parametr `Graph_unique_id` jednoznačně identifikuje graf v rámci jednoho skriptu. Při opakovaném požadavku na graf se pomocí druhé metody `GetFromCache` zjistí, zda není obrázek v adresáři pro vyrovnávací paměť. Pokud ano, zastaví se generování grafu a do prohlížeče je odeslán již existující obrázek z vyrovnávací paměti. V opačném případě dojde k opětovnému vygenerování grafu. Zjišťování změn v obrázku je řešeno pomocí hashovací funkce `md5()`.

Krátký skript využívající vyrovnávací paměť může vypadat například takto:

```
/* datové série */
$DataSet = new pData;
$DataSet->AddPoint(array(3,5,3,1,2,4,8), "S1");
$DataSet->AddPoint(array(5,1,2,4,5,2,1), "S2");
$DataSet->AddAllSeries();
$DataSet->SetAbsciseLabelSerie();
$DataSet->SetSerieName("Leden", "S1");
$DataSet->SetSerieName("Duben", "S2");

/* inicializace cache */
$Cache = new pCache();
$Cache->GetFromCache("graf_1", $DataSet->GetData());

/* generování obrazku (pokud nebyl v cache) */
$obrazek = new pChart(700,230);
$obrazek->setFontProperties("Fonts/tahoma.ttf",8);
$obrazek->setGraphArea(50,30,585,200);
$obrazek->drawScale($DataSet->GetData(), $DataSet->GetDataDescription(),
SCALE_NORMAL,150,150,150,TRUE,0,2);
$obrazek->drawGrid(4,TRUE,230,230,230,50);
$obrazek->drawBarGraph($DataSet->GetData(),
$DataSet->GetDataDescription());

$obrazek->setFontProperties("Fonts/tahoma.ttf",8);
$obrazek->drawLegend(600,30,$DataSet->GetDataDescription(),255,255,255);
$obrazek->setFontProperties("Fonts/tahoma.ttf",10);
$obrazek->drawTitle(50,22,"Ukázka",50,50,50,585);

/* zápis obrázku do cache */
$Cache->WriteToCache("graf_1", $DataSet->GetData(), $obrazek);

/* uložení výsledného obrázku */
$obrazek->Render("graf.png");
```

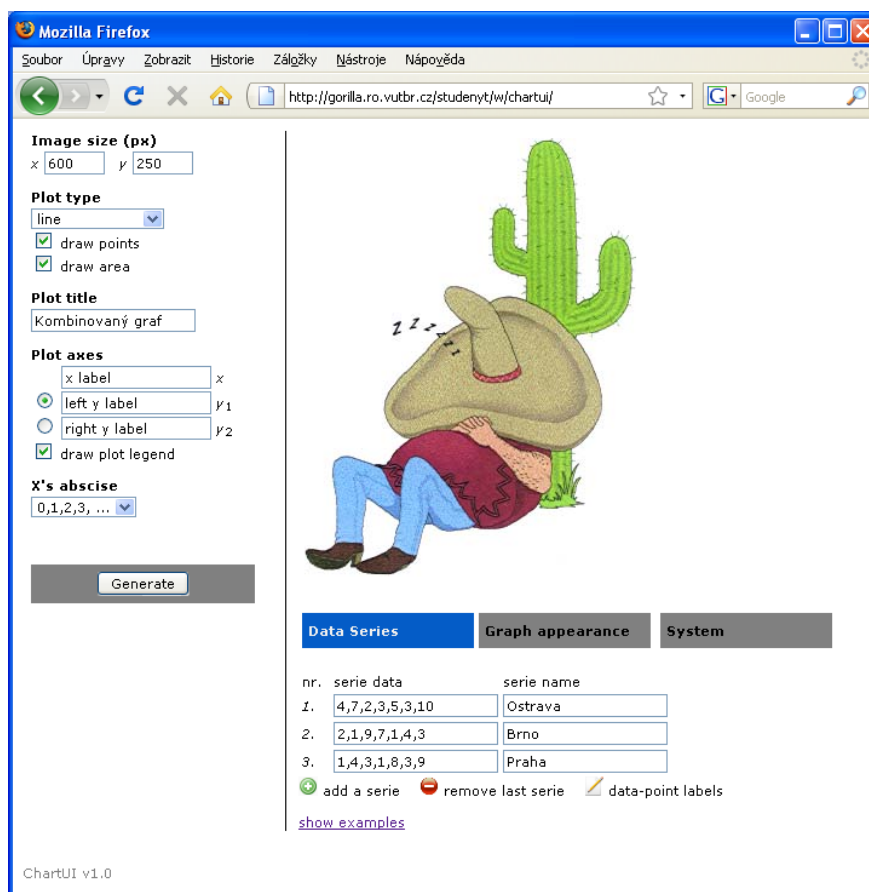
Ne všechny grafy je nutné často obnovovat, některé se mění jednou za den, některé i v delších časových intervalech. Proto je lepší v těchto případech vyrovnávací paměť používat a ušetřit tak procesorový čas a zvýšit výkon a rychlost celého systému.

5 Grafické rozhraní ChartUI

V předchozí kapitole bylo ukázáno, jak se postupuje při vytvoření grafu pomocí knihovny pChart. Programování grafu není složité, nicméně náročné na znalost metod třídy pChart. Stylování grafu pomocí přidávání řádků kódu do skriptu s grafem není příliš komfortní.

Nutné jsou také rámcové znalosti programování v jazyce PHP.

Výše uvedené důvody vedly k vytvoření grafického rozhraní mezi knihovnou pChart a uživatelem. Rozhraní pChartGUI (obrázek č. 11) zobrazuje drtivou většinu nastavení a umožňuje pracovat s grafy typu bodový graf, spojnicový graf (úsečkový i spojnicový), sloupcový graf, xy scatter graf, limitní grafy (úsečkové i plošné), koláčový graf (2D i 3D), radar graf a v neposlední řadě možnost kombinovat několik typů grafu do jednoho obrázku.



Obrázek č. 11 – úvodní stránka ChartUI

Základem vytvoření uživatelského rozhraní bylo důkladné prozkoumání jednotlivých tříd knihovny pChart na úrovni zdrojového kódu. Dále pak zmapování jednotlivých metod a nacvičení jejich používání a kombinování.

Stránka grafického rozhraní je napsána validním xhtml kódem. Pro stylování byly použity kaskádové styly CSS, díky kterým je možné jednoduše změnit grafiku stránky pouhou výměnou, případně úpravou předpisu CSS v souboru grafph.css.

Interaktivní prvky stránky byly řešeny pomocí Java Framework knihovny jQuery, pomocí které bylo možné vyřešit jednotlivé záložky v nastavení grafu nebo například paletu pro výběr barev jednotlivých částí grafu [jquery].

Jako úložiště byl zvolen soubor, resp. sessions z důvodu snadné instalace a nenáročnosti na systém, kde je pChart provozována.

Stránky s rozhraním pro vytváření grafů byly otestovány a optimalizovány pro nejpoužívanější prohlížeče www stránek jako je Microsoft Internet Explorer (verze 5,6,7,8), dále Mozilla FireFox, Gogole Chroma, Opera a v neposlední řadě Apple Safari.

5.1 Instalace ChartUI

Instalace uživatelského rozhraní ChartUI knihovny pChart spočívá v nakopírování všech souborů se zdrojovým kódem tohoto projektu do adresáře na serveru, kde je nainstalován pChart. Například podadresář chartu.

Pak už je grafické rozhraní ChartUI přístupné přes webový prohlížeč a připravené pro vytvoření prvního grafu.

5.2 Vytváření grafů pomocí ChartUI

Na úvodní straně jsou k nalezení základní nastavení, které jsou společné pro všechny typy grafů. Jedná se o rozměr grafu, typ grafu a datové série. K potvrzení změn a jejich provedení slouží tlačítko generovat. Původně se změny projevovaly okamžitě, bez nutnosti potvrzení. Takto odesílané požadavky však neúměrně zatěžovaly server. Ve finální verzi je tedy nutné každou změnu potvrdit.

Hodnoty, shodné pro všechny grafy (levý sloupec okna ChartUI) jsou *velikost výsledného obrázku grafu* (Image size), *volba typu grafu* (plot type), *název grafu* (plot title), *popisky os x a y* (plot axes) a nakonec *volba zobrazení legendy* (draw plot legend).

V pravé horní části se zobrazuje výsledný obrázek a pod ním jsou záložky s dalšími rozšiřujícími vlastnostmi.

První záložka [Obrázek č. 12] obsahuje operace s datovými sériemi (Data Series). Zadávají se zde vstupní data (serie data), oddělená čárkou. Dále je zde popis datové série pro legendu (serie name) a třetí kolonka určuje popisky datových bodů (data point labels).

Data Series	Graph appearance	Advanced plots	System
nr.	serie data	serie name	data point labels
1.	<input type="text" value="1,2,3,4,5,8,4,8,15"/>	<input type="text" value="Teplota"/>	<input type="text" value="1,2,3,4,5,8,4,8,15"/>
2.	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="+ add a serie"/> <input type="button" value="remove last serie"/> <input type="checkbox"/> data-point labels			

Obrázek č. 12

V záložce druhé (Graph appearance) je možné podrobněji nastavovat vzhled (obrázek č. 13) jednotlivých částí grafu. Zde je možné nastavit barvu pozadí obrázku grafu, font, velikost a poloha legendy, formát popisku apod.

Data Series	Graph appearance	Advanced plots	System
Plot palette			
<input type="text" value="tones.txt"/> <input type="button" value="use it!"/>			
Image background			
fill type <input type="text" value="white"/> color <input type="text" value="#dddddd"/> gradient (0-100) <input type="text" value="100"/>			
Graph background			
fill type <input type="text" value="white"/> color <input type="text" value="#dddddd"/> gradient (0-100) <input type="text" value="100"/> stripe <input type="checkbox"/>			
Legend style			
face <input type="text" value="tahoma.ttf"/> size <input type="text" value="8"/> color <input type="text" value="#000000"/> bground <input type="text" value="#dddddd"/> border <input checked="" type="checkbox"/>			
position: x <input type="text" value="5"/> y <input type="text" value="5"/>			
Title style			
face <input type="text" value="tahoma.ttf"/> size <input type="text" value="12"/> color <input type="text" value="#777777"/> drop shadow <input type="checkbox"/>			
Axes style			
face <input type="text" value="tahoma.ttf"/> size <input type="text" value="10"/> color <input type="text" value="#888888"/> label spacing: x <input type="text" value="50"/> yleft <input type="text" value="70"/>			
yright <input type="text" value="70"/>			
X-axis			
format <input type="text" value="number"/> units <input type="text"/> range-min <input type="text" value="0"/> range-max <input type="text" value="0"/> ticks <input type="text" value="5"/>			
Left Y-axis			
format <input type="text" value="number"/> units <input type="text"/> range-min <input type="text" value="0"/> range-max <input type="text" value="0"/> ticks <input type="text" value="5"/>			
Right Y-axis			
format <input type="text" value="number"/> units <input type="text"/> range-min <input type="text" value="0"/> range-max <input type="text" value="0"/> ticks (use value above)			
Reset to defaults			
<input type="button" value="Reset"/>			

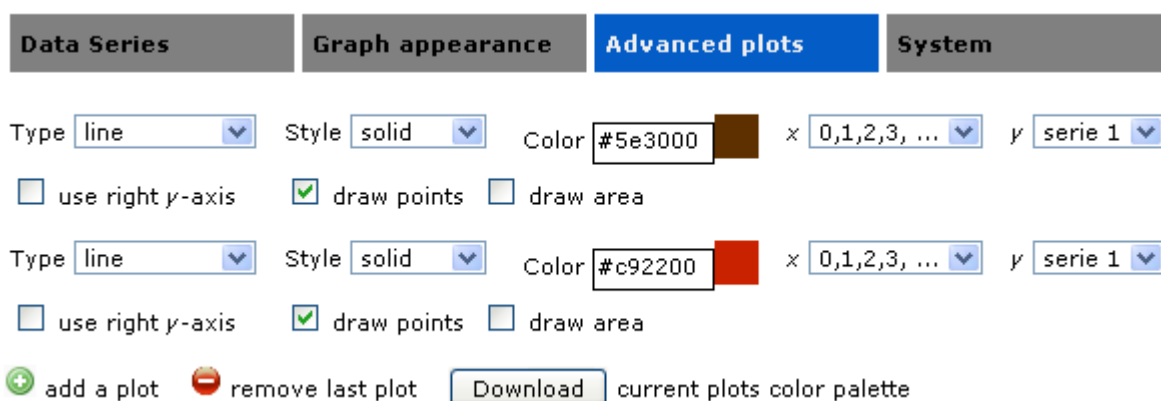
Obrázek č. 13

Pod poslední záložkou [Obrázek č. 14] jsou nastavení aplikace, tj. zapnutí vypisování URL API (viz. dále) a tlačítko pro vymazání vyrovnávací paměti (Reset Cache).



Obrázek č. 14

Speciální typ grafu „mixed plots“, který umožňuje vykreslení různých grafů do jednoho obrázku, obsahuje čtvrtou [Obrázek č. 15] záložku (Advanced plots). Zde se nastavuje barva a typ grafu pro jednotlivé datové série.



Obrázek č. 15

Následují ukázky tvorby jednotlivých typů grafů a jejich stručný popis.

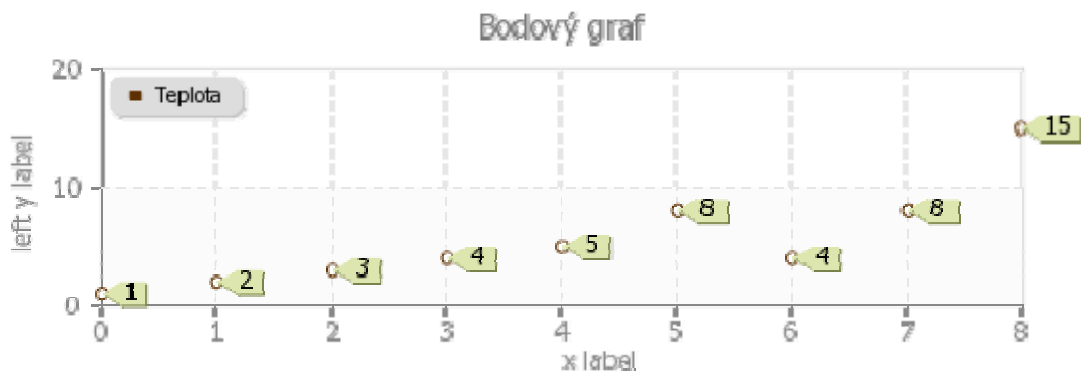
5.2.1 Bodový graf

Tento typ grafu zobrazuje vztah mezi číselnými hodnotami jedné nebo více datových sad. Většinou se používá pro zobrazování vědeckých dat.

Kromě základního nastavení (levý sloupec okna ChartUI), jsou pro tento typ grafu k dispozici záložky (pravá spodní část okna ChartUI) Datová série (Data series) a podrobnější nastavení grafu (Graph appearance).

Datová série je jediná povinná položka (obrázek č. 17). Jméno série se použije pouze v případě, že je zapnuto zobrazování legendy. Nepovinné jsou i popisky datových bodů.

Obrázek ukazuje výsledný graf (obrázek č. 16) a také nastavení datových sérií.



Obrázek č. 16

nr.	serie data	serie name	data point labels
1.	1,2,3,4,5,8,4,8,15	Teplota	1,2,3,4,5,8,4,8,15
2.			

Obrázek č. 17 – nastavení datové série

5.2.2 Spojnicový graf (úsečkový a křivkový)

Tento typ grafu zobrazuje trendy dat za určitou dobu. Obdobně jako u bodového grafu je jedinou povinnou položkou minimálně jedna datová série.

Spojnicový graf může být úsečkový, kde jsou jednotlivé body datové série propojeny úsečkou nebo křivkový, kdy jednotlivé datové body propojuje křivka.

Pomocí zaškrtnutí políčka „draw area“ v levém sloupci Chartu je možné vykreslit plochu pod křivkou, resp. úsečkou.



Obrázek č. 18 – spojnicový, úsečkový graf s vykreslenou plochou



Obrázek č. 19 – spojnicový, křivkový graf s/bez vykreslenou plocho

Image size (px)
x 600 y 200

Plot type
cubic
 draw points
 draw area

Plot title
jnicový graf - křivkový

Plot axes
x label x
 left y label y1
 right y label y2
 draw plot legend

X's abscise
0,1,2,3, ...

Generate

Spojnicový graf - křivkový

left y label

right y label

x label

Teplota

1 2 3 4 5 8 4 8 15

Data Series **Graph appearance** **System**

nr.	serie data	serie name	data point labels
1.	1,2,3,4,5,8,4,8,15	Teplota	1,2,3,4,5,8,4,8,15
2.			

+ add a serie - remove last serie ✎ data-point labels

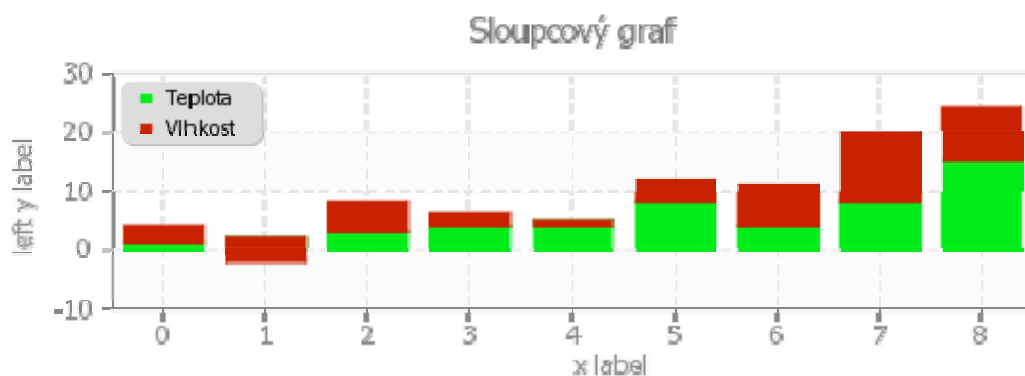
[show examples](#)

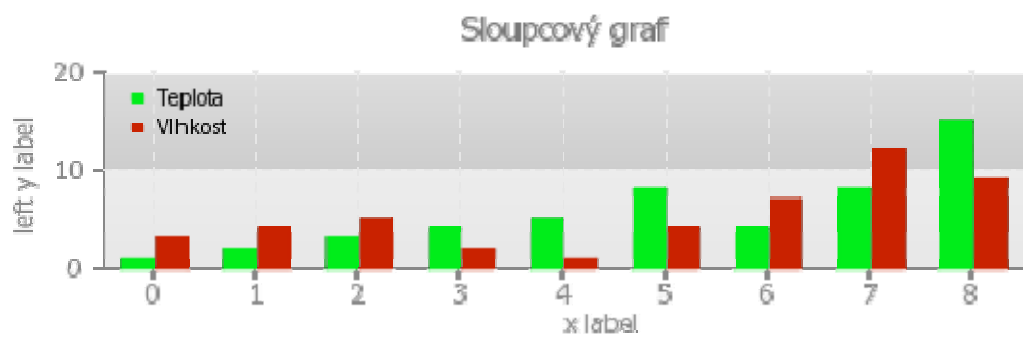
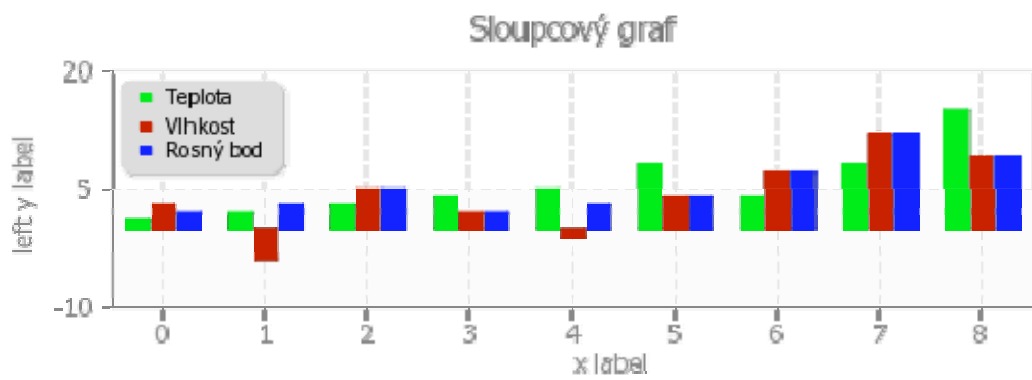
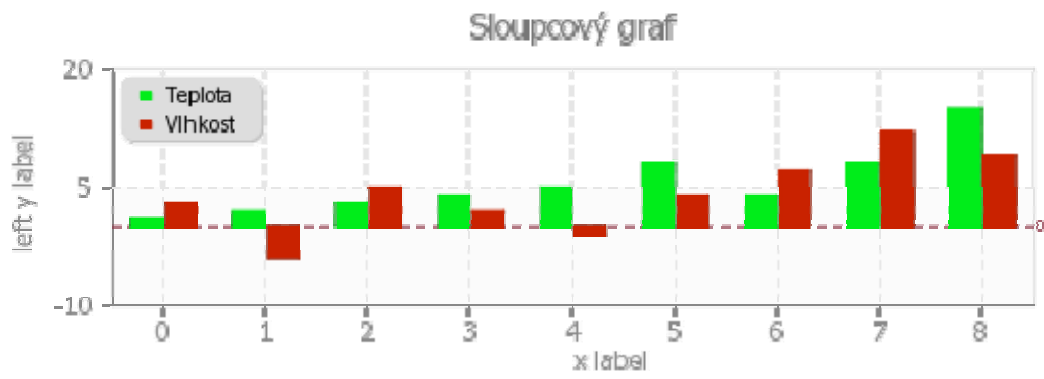
ChartUI v1.0

Obrázek č. 20 – nastavení datové série

5.2.3 Sloupcový graf

Velmi často využívaný typ grafu. Sloupcový graf porovnává hodnoty pro různé kategorie. Požadavky na vstupní data jsou obdobné jako u předchozích typů grafů, tj. povinná je pouze alespoň jedna datová série.





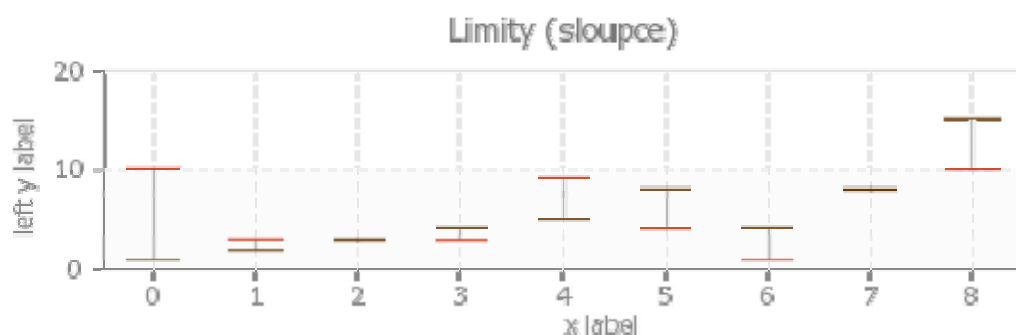
ChartUI v1.0

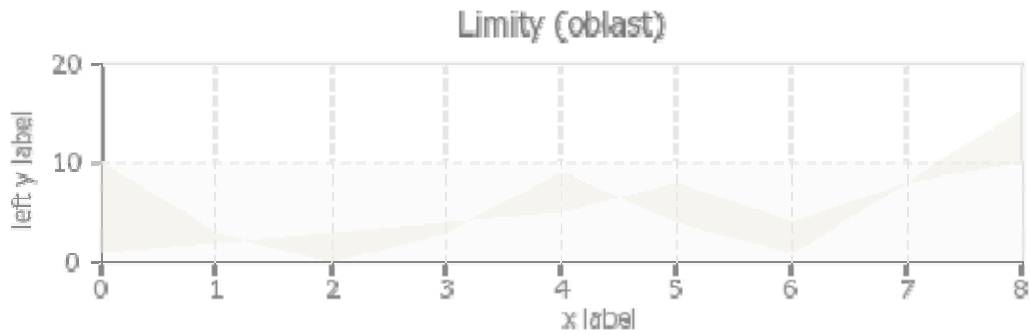
nr.	serie data	serie name	data point labels
1.	1,2,3,4,5,8,4,8,15	Teplota	
2.	3,-4,5,2,-1,4,7,12,9	Vlhkost	

Obrázek č. 21 – ukázka nastavení pro sloupcové grafy

5.2.4 Grafy limit (sloupce a oblasti)

Jedná se o specifický typ grafu, který své využití nalezne ve statistice, případně je ho možné využít k zobrazování vědeckých dat.





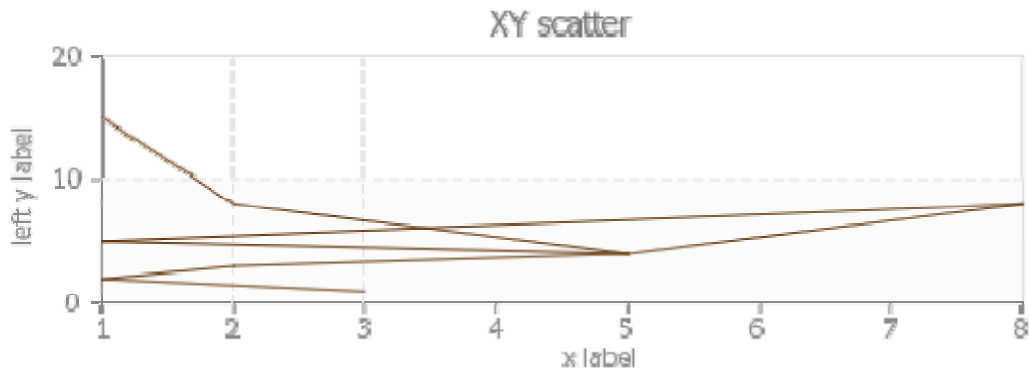
Obrázek č. 22

nr.	serie data	serie name	data point labels
1.	1,2,3,4,5,8,4,8,15		
2.	10,3,ž,3,9,4,1,8,10		

Obrázek č. 23 – Nastavení datových sérií pro grafy limit

5.2.5 XY, Scatter

Graf používaný ve vědeckých kruzích. Zobrazuje vztahy mezi jednotlivými číselnými hodnotami několika datových sérií.



Obrázek č. 24

Image size (px)
x: 600 y: 200

Plot type
XY scatter

Plot title
XY scatter

Plot axes
x label x
 left y label y1
 right y label y2
 draw plot legend

X's abscise
0,1,2,3, ...

Data Series

nr.	serie data	serie name	data point labels
1.	1,2,3,4,5,8,4,8,15		
2.	3,1,2,5,1,8,5,2,1		

add a serie remove last serie data-point labels

show examples

Series 1 is used for y axe, serie 2 for x axe. Point labels will not be visible.

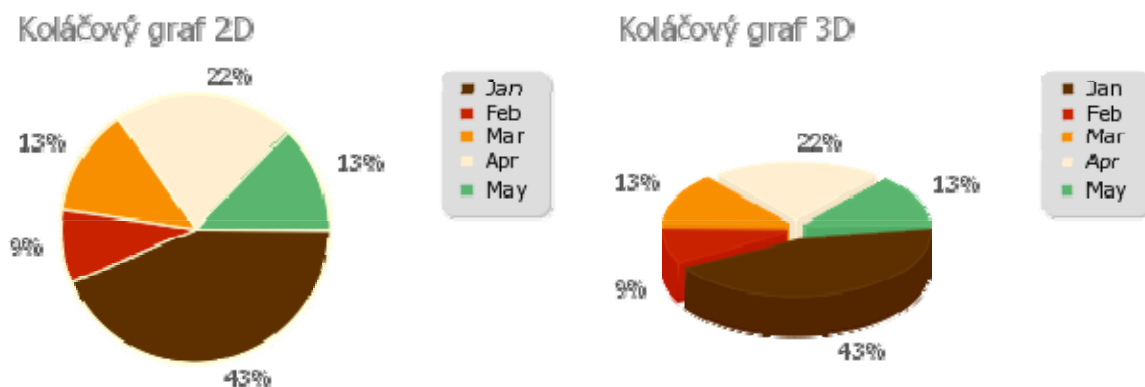
ChartUI v1.0

Obrázek č. 25

5.2.6 Koláčový graf (2D a 3D)

Koláčový graf zobrazuje procentuální poměr jednotlivých hodnot datové série. Součet všech dílů dává 100%. U těchto typů grafu se mírně odlišuje zadávání datových hodnot. Datová série 1 obsahuje popisky dat, které budou zobrazeny v nápovědě. Samotná zobrazovaná data se umísťují do série 2.

Následují ukázky variant v provedení 2D a 3D.



Obrázek č. 26

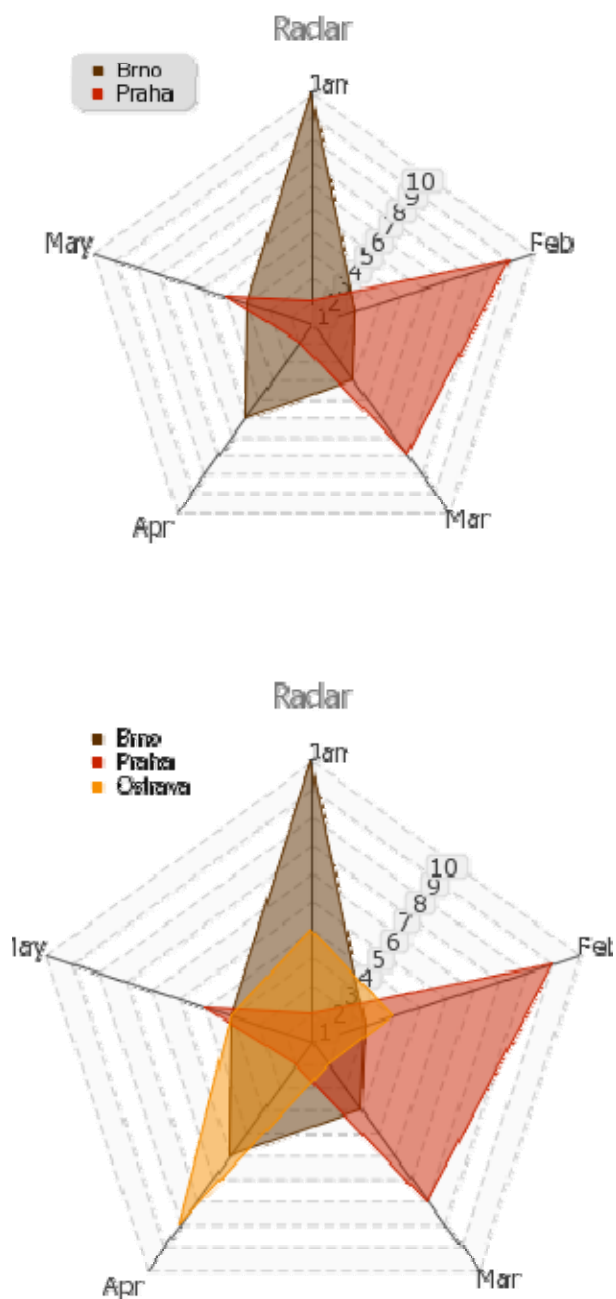
ChartUI v1.0

Obrázek č. 27

5.2.7 Radar graf

Tento typ grafu je také označován jako paprskový graf. Porovnává úhrnné hodnoty několika datových sérií.

Pro zadávání datových sérií platí stejné pravidla jako pro koláčový graf. První datová série obsahuje popisky jednotlivých os grafu. Další série už obsahují samotné datové hodnoty k vykreslení.



Obrázek č. 28

Image size (px)
x 300 y 300

Plot type
radar plot
 draw area

Plot title
Radar
 draw plot legend

X's abscise
0,1,2,3, ...

Generate

Abscise values must be in the Serie 1 dataset!

Radar

Brno
Praha

Jan Feb Mar Apr May

10
9
8
7
6
5
4

Data Series Graph appearance System

nr.	serie data	serie name	data point labels
1.	Jan, Feb, Mar, Apr, May		
2.	10, 2, 3, 5, 3	Brno	
3.	1, 9, 7, 1, 4	Praha	

+ add a serie - remove last serie ✎ data-point labels

[show examples](#)

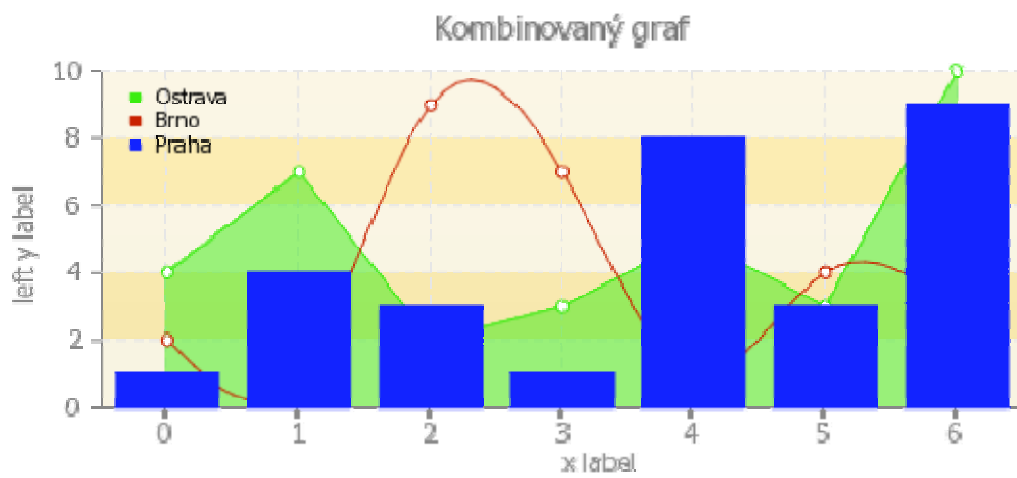
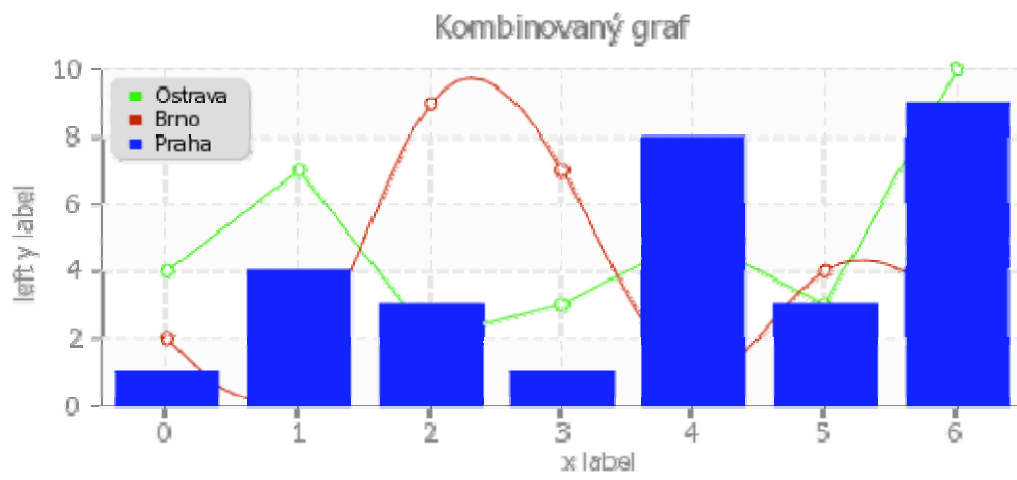
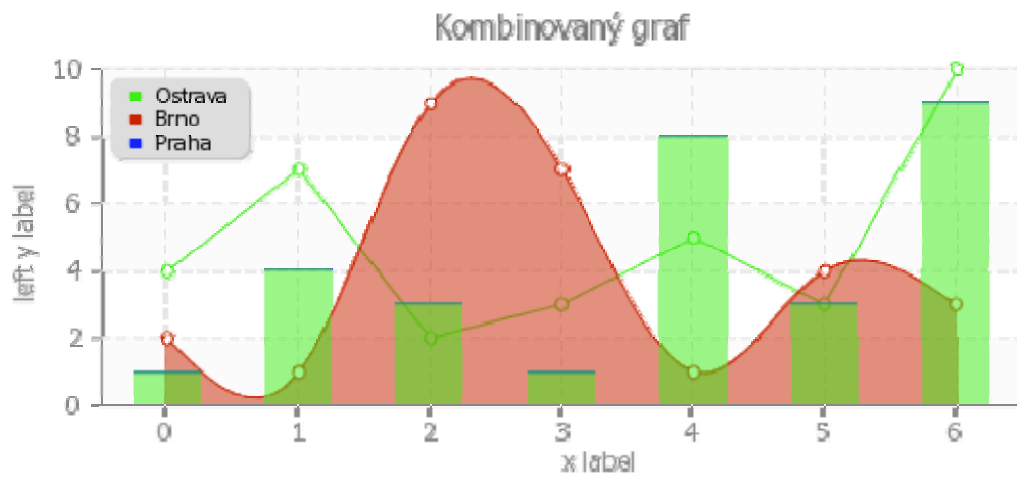
ChartUI v1.0

Obrázek č. 29

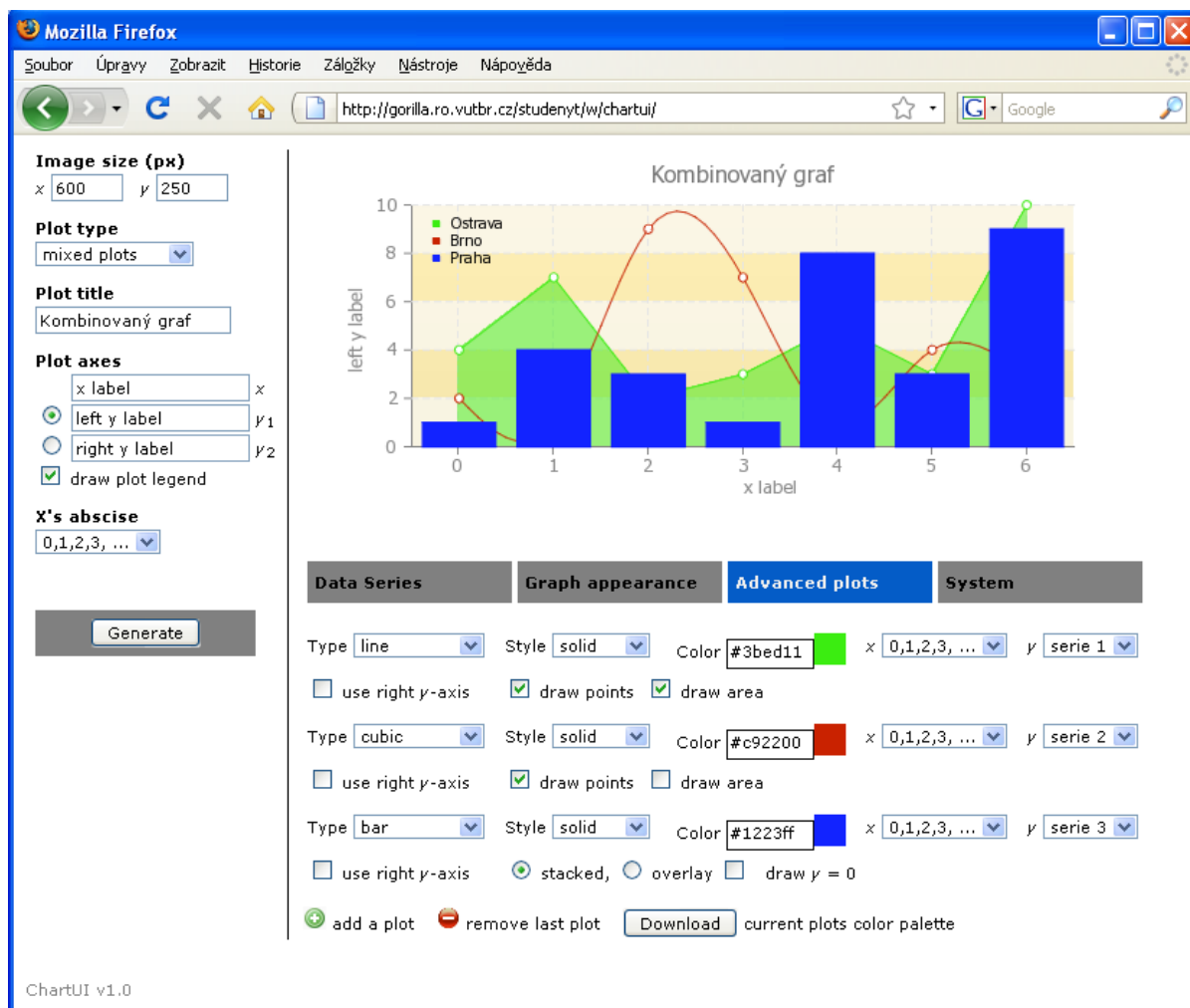
5.2.8 Kombinace několika typů grafu

Samozřejmostí je vložení více typů grafů do jednoho obrázku. Při volbě několik grafů v jednom (mixed plots) je k dispozici 4. záložka (Advanced plots), ve které je možné nastavit typy jednotlivých grafů, změnit jejich barvy a přiřadit jim datové série.

K dispozici jsou všechny grafy uvedené v předchozích kapitolách, kromě koláčového grafu a radar grafu.



Obrázek č. 30



Obrázek č. 31

5.3 Automatika v ChartUI

Vytváření grafu v samotné knihovně pChart často narazí na problém překrývání a zasahování jednotlivých prvků grafu do sebe. To je nepříjemné a je pak třeba graf „doladovat“ např. změnou velikost obrázku grafu, posunem legendy apod.

Rozhraní ChartUI se snaží toto řešit a automaticky vykalkulovat pozice jednotlivých prvků tak, aby byl graf čitelný a nebylo potřeba měnit rozměry a pozice jednotlivých částí. Uživatel tak má prostor řešit obsahovou a estetickou část grafu.

5.4 Použití ChartUI v jiných projektech

Výsledný graf, vytvořený pomocí grafického rozhraní je možné použít třemi způsoby. Tím prvním je uložení samotného obrázku a jeho vložení do stránek, resp. jeho použití v jiném dokumentu. Vzniklý obrázek však již není možné dynamicky měnit.

Další možností je využití jakéhosi rozhraní pomocí předávání parametru grafu přes URL odkaz (obrázek č. 33). Takovýto odkaz přenesení veškeré nastavení pro vytvoření obrázku grafu. Generování tohoto odkazu je třeba zapnout na záložce systém (zaškrtnuté políčko *show api calls*).



Obrázek č. 32

```
graph.php?sirka_grafu=600&vyska_grafu=250&plot_type=41&title=Kombinovaný
graf&xaxe=xlabel&yaxe=leftylabel&y2axe=rightylabel&points=1&area=1&plot_1
egend=1&bargraf=1&ibgc=dddddd&gbgc=fff0b6&ibgc_type=1&ibgc_gradient=100&gb
gc_type=3&gbgc_gradient=10&title_fs=12&title_font=tahoma.ttf&tfc=777777&lp
xl=5&lpxt=5&legend_fs=8&legend_font=tahoma.ttf&lbgc=dddddd&lfc=000000&pr=3
&xformat=number&xunits=&yformat=number&yunits=&y2format=number&y2units=&pp
=3&axe_fs=10&axe_font=tahoma.ttf&axec=888888&xmin=0&xmax=0&ymin=0&ymax=0&y
parts=5&y2min=0&y2max=0&sel_yaxe=1&xaxe_abscise=0&colorset=tones.txt&ly_sp
ace=70&x_space=50&ry_space=70&serie_1=4,7,2,3,5,3,10&serie_1_name=Ostrava&
serie_1_dpl=&serie_2=2,1,9,7,1,4,3&serie_2_name=Brno&serie_2_dpl=&serie_3=
1,4,3,1,8,3,9&serie_3_name=Praha&serie_3_dpl=&plot_type_1=2&area_1=1&point
s_1=1&xaxe_1=0&yaxe_1=1&y1axe_1=1&y2axe_1=1&bargraf_1=2&line0_1=1&ls_1=1&p
lc_1=3bed11&plot_type_2=3&points_2=1&xaxe_2=0&yaxe_2=2&y1axe_2=1&y2axe_2=1
&bargraf_2=2&line0_2=1&ls_2=1&plc_2=c92200&plot_type_3=7&points_3=1&xaxe_3
=0&yaxe_3=3&y1axe_3=1&y2axe_3=1&bargraf_3=2&ls_3=1&plc_3=1223ff
```

Obrázek č. 33

Vzniklý odkaz pak stačí vložit do html stránky jako klasický, nepárový tag pro vložení obrázku, tj. ``.

Poslední možností je exportování zdrojového kódu grafu do formátu, který je popsán v kapitole „První graf pomocí pChart“. Tento export je pak možné použít v PHP skriptu na místě, kde má být graf zobrazen a pouze se vymění vstupní datové série. Tlačítko pro export se nachází v záložce „Systém“ (obrázek č. 32).

Následuje ukázka vložení vyexportovaného grafu do PHP skriptu.

```
include("pChart/pData.class");
include("pChart/pChart.class");
include("pChart/pCache.class");

include("exportovany_script_z_chartui.php");
```


6 Zásuvný modul pro MediaWiki

Jedním z cílů bylo vytvoření rozhraní mezi pChart GUI projektem a samotnou knihovnou pChart. Díky tomuto rozhraní je možné vytvářet a vkládat grafy do stránek MediaWiki jako prostý text pomocí speciálních tagů. Při zobrazení stránky MediaWiki se pak na místě takového kódu zobrazí výsledný graf. Vkládané grafy je pak možné různě uzpůsobovat a vytvářet tak velice dobře vypadající výstupy.

6.1 MediaWiki

MediaWiki je volně šiřitelný software pod licencí GNU General Public Licence. Jedná se o projekt založený na skriptovací jazyce PHP, který data ukládá do MySQL databáze. MediaWiki umožňuje spolupráci uživatelů na nejrůznějších projektech, díky týmové práci na jednotlivých stránkách, které může každý člen týmu (a nebo návštěvník) vytvářet, upravovat či doplňovat. Vzniká tak velmi silný nástroj pro komunikaci při vytváření týmových projektů. Důležité však je, že stav před úpravou se uchovává v databázi a proto je možné se později vrátit k původní verzi. Každá stránka má tedy svoji historii úprav, a ať už nechtěný nebo chtěný zásah je možné dohledat a vrátit zpět.

Uměle vytvořené slovo „wiki“ označuje webové stránky určené k práci a komunikaci v týmu. MediaWiki původně vznikla jako základ pro známou otevřenou internetovou encyklopedii Wikipedie.

Další nespornou výhodou jádra MediaWiki je připravenost projektu na provoz na velkých serverových farmách [Obrázek č. 34], které pak bez problémů zvládnou odbavit velké množství uživatelů / požadavků. To je možné díky rozložení zátěže na více serverů, které mohou být rozmístěny kdekoli na světě a propojeny přes Internet (obrázek č. 34).



Obrázek č. 34

Už z názvu plyne, že si MediaWiki poradí s různými typy médií. Ať se jedná o prostý text, obrázky nebo videa. Multimediální obsah je ukládán do filesystému na serveru a podobně jako u stránek je uchovávána historie změn těchto souborů.

MediaWiki je velmi silný nástroj a díky licenci GLP a jedinečné možnosti MediaWiki upravovat a přizpůsobovat vlastním projektům se stala základem pro desítky tisíc „Wiki“ webu po celém světě.

6.2 Zásuvné moduly do MediaWiki

Schopnosti MediaWiki je možné dále rozšiřovat pomocí zásuvných modulů (tzv. plug-in). Napojením externí knihovny přibývají další možnosti vkládání objektů do stránek MediaWiki pomocí prostého textu.

Velmi užívaný je například modul pro vkládání matematických nebo fyzikálních vzorců. Vkládání je velice jednoduché, příklad napoví více (obrázek č. 35).

```
f(x,y,z) = 3y^2 z \left( 3 + \frac{7x+5}{1+y^2} \right)
```

$$f(x, y, z) = 3y^2 z \left(3 + \frac{7x + 5}{1 + y^2} \right)$$

Obrázek č. 35

Samotná instalace modulu se provádí zkopírováním knihovny zásuvného modulu pchart.php do podadresáře „extensions/pchart“, kde je nainstalována MediaWiki. Dále je třeba ve skriptu LocalSettings.php (soubor s nastavením MediaWiki) vložit knihovnu, která bude obsluhovat generování grafů.

```
require_once("extensions/pchart/pchart.php");
```

Tímto je MediaWiki připravena na vkládání grafů a bude umět reagovat na níže uvedené tagy.

6.3 Vložení grafu do WikiTextu

Samotné vložení grafu se provádí zapsáním párového tagu `<pchart></pchart>`. Mezi tyto tagy se vkládají další řídicí značky, které určují typ grafu, datové série apod. Vše, co bude uvedeno uvnitř

těchto tagů MediaWiki nezobrazí, ale bude se snažit provést za pomoci zásuvného modulu pchart.php.

Tag `<pchart>` má tři volitelné parametry: `sirka="xxx"`, `vyska="yyy"`, `typ="z"`. Pokud nejsou tyto hodnoty zadány, nastaví se výchozí hodnoty, a to: `sirka="600"`, `vyska="300"` a `typ="2"`.

Parametr `typ` určuje typ grafu. Možné hodnoty jsou následující:

- 1 - bodovy
- 2 - lomene cary
- 3 - krivky
- 7 - bar
- 21 - pie 2D
- 22 - pie 3D

Uvnitř tagu `<pchart>` se píše vlastnosti grafu a zobrazovaná data. Při zadávání je třeba uvádět každou vlastnost na nový řádek. Tag `<pchart>` by měl obsahovat pouze jeden oddělovač `[DATA]`, který separuje vlastnosti od dat. Se vším, co je za ním, se nakládá jako s daty a se vším, co je před ním, jako s vlastnostmi grafu. Důvodem tohoto oddělení je rozdílné zpracování vlastnosti a dat.

Každý řádek s vlastnostmi začíná značkou `[název vlastnosti]` za ní je mezera, následovaná hodnotou vlastnosti. Akceptovány jsou následující značky.

`[NAZEV]` titulek grafu

`[POPIS-X]` název osy x

`[POPIS-Y-LEVA]` název osy y

Vlastnosti u kterých není očekávána žádná hodnota jsou tyto:

`[NOLEGEND]` – pokud je uvedeno, nevykreslí legendu

`[NOPOINTS]` – pokud je uvedeno, nevykreslí v grafu datové body

Dále pak následuje oddělovač `[DATA]` a datové hodnoty k znázornění v grafu. Najednou je možné vykreslit až 10 datových řad, vyjma grafu typu *pie*, kde je datová řada pouze jedna. Řada může obsahovat 1 až 3 řádky. Nutný je pouze 1 řádek, který začíná značkou `[SERIE x]`, kde x je pořadové číslo série jdoucí po sobě od 1 až po 10. Hodnota 10 je nastavitelné maximum, které se ukázalo jako dostatečné. Za `[SERIE x]` jsou pak uvedena data oddělená čárkou, například: `[SERIE1] 2,8,9,7,8`.

Ke každé datové řadě je možné přidat další dva řádky určující popisky hodnot.

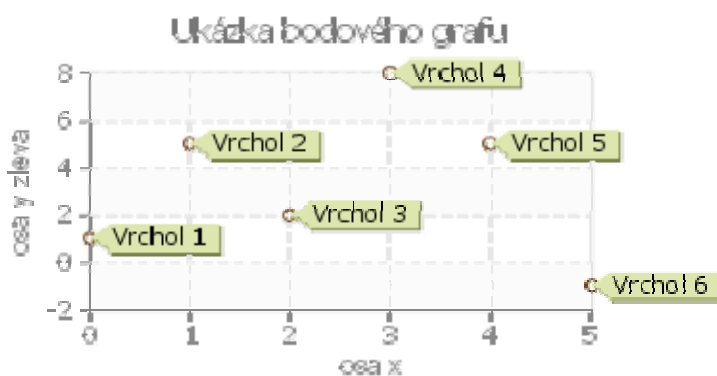
[NSx] název série (pro výpis v legendě)

[LABELSx] popisky jednotlivých datových bodů, hodnoty oddělené čárkou.

Malé písmeno x opět označuje číslo datové řady, ke které se popisek vztahuje. Počet popisků bodu v *[LABELSx]* by měl být shodný s počtem datových hodnot v *[SERIEx]*.

Pokud se vykresluje do jednoho grafu více sérií, měly by mít všechny stejný počet datových bodů. Speciálním případem je *pie* graf (2D i 3D), kde datová série 1 vždy musí obsahovat legendu a data k vykreslení jsou v datové sérii 2.

Následují ukázky všech 6 nejpoužívanějších typů grafů a ukázky zápisu tagů, které se postarají o zobrazení grafu v MediaWiki.



Obrázek č. 36

```
<pchart sirka="401" vyska="200" typ="1">
[NAZEV] Ukázka bodového grafu
[POPIS-X] osa x
[POPIS-Y-LEVA] osa y zleva
[NOLEGEND]
[DATA]
[SERIE1] 1,5,2,8,7,-1
[LABELS1] Vrchol 1,Vrchol 2,Vrchol 3,Vrchol 4,Vrchol 5,Vrchol 6
</pchart>
```



Obrázek č. 37

```

<pchart sirka="390" vyska="200" typ="2">
[NAZEV] Ukázka spojnicového grafu (lomené čáry)
[POPIS-X] osa x
[POPIS-Y-LEVA] osa y
[NOLEGEND]
[DATA]
[SERIE1] 1,2,8,5,7,10
</pchart>

```



Obrázek č. 38

```

<pchart sirka="390" vyska="200" typ="3">
[NAZEV] Ukázka spojnicového grafu (křivky)
[POPIS-X] osa x
[POPIS-Y-LEVA] osa y
[NOLEGEND]
[DATA]
[SERIE1] 1,2,8,5,7,10
</pchart>

```

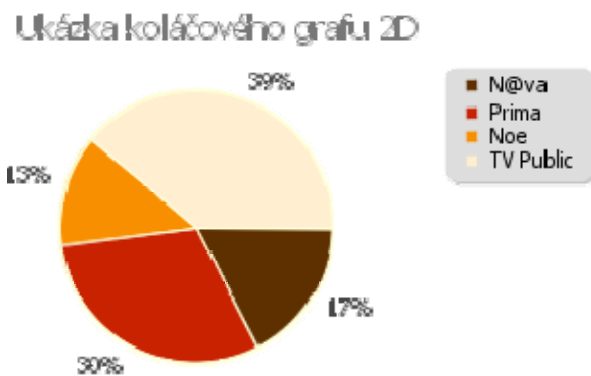


Obrázek č. 39

```

<pchart sirka="390" vyska="200" typ="7">
[NAZEV] Ukázka sloupcového grafu
[NOPOINTS]
[DATA]
[SERIE1] 4,7,3,9
[NS1] jméno série 1
[SERIE2] 5,8,6,1
[NS2] jméno série 2
</pchart>

```



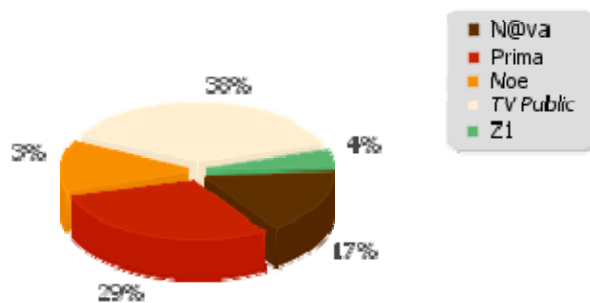
Obrázek č. 40

```

<pchart sirka="390" vyska="200" typ="21">
[NAZEV] Ukázka koláčového grafu 2D
[DATA]
[SERIE1] N@va,Prima,Noe,TV Public
[SERIE2] 4,7,3,9
</pchart>

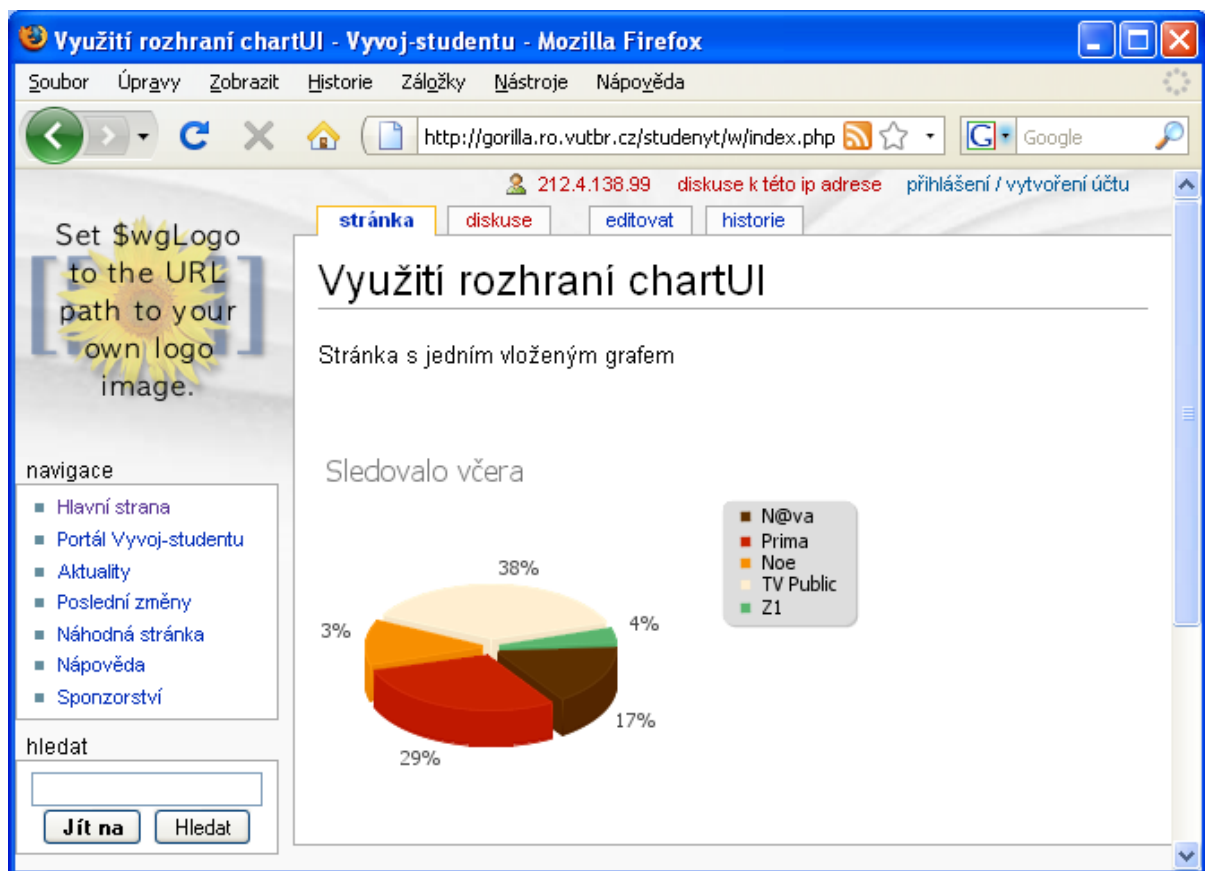
```

Ukázka koláčového grafu 3D



Obrázek č. 41

```
<pchart sirka="390" vyska="200" typ="22">
[NAZEV] Ukázka koláčového grafu 3D
[DATA]
[SERIE1] N@va,Prima,Noe,TV Public,Z1
[SERIE2] 4,7,3,9,1
</pchart>
```



Obrázek č. 42 – ukázka stránky v MediaWiki

7 Fonty a čeština

Nedílnou součástí grafického zobrazování informací a dat pomocí grafů jsou textové výstupy, jako popisky os, název grafu, legenda grafu atd. Většina knihoven generující grafy využívá TrueType fonty. Ne všechny fonty jsou však k dispozici v české mutaci a může tak docházet k problému se správným zobrazováním českých znaků. Jestliže dojde k použití fontu, který nepodporuje češtinu, místo požadovaného znaku se zobrazí znak jiný nebo je prostě vynechán.

Práce s fonty v knihovně pChart je velice jednoduchá. Stačí požadovaný TrueType font nahrát do podadresáře Fonty, který se nachází ve stejném umístění jako samotná knihovna pChart. Takto nahraný font je pak možné použít k vytváření grafů v rozhraní ChartUI.

Problematika fontů je s grafy úzce spjatá a proto si ukažme nečastěji používané typy fontů.

7.1 TrueType fonty

TrueType je standard pro popis vektorových písem, který byl vyvinut firmou Microsoft koncem 80. let. Zamýšlen byl jako konkurent fontu Type 1 pro jazyk Postskript firmy Adobe. TrueType fonty využívala první grafická nástavba na systém MS-DOS, tj. Windows 3.1.

V současné době fonty TrueType podporuje většina platforem: Microsoft Windows, GNU/Linux (ve formě FreeType), Mac OS.

Na internetu existuje velká řada specializovaných serverů, kde je možné stáhnout velké množství TrueType fontů a to většinou zdarma [grafikacz].

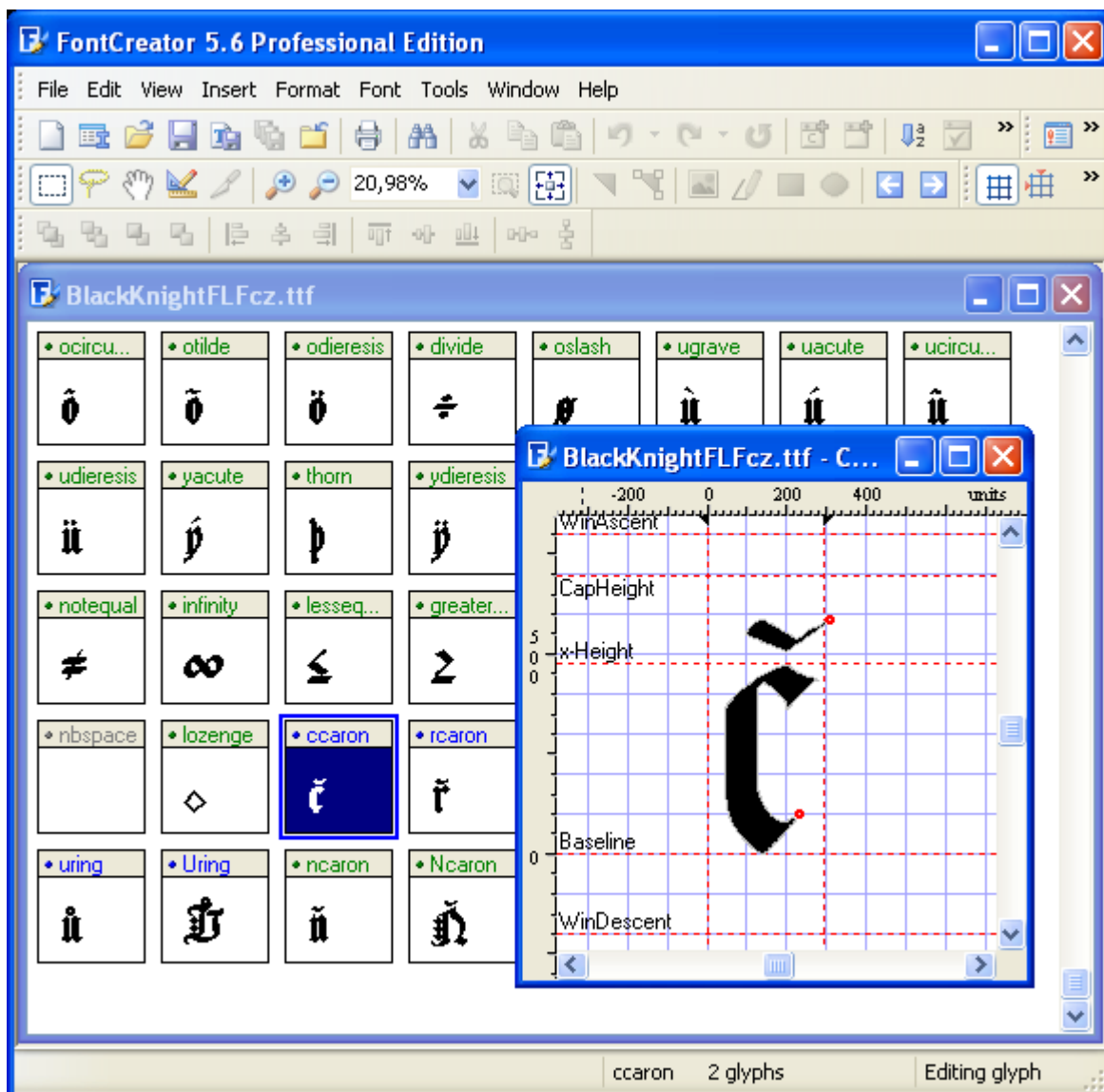
7.1.1 Aplikace TrueFontEditor

Problémy s češtinou se mohou objevit i s fontem, který na jiném stroji fungoval bez problémů. To co fungovalo na platformě Macintosh, nemusí správně fungovat na systémech běžících s operačním systémem Microsoft Windows. Hlavním důvodem většinou bývá špatné mapování národních znaků, tj. znaků s diakritikou, ve znakové tabulce příslušného fontu.

V některých situacích je obtížné najít font s podporou českých znaků podobný tomu, který je právě potřeba. Tuto situaci pomůže vyřešit velice rychle nástroj pro editaci fontů TrueFontEditor.

Aplikace zvládá pracovat jak s fonty TrueType, tak s novější variantou OpenType. Možné je upravit nejen tabulku znaku a její mapování (obrázek č. 44) pro jednotlivé platformy, ale také chybějící znaky vytvořit.

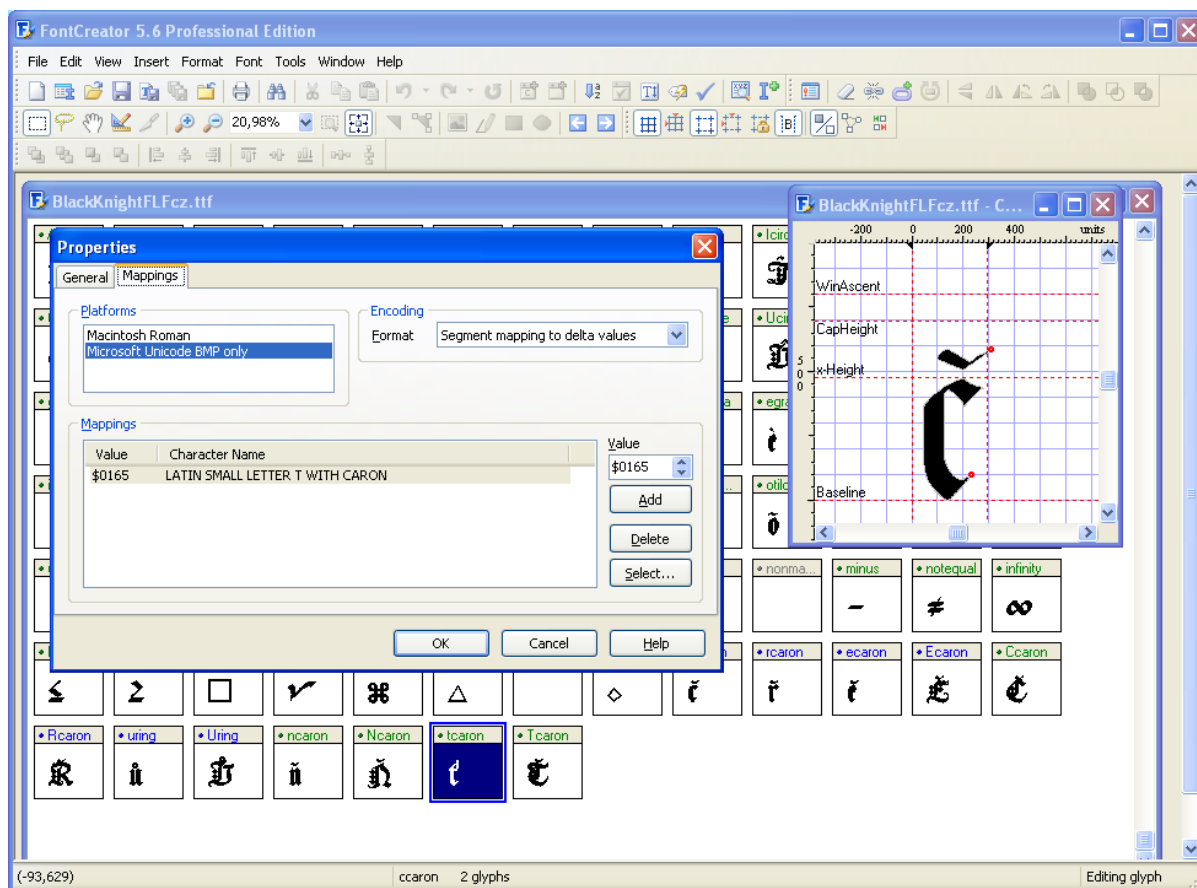
To je možné díky tomu, že nedílnou součástí aplikace je i jednoduchý, ale výkonný editor jednotlivých znaků fontu (obrázek č. 43).



Obrázek č. 43 – editor znaků fontu

Tento editor také umožňuje zajímavou funkci vytvoření znaků z loga, případně je možné vytvoření fontu z vlastního rukopisu.

Z přehledného zobrazení všech znaků ve fontu je možné velice rychle zjistit, který znak v tabulce chybí a pomocí editoru ho doplnit.



Obrázek č. 44 – nastavení mapování znaků v tabulce fontů

7.2 OpenType fonty

Novější standard a nástupce fontů TrueType, který vzniknul ve spolupráci firem Microsoft a Adobe. Formát OpenType sdružuje kladné vlastnosti TrueType a Postskript Type 1 fontů.

Mezi přednosti tohoto nového standardu patří možnost popisu znaků jak formátem TrueType, tak Type 1. Soubor je schopen popsat až 65 536 znaků, snižuje se problém s lokalizací díky podpoře kódování unicode. V neposlední řadě jsou fonty OpenType přenositelné mezi platformami Microsoft Windows, Apple a Unix [grafikacz].

8 Závěr

Tento diplomový projekt si dal za úkol seznámit se s problematikou generování grafů na webu. Zjistit, jaké pro tento účel existují nástroje a zvolit jeden – ten, který nejlépe řeší uvedenou problematiku. Zároveň bylo požadováno prostudování ukládání obrázků do cache paměti, což je téma, které s generováním grafů pro rozsáhlé weby úzce souvisí, protože v některých případech pomáhá dramaticky snížit jak zátěž serveru, tak zátěž konektivity.

Práce navazuje na semestrální projekt, který se více zaměřil na knihovnu JPGgraph. Nicméně přednosti knihovny pChart, které byly uvedeny dříve, nakonec nasměrovaly diplomový projekt právě na tuto knihovnu.

Výsledkem je přívětivé uživatelské rozhraní knihovny pChart, ve kterém je možné jednoduše ze zadaných dat (číselná řada, popis číselné řady, typ grafu, nadpis, apod.) vytvořit profesionálně vypadající graf, měnit jeho vzhled nebo zvolit nejvhodnější typ tak, aby co nejlépe, přehledně a přesně zobrazoval modelovaná vstupní data. Výsledný graf je pak možné uložit jako běžný jpeg, popřípadě png obrázek na disk a dále s ním pracovat (publikování na web, místní intranet, použití v dokumentu apod.). Je zde také možnost začlenění do informačního systému nebo webu, kdy se výsledný obrázek generuje podle dat poslaných metodou GET v adrese generujícího skriptu. Jedná se o jednoduché rozhraní a výsledný odkaz, který obrázek vytvoří, je možné zkopírovat z uživatelského rozhraní po dokončení uprav grafu. Další možností je vyexportování zdrojového skriptu grafu a jeho samostatné použití ve spojení s knihovnou pChart.

Projekt tedy zpřístupnil na pohled velmi pěkně vypadající grafy generované knihovnou pChart i pro uživatele, kteří požadují rychlé vytvoření dobře vypadajícího grafu, bez nutnosti psát obsáhlý kód.

Díky modulu pro MediaWiki je možné graf vytvořit přímo z textu za pomoci metaznaků a xml značek. Vkládání grafů je pak velice jednoduché. Graf se vkládá jako určitá posloupnost textu a není nutné ho ukládat jako obrázek, což ušetří nejen práci, ale i místo, které by vložený jpg obrázek obsadil.

Jeden z cílů projektu bylo uplatnění vytvořeného nástroje v prostředí webového portálu VUT Brno. Právě funkce export parametru grafu jako zdrojový kód pro pChart pomohou zjednodušit implementaci dobře čitelných a esteticky pěkně vypadajících grafů nejen v portálu VUT Brno, ale i v navazujících projektech.

Na tuto diplomovou práci by bylo možné v budoucnu navázat a projekt rozšířit na službu poskytující vytváření online grafů. Stejně tak je potřeba udržovat krok s tvůrci pChart a doplňovat nové možnosti práce s grafy.

Diplomový projekt – demonstraci uživatelského rozhraní a použití zásuvného modulu pro MediaWiki je možné nalézt a vyzkoušet na adresách:

<http://gorilla.ro.vutbr.cz/studenyt/w/chartui>

<http://gorilla.ro.vutbr.cz/studenyt/w/index.php>

Jako alternativní odkazy je možné použít následující adresy:

<http://www.freeze.cz/chartui>

<http://www.freeze.cz/mediawiki>

Literatura

- [kosek98] Košek, Jiří: PHP – Tvorba interaktivních internetových stránek, 1. vydání.
Granada, 1998
<http://www.kosek.cz> (duben 2005)
- [kosek98] Mach, J.: PHP pro úplné začátečníky.
Computer Press, 2002
- [phpnet] PHP: Hypertext Preprocessor.
<http://www.php.net> (duben 2008)
- [graphpите] Graphpите: Open Source PHP OO Graph Rendering Utility
<http://graphpите.sourceforge.net> (listopad 2008)
- [chart] Chart: PHP4 library for generating charts
<http://quimby.gnus.org/circus/chart/chart-manual.php> (listopad 2008)
- [chartdirector] ChartDirector: Professional charts komponent
<http://www.advsofteng.com> (listopad 2008)
- [postgraph] PostGraph: Free PHP graph class
<http://www.advsofteng.com> (listopad 2008)
- [phplot] PHPlot
<http://www.phplot.com> (listopad 2008)
- [jpgraph] JPGraph: OO knihovna pro generování grafů
<http://www.aditus.nu/jpgraph> (listopad 2008)
- [XAMPP] XAMPP: Apache Friends
<http://www.apachefriends.org/en/xampp.html> (květen 2008)
- [jquery] jQuery: JavaScript library
<http://www.jquery.com> (prosinec 2008)

- [mediawiki] MediaWiki: Domovské stránky projektu MediaWiki
<http://www.mediawiki.org> (leden 2009)
- [pchart] pChart: Domovské stránky projektu pChart
<http://pchart.sourceforge.net> (leden 2009)
- [grafikacz] grafika.cz: Novinky ze světa grafiky, polygrafie, digitální fotografie
<http://www.grafika.cz> (leden 2009)

Příloha A

Tato příloha popisuje obsah přiloženého DVD. Přiložený datový nosič (1 ks DVD) obsahuje následující části diplomové práce:

adresář SOURCE

- zdrojový kód systému a další jeho grafické části (v archivu ZIP)

adresář DOCUMENTATION

- dokumentaci v elektronické podobě (ve formátu PDF)

adresář SCREENSHOT

- screen-shoty aplikace, grafů (ve formátu PNG)

adresář DEMO

- ukázková videa vytváření grafů (ve formátu WMV)

Seznam příloh

Příloha A. popis přiloženého DVD se zdrojovými kódy

Příloha B. DVD se zdrojovými kódy