

Návrh knihovny vybraných kartografických technik umožňujících rozšíření funkcionality mapových serverů

Tomáš Minarčík

Institut geoinformatiky, Hornicko-geologická fakulta, Vysoká škola báňská - Technická
univerzita Ostrava, 17. listopadu 15,
708 33, Ostrava - Poruba, Česká republika
Kontakt .min057@vsb.cz

Abstrakt. Tato diplomová práce je zaměřena na návrh a realizaci programové knihovny, obsahující sadu tříd, poskytujících možnosti nastavení vybraných typů legendy a klasifikačních metod pro systémy pracující na bázi mapových serverů. Silnou stránkou navrženého modelu je snadná rozšiřitelnost pro potřeby budoucího vývoje. Dále se práce zaměřuje na návrh klientské aplikace, představující jednu z možností jak využít celého rozsahu funkčnosti vytvořené programové knihovny. Výsledkem praktické části této diplomové práce je ucelený projekt realizovaný s nasazením open-source technologií.

Klíčová slova: programová knihovna, typ legendy, klasifikační metoda, mapový server, Internet, objektově orientovaný návrh, klientská aplikace.

Abstract. Design of the library of selected cartographic techniques extending map server's functionality. This graduation theses is focused on the design and realization of program library containing classes that offer possibilities for setting up assorted legend types as well as classification methods for systems working on the basis of mapping servers. The strong point of designed model is an easy extensibility for the needs of the future development. In addition to this, the theses is also focused on the design of client application presenting one of the ways of utilizing the whole potential of the program library . The result of this practical part of this theses is an integrated project realized using open-source technology.

Keywords: program library, legend type, classification Method, mapping server, Internet, object oriented design, klient application.

1 Úvod

Zobrazení prostorových dat bylo ještě před několika lety výsadou pouze desktopových GIS aplikací. Vývoj s použitím Internetu se ubíral od vizualizace v podobě statických obrázků map na internetové stránce až k mapovým serverům nabízejících i ovládací prvky pro interaktivní manipulaci se zobrazenými prostorovými daty. Mapové servery v prostředí internetu se v dnešní době stále častěji stávají součástí internetového života nejen GIS specialistů ale i „obyčejných“ lidí. Další vývoj internetových aplikací obecně i internetových aplikací pro oblast geoinformatiky se ubírá směrem ke tvorbě dynamického přizpůsobení a změny obsahu s možností poskytovat funkcionalitu internetové aplikace koncovým uživatelům.

Výběr způsobu zobrazení vrstvy prostorových dat provádí vývojáři a správci mapového serveru. Uživatelé, pro které je ovšem systém v první řadě určen, nemají ve většině případů možnost zasáhnout do tohoto procesu a nemohou si nastavit parametry pro způsob vizualizace např. požadovaného geografického jevu a musejí se spokojit s přednastavenými parametry.

Tato práce se snaží podat návrh programové knihovny, pomocí níž lze vytvářet interaktivní prostředí, kde by uživatelé měli možnost výběru a nastavení typů legendy s použitím klasifikačních metod využitelných ke změně způsobu vykreslování vrstev. Při objektově orientovaném návrhu programové knihovny byl kladen důraz na vysokou rozšiřitelnost o nové možnosti k usnadnění budoucího vývoje.

Součástí práce je kromě navržené programové knihovny také tvorba ukázkové klientské aplikace, kterou je možno implementovat do již vytvořených systémů pracujících s mapovými servery a rozšířit tak jejich funkcionalitu o možnosti nastavení typů legendy a klasifikačních metod. Pomocí klientské aplikace je koncovým uživatelům poskytnuta rozšířená možnost konfigurace systému pomocí přehledného grafického uživatelského rozhraní.

Programová knihovna s klientskou aplikací se snaží být jedním z malých kroků při tvorbě internetových aplikací, které by byly schopny přizpůsobení systému požadavkům svých uživatelů oproti většině stávajících systémů, kterým se musejí přizpůsobovat uživatelé.

2 Postup realizace projektu

Realizace projektu byla rozdělena do několika dílčích úkolů, které byly postupně nebo v určitých případech společně řešeny.

- Základní pojmy a současný stav problematiky
- Seznámení se s technologiemi
- Výběr technologií a programovacích jazyků
- Objektově orientovaný návrh
- Realizace programové knihovny
- Realizace klientské aplikace

2.1 Základní pojmy a současný stav problematiky

Část práce se opírá o znalosti z vědního oboru tématická kartografie a část o znalosti návrhu a vývoje programových aplikací v prostředí internetu.

Z oboru tématická kartografie se seznámí s oblastmi zabývající se typy legendy a klasifikačními metodami. Dále bylo nutné prostudovat problematiku objektově orientovaného modelování a objektově orientovaného programování v prostředí internetu.

2.1.1 Objektově orientované programování

Objektově orientovaným programováním je označován způsob psaní programového kódu, přístup vývojáře při návrhu a realizaci a také samotný pohled na zkoumaný problém. Takovéto vytváření programů začalo více pronikat do podvědomí vývojářů až poslední dobou, ale jeho počátky jsou již v šedesátých letech 20. století.

Objektově orientované programování bylo poprvé použito v programovacích jazycích Simula. SIMULA I (1962-65) a Simula 67 (1967) jsou první dva objektově orientované jazyky. Simula 67 zavedl nejvíce klíčových pojmů objektově orientovaného programování jako jsou objekty, třídy, podtřídy (obvykle nazývány jako dědičnost) a virtuální procedury, kombinované s bezpečným odkazováním a mechanismy pro vytváření programových kolekcí programových struktur popsaných jednotným názvem třídy. Jazyky Simula vyvinuli Ole-Johan Dahl a Kristen Nygaard v Norském počítačovém centru v Oslu.

Pojmy objektově orientovaného programování:

- Objekt
- Třída
- Instace
- Metoda
- Vlastnost
- Úroveň přístupu
- Abstrakce

- Zapozdření
- Dědičnost
- Polymorfismus

Objektově orientovaný způsob se snaží vytvářet programy pomocí jednotlivých objektů, které jsou v určitém stavu, který si pamatují a dokáží také svůj stav změnit. Objekty v počítači by měly co nejvíce připomínat objekty reálného světa okolo nás. Komunikace mezi objekty se uskutečňuje pomocí zpráv, které si mezi sebou objekty zasílají a reagují na ně pomocí metod. [1], [4], [5], [7], [14], [21]

2.1.2 Tématické mapy

Tato diplomová práce se zabývá návrhem programové knihovny, obsahující metody k vývoji nástrojů pro nastavení a výpočet parametrů pro typy legendy a klasifikační metody využitelné při vytváření tématických map v prostředí mapového serveru na Internetu, proto problematika tématických map je jedna z součástí této práce.

Termín tématická mapa lze definovat jako mapu, která na topografickém podkladu znázorňuje jedno nebo více zvláštních témat na úkor nepodstatných témat a je určena ke zcela specifickému účelu. [19]

2.1.3 Určení počtu tříd

K zajištění adekvátního sdělení kartografické informace je nutné určit správný počet tříd pro hodnoty zpracovávaného statistického souboru dat. Určený počet tříd je důležitý z hlediska ztráty informace o rozdělení počtu jednotek v celé šíři možného výskytu nebo snížení přehlednosti třídění. Ztráta informace se projevuje při zvolení malého počtu tříd. Snížení přehlednosti třídění a zánik charakteristických rozložení hodnot se projevuje při zvolení nadměrně velkého počtu tříd. Stav, kdy je počet tříd shodný s počtem prvků datového souboru je brán jako nepřijatelný stejně jako pokud nastane možnost, kdy počet tříd je vyšší než počet prvků datového souboru a tedy některé třídy zůstávají neobsazeny. [19]

2.1.4 Dělení stupnic

Při dělení stupnic lze rozlišit dvě hlavní kategorie stupnic použitelné pro kartogramy nebo kartodiagramy. Tyto kategorie lze dále dělit do několika skupin. Navrženo bylo terminologicky propracovanější dělení stupnic pro tematickou kartografii a počítačovou kartografii, které je logické a respektuje teorii i praxi tvorby stupnic. Tento systém dělení je rozšířitelný o další stupnice.

Tab. 1 Dělení stupnic [11]

STUPNICE			
1. INTERVALOVÁ		2. FUNKČNÍ	
1.1. plynule navazující	1.2. skoková	2.1. spojitá	2.2 skoková
1.1.1. konstantní	1.2.1. s hiátem		2.2.1. hiátem
1.1.2. Pravidelně rostoucí (pravidelně klesající)			2.2.2. v důsledku změny vzorce
1.1.3. nepravidelná			

Pro tvorbu kartogramů se používá intervalová stupnice. Dělí se dále na dvě skupiny, z nichž nejčastěji se používá plynule navazující stupnice, která lze sestavit několika metodami:

- Konstantní stupnice
- Pravidelně rostoucí (pravidelně klesající) stupnice
- Nepravidelná stupnice

[11], [12], [19]

2.2 Seznámení se s technologiemi

Bylo důležité se seznámit s nastavením a funkcemi dnešních technologií na poli mapových serverů. Při analýze možností mapových serverů byly zahrnuty programové produkty spadající do oblastí komerční sféry i sféry open-source. Zjištěné informace byly využity při výběru vhodných programových prostředků k realizaci i během celého procesu tvorby programové knihovny.

2.3 Výběr technologií a programovacích jazyků

Při objektivě orientovaném návrhu i pro realizaci knihovny bylo nutné vybrat vhodný jazyk. Programovací jazyk potřebný k vytvoření konečné podoby programové knihovny i klientské aplikace bude muset umožňovat objektivě orientované programování a také možnosti vytvářet aplikace použitelné v prostředí internetu. Budou využity jazyky skriptovací, modelovací a značkovací. Vhodný výběr programových produktů i programovacích jazyků je důležitou částí práce, jejíž výsledek se projeví při realizaci.

Byly vybrány tyto technologie a jazyky:

2.3.1 Unifikovaný modelovací jazyk – UML

Na počátku každého projektu je nutné, než se přistoupí k realizaci, vytvořit návrh. Tvorba objektivě orientovaného návrhu je jeden z hlavních cílů této diplomové práce. Jazyk UML se využívá k tvorbě návrhu, vizualizace, upřesnění a dokumentování různých typů aplikací a k tomu využívá stejnou formální syntaxi, která umožňuje snadné a rychlé sdílení informací mezi návrhářem, vývojářem a analytikem.

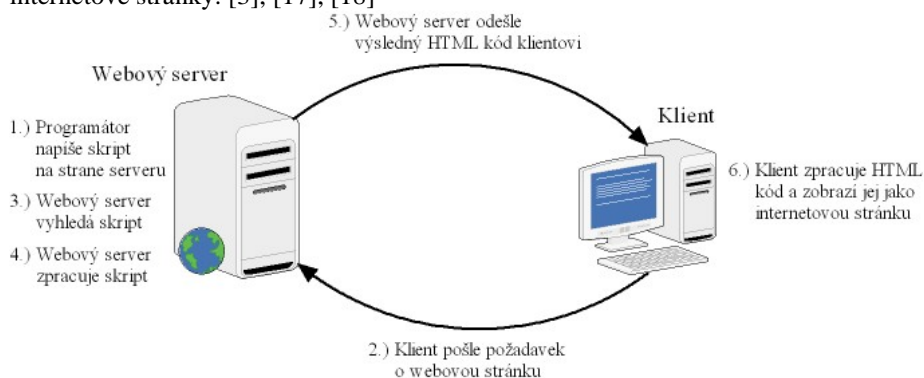
UML je Standard pro zápis softwarových projektů.

UML je jazyk pro vizualizaci, specifikaci, stavbu a dokumentaci softwarových systémů. Návrh takového systému vyžaduje pohled z různých perspektiv. Každý člen týmu různého zaměření pracující na projektu, pohlíží na architekturu systému jiným způsobem a v jiný čas. Systémovou architekturu je možné popsat pěti překrývajícími se pohledy. [2], [6], [10]

2.3.2 Skriptovací jazyk PHP5

PHP je skriptovací jazyk umožňující programování skriptů na straně serveru. Je určen k tvorbě dynamických internetových stránek. Jazyk PHP umožňuje použití procedurálního programování i objektově orientované programování. Pro potřeby diplomové práce byl zvolen tento skriptovací jazyk ve verzi PHP5, protože již podporuje objektově orientované programování a je možné pomocí tohoto jazyka komunikovat s prostředím internetového mapové serveru.

Jednou z možností jak zpracovat skripty jazyka PHP je mít nainstalován a spuštěn webový server s podporou PHP. Proces zpracování PHP skriptů a komunikace klienta s webovým serverem lze ukázat na několika krocích. Nejdříve musí programátor vytvořit zdrojový kód v jazyce PHP a uložit jej na běžícím webovém serveru. Ve chvíli kdy uživatel bude požadovat zobrazení internetové stránky, klient odešle na webový server požadavek o internetovou stránku. Webový server přijme požadavek klienta a vyhledá správný PHP skript. Nalezený skript zpracuje na straně webového serveru a klientovi odešle již výsledek ve formě HTML kódu. Klient přijme HTML kód, zpracuje jej a zobrazí ho v uživatelské internetové prohlídce ve formě internetové stránky. [3], [17], [18]



Obr 1 Proces získání internetové stránky zpracováním PHP skriptu

2.3.3 Webový server

Úkolem webového serveru je naslouchat na některém portu (obvykle 80 pro HTTP a 443 pro HTTPS) a odpovídat na požadavky klientů. Požadavek klienta lze rozdělit do tří částí:

- Příkaz protokolu HTTP, URI a verzi protokolu HTTP
- Hlavičkové informace s údaji o klientovi a o datech, které posílá

- Tělo entity obsahující data posílána klientem

Webový server zpracuje požadavek klienta a zašle odpověď, kterou se skládá ze tří částí:

- Verze protokolu HTTP, třímístný stavový kód požadavku a text popisující stavový kód
- Hlavičkové informace
- Tělo odpovědi

Při práci na vývoji programové knihovny byl zvolen HTTP server Apache ve verzi Apache/2.0.59 (Win32). Použití webového serveru Apache nabízí velké množství výhod a důvodů pro jeho nasazení.

Lze uvést jedny z nejdůležitějších důvodů proč využít právě služby serveru Apache:

- Apache je vysoce nastavitelný webový server s modulární architekturou
- Apache je vyvíjen jako open-source projekt
- Apache výborně funguje se skriptovacími jazyky jako jsou Perl, PHP a další
- Apache funguje na operačních systémech Linux, Unix, Mac OS X, MS Windows a dalších

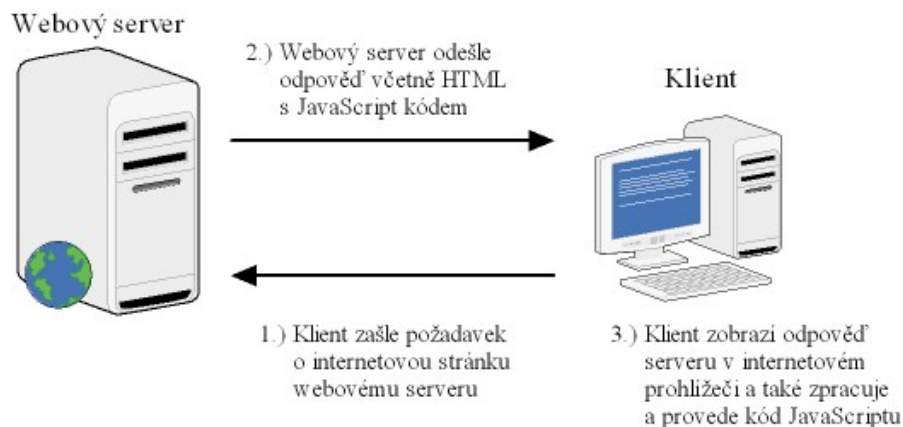
[9], [15]

2.3.4 Mapový server University of Minnesota

UMN MapServer je softwarová CGI aplikace umožňující publikaci prostorových dat na Internetu. Je vyvíjen jako open-source řešení mapového serveru na Minnesotské univerzitě. UMN Mapserver představuje také vývojové prostředí k tvorbě aplikací zaměřených na prostorová data. Ke komunikaci s prostředím UMN MapServeru a k vývoji aplikací lze využít skriptovací a programovací jazyky jako jsou : PHP, Python, Perl, Ruby, Java, a C#. Pro účely diplomové práce byl zvolen mapový server UMN MapServer, kde hlavními důvody pro tuto volbu byla skutečnost, že UMN Mapserver je poskytován zdarma a obsahuje silné prostředí pro vývojáře a také umožňuje při vývoji použít skriptovací jazyk PHP. [13]

2.3.5 Skriptovací jazyk Javascript

JavaScript je skriptovací objektově orientovaný jazyk fungující na straně klienta. Javascript je jazyk interpretovaný, což znamená že zdrojový kód se nekompiluje, ale zůstává v nezměněné formě. Zdrojový kód javascriptu se vkládá přímo do kódu internetových stránek a zpracováván je internetovým prohlížečem u klienta. Pomocí tohoto jazyka lze psát interaktivní internetové stránky a internetové aplikace s dynamicky měnícími se prvky a obsahem. Univerzální jádro jazyka je součástí všech nejvíce se používaných internetových prohlížečů.



Obr 2 Zpracování skriptu na straně klienta

JavaScript má i omezení a nelze pomocí něj dělat vše, ale naopak pokud vývojář zná dobře jazyk i s jeho omezeními, shledá v JavaScriptu mocný nástroj k tvorbě dynamického webu. [8], [22]

2.3.6 Značkovací jazyk XHTML

XHTML je značkovací jazyk používající se k tvorbě hypertextových dokumentů. XHTML je přepracovaný jazyk HTML 4 do jazyku XML 1.0. Jazyk HTML se již dál nevyvíjí a všechno úsilí se nyní zaměřuje na jazyk XHTML. Jazyk XHTML je vyvíjen konsorciem W3C. Konsorcium W3C je organizace, která vytváří standardy, které využívají výrobci internetových prohlížečů při vývoji produktů. Existuje několik verzí jazyka XHTML. Pro potřeby diplomové práce byla využita verze XHTML 1.0 Transitional. Při používání jazyka XHTML je možno se rozhodnout mezi několika druhy a verzemi [20]

2.3.7 Kaskádové styly – CSS

Kaskádové styly CSS představují samostatný jazyk rozšiřující jazyk (X)HTML tím, že umožňuje vytváření stylů k zajištění jednotného vzhledu publikovaného obsahu. Pomocí CSS může uživatel předefinovat formátování jakékoli značky jazyka (X)HTML. Hlavním záměrem CSS je oddělení vzhledu dokumentu od obsahu. Konsorcium W3C stojící za vývojem CSS stále pracuje na zlepšení jazyka a do dnešní doby bylo vydáno několik verzí. [16]

2.4 Objektově orientovaný návrh

Po ujasnění poznatků z analýzy bylo přistoupeno k návrhu programové knihovny. K tomuto účelu byl využit modelovací jazyk UML v němž byl vytvořen model tříd důležitý pro následnou fázi implementace. S ohledem na budoucí vývoj projektu byl při tvorbě objektově orientovaného návrhu kladen velký důraz na rozšiřitelnost. Také

na srozumitelnost vytvořeného modelu bylo dbáno z toho důvodu, aby mu v rámci implementace snadno porozuměli i lidé nezapojení do návrhu. Vytvořený návrh bude využíván při realizaci programové knihovny pomocí programovacího jazyka.

2.5 Realizace programové knihovny

Implementační fáze obsahuje realizaci objektově orientovaného návrhu programové knihovny v programovacím jazyce. Byl zvolen jazyk PHP5, který umožňuje tvorbu aplikací v prostředí Internetu pomocí objektově orientovaného programování. Při realizaci byly vytvořeny třídy podle navrženého modelu tříd. V této fázi práce bylo plně využito možností jazyka PHP5. Realizovány byly algoritmy pro všechny části objektově orientovaného návrhu, které řeší:

- zpracování vstupních a výstupních dat
- výpočtů statistických ukazatelů
- nastavení klasifikačních metod
- nastavení typů legendy
- generování barev a barevných spojitých palet

2.5.1 Správa a zpracování vstupních dat

Pro přístup k vstupním datům a jejich zpracování byla vytvořena třída DataLoader. Úkolem třídy je načtení vstupních dat, uložení a jejich zpřístupnění ostatním třídám. Hlavním důvodem pro vytvoření takovéto třídy bylo, aby se načtení vstupních dat z datového zdroje provedlo pouze jednou, a dále s nimi bylo nakládáno již v rámci programové knihovny bez nutnosti opětovného načtení vždy, když některá metoda určité třídy vyžaduje pro správnou funkci vstupní data. Tímto byly eliminovány časové prodlevy způsobené komunikací mezi aplikací s implementovanou programovou knihovnou a vnějším datovým zdrojem. Pro situace, kdy jsou k zapotřebí nová data obsahuje třída DataLoader metodu pro opětovné načtení a zpracování vstupních dat.

2.5.2 Statistické ukazatele

Třída starající se o výpočty statistických ukazatelů ze vstupních dat se jmenuje BasicStatistics. Důvodem vytvoření této třídy byla nedostupnost statistických funkcí nabízených jazykem PHP5. Byly implementovány algoritmy pro výpočet základních statistických ukazatelů popisujících statistický soubor hodnot. Jednotlivé hodnoty jsou nutné pro další výpočty v jiných třídách, zabývajících se klasifikačními metodami.

2.5.3 Klasifikační metody

Navržena programová knihovna je tvořena čtyřmi hlavními pilíři. Jedním z nich je část zabývající se klasifikačními metodami. Základní třídou této části je abstraktní

třída `ClassificationMethod`, která obsahuje metody zajišťující jednotnost prvků v podtřídách, protože všechny abstraktní metody definované v této třídě musí mít každá podtřída implementovány.

Takovýto hierarchický návrh umožňující rozšiřitelnost o nové klasifikační metody. Podtřídy abstraktní třídy `ClassificationMethod` představují jednotlivé klasifikační metody. Názvy nejdůležitějších metod objektu zůstávají, díky dědičnosti, pro každou klasifikační metodu stejné. Tímto je zajištěna přehlednost programového kódu a usnadněná použitelnost při vývoji aplikací.

2.5.4 Typy legendy

Tato část programové knihovny má implementované typy legendy pro vybranou metodu kartografického znázornění tématického obsahu a je reprezentována jednou hlavní abstraktní třídou a dále třídami, které jsou z hlavní třídy odvozeny a představují jednotlivé typy legendy. Použitím abstraktní třídy je zajištěna jednotnost názvu vlastností a metod pro všechny podtřídy. Podtřídy musí mít implementovány všechny abstraktní metody a vlastnosti abstraktní nadřazené třídy. Systém hierarchie tříd je navržen tak, aby bylo možné vytvářet nové třídy s jejich snadným a efektivním začleněním. Tímto je zajištěna vysoká rozšiřitelnost, která představuje nejdůležitější vlastnost pro budoucí rozvoj projektu.

2.5.5 Zpracování barev

Programová knihovna není navržena pouze pro výpočetní algoritmy statistických ukazatelů, klasifikačních metod a typů legendy, ale také obsahuje algoritmy pro tvorbu barev a barevných spojitých palet. Třídy starající se o generování barev a jejich zpracování se nazývají `Color` a `ColorRamp`. Tyto třídy vznikly z potřeby přiřazení barvy jednotlivým vypočteným intervalům nebo prvkům datového souboru pro účely vizualizace v klientské aplikaci a také pro rozšíření funkcí poskytovaných programovou knihovnou.

2.6 Klientská aplikace

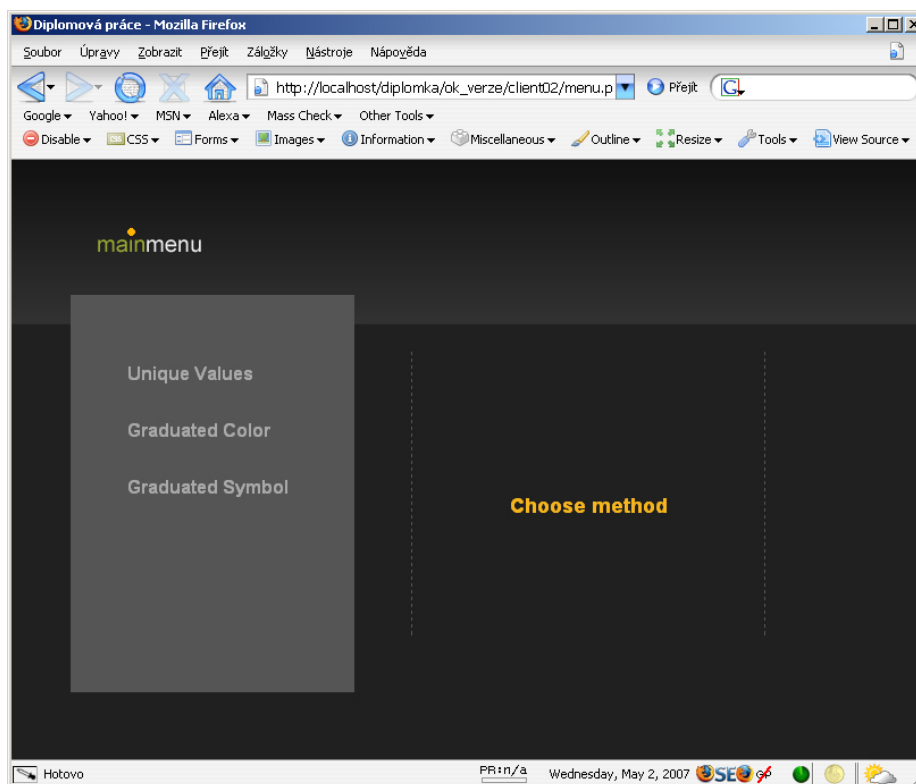
Klientská aplikace byla vytvořena z důvodu demonstrace celého rozsahu funkčnosti navržené programové knihovny. Představuje možnost jak lze jednotlivé třídy použít a celý systém propojit s vnějším prostředím aplikace fungující na bázi mapového serveru. Nezanedbatelnou součástí klientské aplikace je grafické uživatelské rozhraní, obsahující různorodé ovládací prvky usnadňující uživatelům práci. Ovládací prvky umožňují interakci systému s uživatelem. Výsledné výstupní hodnoty jsou přehledně zobrazeny v grafickém prostředí. Pro komunikaci a výměnu dat mezi klientskou aplikací a jiným systémem bylo vytvořeno API.

Cílem bylo navrhnout grafické uživatelské prostředí pro klientskou aplikaci, které by fungovalo v prostředí internetového prohlížeče a obsahovalo všechny ovládací prvky, potřebné pro využívání funkcí dodaných třídami programové knihovny. Důraz byl

kladen na vytvoření grafického prostředí, které by bylo plně interaktivní, obsahovalo intuitivní ovládání a umožňovalo předvolení požadovaných parametrů.

2.6.6 Hlavní menu

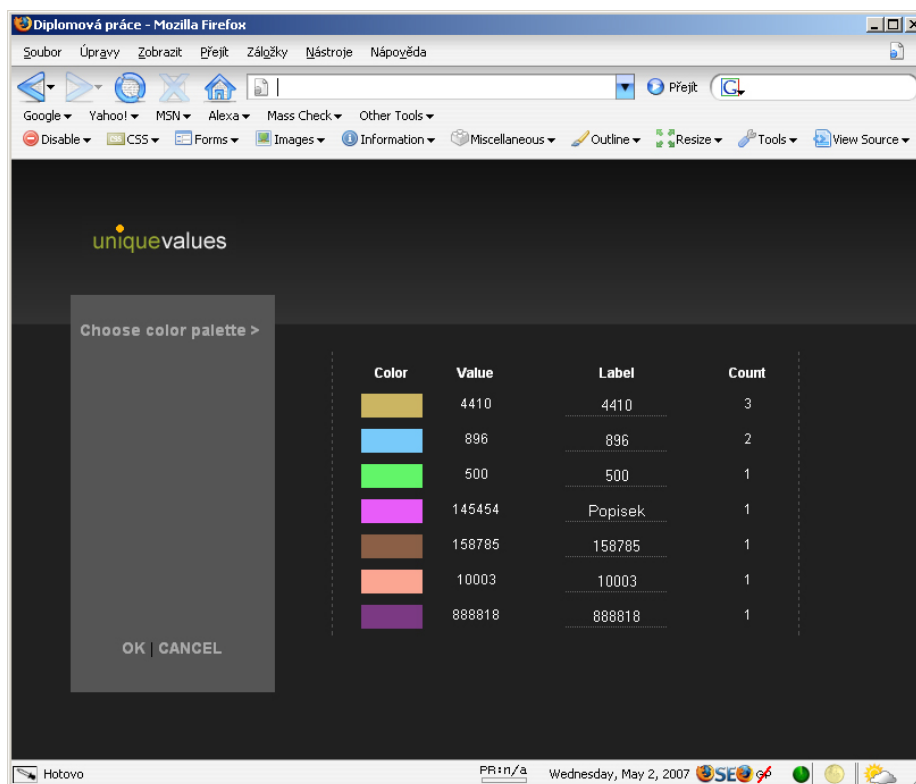
Jednou z hlavních částí klientské aplikace je úvodní obrazovka s hlavním menu, umožňujícím výběr z nabízených typů legendy.



Obr 3 Klientská aplikace - hlavní menu

2.6.7 Specifická hodnota

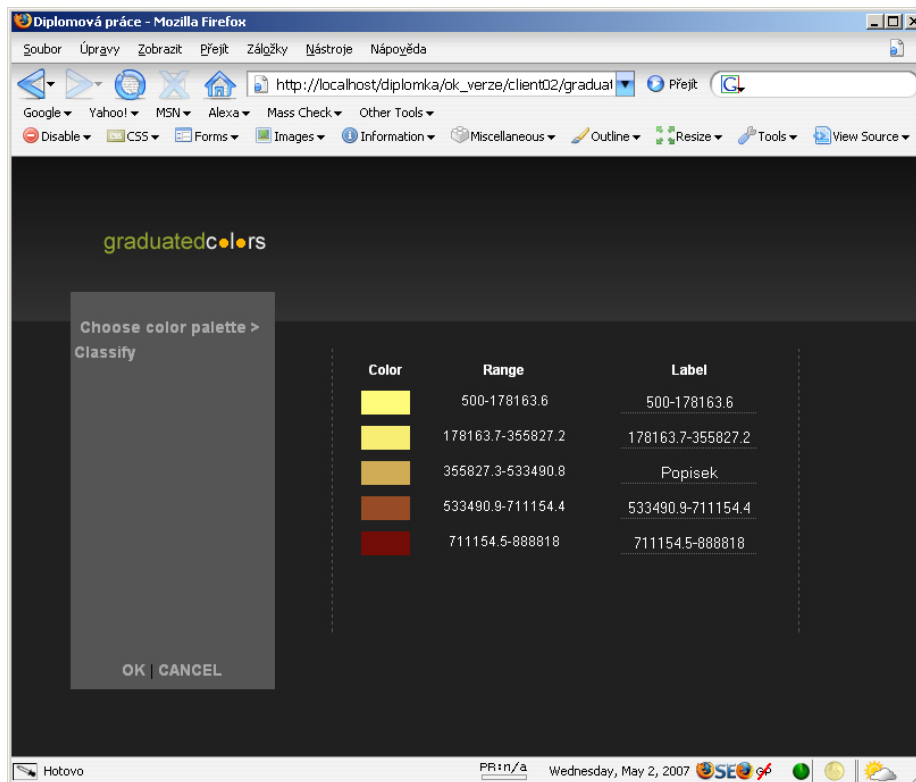
V této části klientské aplikace se uživateli zobrazí všechny parametry důležité k nastavení typu legendy specifická hodnota. Uživatel má možnost výběru barevné palety z níž se nastaví barvy pro jednotlivé specifické hodnoty. Také má možnost si pro každou specifickou hodnotu nastavit popisek.



Obr 4 Klientská aplikace - specifická hodnota

2.6.8 Stupňované barvy

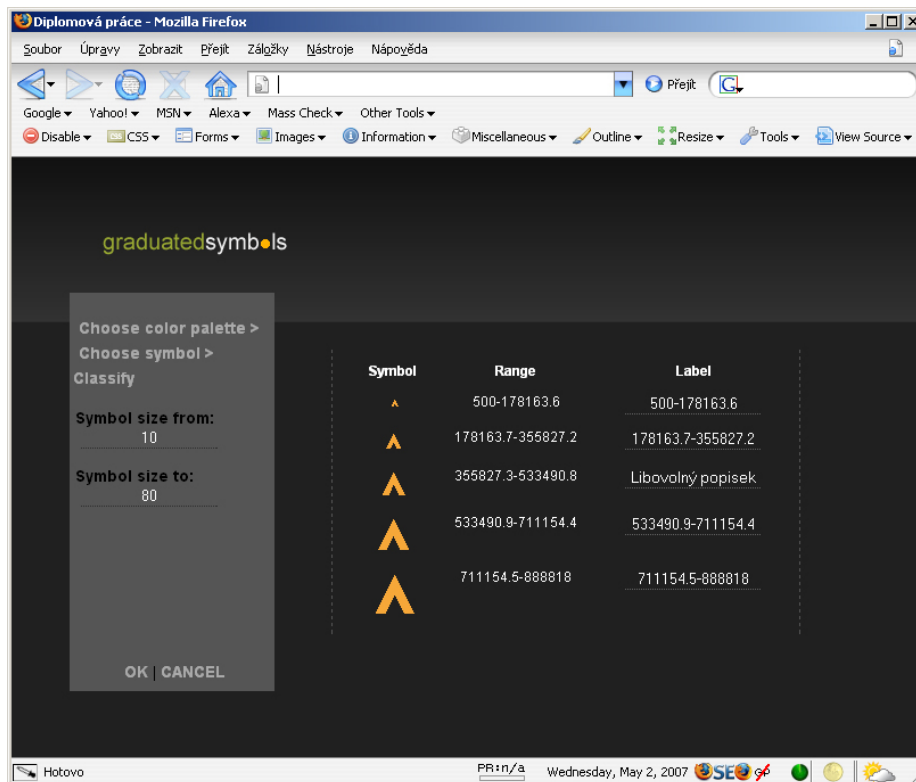
Část klientské aplikace věnující se typu legendy spojité barvy nabízí uživateli graficky zobrazeny parametry pro tento typ legendy. Uživatel má možnost si vybrat barevnou spojitou paletu z celé škály nabízených barevných spojitých palet. Dále má možnost změny popisku pro každý interval a také tento typ legendy umožňuje přejít do nastavení klasifikačních metod.



Obr 5 Klientská aplikace - spojité barvy

2.6.9 Stupňované symboly

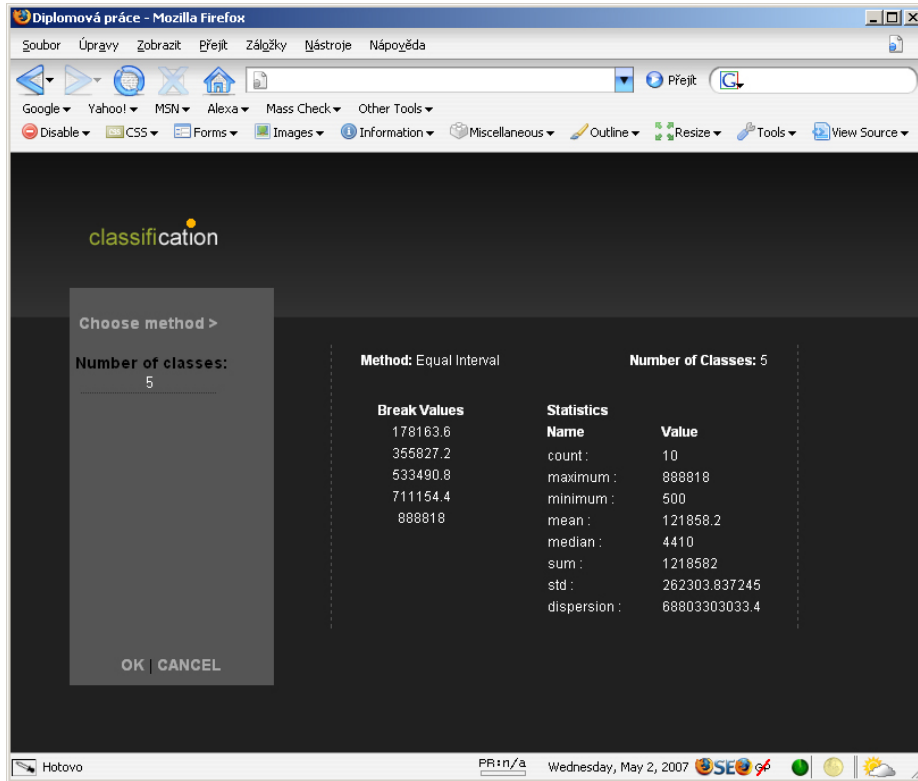
U tohoto typu legendy se uživateli zobrazí odstupňovaný symbol ve vybrané barvě, vypočtené intervaly a libovolně změnitelný popisek. Je možno dále nastavovat spodní a horní hranici z níž se počítají velikosti pro symboly připadající jednotlivým intervalům. Lze provést výběr symbolů. Symboly jsou generovány přímo mapovým serverem a jejich definice je uložena v externím textovém souboru nazývaném symbol list. Dále má uživatel možnost si měnit barvu symbolu a také využít části klientské aplikace zabývající se nastavením klasifikačních metod.



Obr 6 Klientská aplikace - stupňované symboly

2.6.10 Klasifikační metody

Tato část klientské aplikace se zabývá nastavením a výběrem klasifikačních metod. Uživatel má zde možnost zjistit hodnoty statistických ukazatelů, nastavit počet tříd a také vybrat požadovanou klasifikační metodů. Po výběru klasifikační metody a se podle nastaveného počtu tříd vypočtou a přehledně graficky zobrazí jednotlivé zlomové hodnoty, z nichž se poté vytváří intervaly.



Obr 7 Klientská aplikace - Klasifikační metody

2.6.11 Závěr

Na počátku práce byly definovány čtyři hlavní cíle, které byly splněny. Prvním cílem bylo seznámit se s dostupnými programovými prostředky používané v geoinformatice a zjistit jaké možnosti nabízejí z hlediska nastavení typů legendy využívané pro metody kartografického znázornění tematického obsahu a klasifikačních metod. Pozornost byla věnována nastavení typů legendy používaných u metody kartogramů a nastavením klasifikačních metod využívaných u těchto typů legendy. Pro tuto analýzu byly vybrány dva často používané programové produkty zastupující jak sféru komerční tak sféru open-source. Výsledky této analýzy pak posloužily v rámci přípravy objektivně orientovaného návrhu řešení jako vstupní informace.

Po analýze současného stavu programových produktů následovalo splnění jednoho ze dvou nejproblematictějších cílů a to návrh modelu tříd programové knihovny. Byly definovány čtyři hlavní tematické části, které tvoří páteř celého objektivně orientovaného návrhu. Tyto části byly navrženy pro přístup a následné zpracování vstupních dat, výpočet statistických ukazatelů, správu a nastavení typů legendy a v neposlední řadě nastavení klasifikačních metod. Hlavní důraz při objektivně orientovaném návrhu byl kladen na velkou a snadnou rozšiřitelnost o nové možnosti, která představuje základ pro budoucí vývoj. Řešením tohoto problému byl návrh některých z hlavních částí pomocí abstraktních tříd, které zajistily jednotnost prvků ve všech jejich podtřídách. Tímto byl vytvořen robustní model tříd umožňující snadnou tvorbu a začlenění nových tříd do existujícího návrhu, aniž by byla ohrožena jeho přehlednost a použitelnost.

Realizace objektivně orientovaného návrhu ve vybraném programovacím jazyce je obsahem třetího z cílů této práce, který představuje stěžejní krok, jenž ověřuje nasaditelnost v reálném systému. Požadavkem při výběru programovacího jazyka bylo, aby umožňoval použití objektivně-orientovaného programování k tvorbě aplikací v prostředí Internetu. Byl vybrán jazyk PHP5, který je jeden z nejpoužívanějších jazyků pro tuto oblast. Pomocí jazyka PHP5 byly realizovány všechny třídy navržené v modelu tříd programové knihovny. Každá třída je tvořena samostatným souborem, čímž je zachována přehlednost i na úrovni souborového systému. Vytvořené třídy byly využity ve čtvrtém cíli této diplomové práce.

Posledním úkolem práce bylo vytvořit klientskou internetovou aplikaci, která by měla implementovanou programovou knihovnu a demonstrovala by celý její rozsah funkčnosti. Bylo navrženo grafické uživatelské rozhraní určené pro uživatele, kterým je jeho prostřednictvím umožněno provádět nastavení a výběr implementovaných typů legendy a také klasifikačních metod. Grafické prostředí klientské aplikace bylo tvořeno s ohledem na intuitivní ovládání, přehledné rozmístění ovládacích prvků a hlavně na interaktivní komunikaci uživatele s aplikací. Pro vývojáře klientská aplikace poskytuje funkci pro oboustrannou komunikaci s externími systémy. Pro tvorbu klientské aplikace bylo třeba využít mimo jazyka PHP5 také možností dalších skriptovacích a značkovacích jazyků jako jsou XHTML, JavaScript a CSS.

Všechny čtyři cíle práce byly splněny a výsledné řešení představuje návrh a realizaci programové knihovny včetně kompletní dokumentace. Jelikož se jedná o projekt

open-source je možno pro budoucí vývoj celé řešení umístit do některého z internetových katalogů open-source projektů, které poskytují prostor pro vývojáře svobodného softwaru.

S ohledem na rozsah problematiky tvorby tématických map, nastavení typů legendy, určování intervalů velikostních stupnic a statistických způsobů vytváření stupnic se otevírá široká škála možností dalšího rozvoje, jak výsledků této diplomové práce, tak této oblasti obecně. Podobná řešení jsou v současnosti doménou spíše komerčních desktopových aplikací, avšak vzhledem k rychlému rozvoji internetových technologií, se nabízí místo pro nasazení takovýchto pokročilých prvků digitální kartografie v oblasti internetových aplikací. Vznik silných internetových aplikací, poskytujících takovéto služby uživatelům bez nutnosti instalace na jednotlivé počítače a s možností nepřetržitého využívání těchto služeb odkudkoli a kdykoli prostřednictvím internetového prohlížeče, představuje další stupeň ve vývoji počítačových aplikací.

Reference

1. BERARD, Edward V.. *Basic Object-Oriented Concepts* [online]. August 23, 1998, August 23, 1998. [cit. 2007-03-07]. Dostupný z WWW: <<http://www.toa.com/pub/oobasics/oobasics.htm>>.
2. BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivar. *The Unified Modeling Language User Guide*. J. Carter Shanklin; Simone R. Payment. 1st edition. Massachusetts : Addison-Wesley Longman Inc., 1998. 512 s. ISBN 0-201-57168-4.
3. COGGESHALL, John. *PHP 5 unleashed*. 1st edition. Indianapolis : Sams Publishing, 2005. 819 s. ISBN 0-672-32511-X.
4. DAHL, Ole-Johan, NYGAARD, Kristen. *How Object-Oriented Programming Started* [online]. [cit. 2007-03-04]. Dostupný z WWW: <http://heim.ifi.uio.no/~kristen/FORSKNINGSODOK_MAPPE/F_OO_start.html#Stopp>.
5. DRBAL, Pavel. *Objektové myšlení* [online]., Stránku naposledy modifikoval Lubos Pavlicek, 22.06.2005, 21:53 [cit. 2007-03-09]. Dostupný z WWW: <<http://objekty.vse.cz/Objekty/ObjektoveMysleni>>.
6. FOWLER, Martin, KENDALL, Scott. *UML Distilled : A Brief Guide to the Standard Object Modeling Language*. 2nd edition. Massachusetts : Addison Wesley Longman, Inc., 1999. 224 s. ISBN 0-201-65783-X.
7. GILMORE, W. Jason. *Velká kniha PHP5 & MySQL : Kompendium znalostí pro začátečníky i profesionály*. Ing. Pavel Kristián; RNDr. Jan Pokorný. 1. vyd. Brno : Zoner Press, 2005. 711 s. ISBN 80-86815-20-X.
8. GOODMAN, Danny. *JavaScript Bible : Gold edition*. New York : Hungry Minds, Inc., 2001. 1511 s. ISBN 0-7645-4718-6.
9. KABIR, Mohammed J. *Apache Server 2 Bible*. 1st edition. New York : Hungry Minds, Inc., 2002. 793 s. ISBN 0-7645-4821-2.
10. KANISOVÁ, Hana, MULLER, Miroslav. *UML srozumitelně*. Ivo Magera. 1. vyd. Brno : Computer Press, 2004. 158 s. ISBN 80-251-0231-9.
11. KAŇOK, Jaromír. *Tématická kartografie*. 1. vyd. Ostrava : Ostravská Univerzita v Ostravě, 1999. 318 s. ISBN 80-7042-781-7.
12. KUCHEJDA, Libor. *Implementace klasifikačních metod v prostředí ESRI MapObjects*. Ostrava : Vysoká škola báňská - Technická univerzita Ostrava, 2000. 45 s. , Výpis části programové jednotky klasifiklib. Vysoká škola báňská - Technická univerzita Ostrava. Vedoucí diplomové práce Ing. Pavel Hrbáček.
13. Regents of the University of Minnesota. *MapServer : Welcome to MapServer* [online]. 1996-2007 , last modified 2007-03-18 20:52 [cit. 2007-03-20]. Dostupný z WWW: <<http://mapserver.gis.umn.edu/>>.

14. SCHMULLER, Joseph. *Myslíme v jazyku UML : Knihovna programátora*. Petr Somogyi. 1. vyd. Praha : Grada Publishing, spol. s.r.o., 2001. 360 s. ISBN 80-247-0029-8.
15. SPAINHOUR, Stephen, ECKSTEIN, Robert. *Webmaster v kostce : Pohotová referenční příručka*. Jiří Procházka; Jiří Veselý. 1. vyd. Praha : Computer Press, 1999. 504 s. ISBN 80-7226-251-3.
16. TEAGUE, Jason Cranford. *DHTML a CSS pro World Wide Web : Praktická vizuální příručka*. Boris Lehečka; Jan Gregor. 3. vyd. Praha : SoftPress, 2005. 522 s. ISBN 80-86497-77-1.
17. The PHP Group. *Historie PHP a souvisejících projektů* [online]. 2005 , Last updated: Sun, 20 Feb 2005 [cit. 2007-03-17]. Dostupný z WWW: <<http://php.mirror.camelnetwork.com/manual/cs/history.php#history.php>>.
18. The PHP Group. *What can PHP do?* [online]. Last updated: Wed, 14 Mar 2007 [cit. 2007-03-17]. Dostupný z WWW: <<http://www.php.net/manual/en/intro-whatcando.php>>.
19. VOŽENÍLEK, Vít. *Aplikovaná kartografie I. : Tématické mapy*. Vladimír Dostál. 2. vyd. Olomouc : Univerzita Palackého v Olomouci, 2001. 187 s. ISBN 80-244-0270-X.
20. W3C. *W3C Technical Reports and Publications* [online]. 1994-2006 , \$Date: 2007/03/16 15:00:33 \$ [cit. 2007-03-19]. Dostupný z WWW: <<http://www.w3.org/TR/>>.
21. WELLING, Luke, THOMSON, Laura. *PHP a MySQL : Rozvoj webových aplikací*. Miroslav Klíma; Jan Kuklínek. 3. aktualiz. vyd. Praha : SoftPress s.r.o., 2005. 830 s. ISBN 80-86497-83-6.
22. ZAKAS, Nicholas C. *Professional JavaScript™ for Web Developers*. 1st edition. Indianapolis, Indiana : Wiley Publishing Inc., 2005. 674 s. ISBN 0-7645-7908-8.