

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## APLIKACE PRO TVORBU MAP Z DAT PROJEKTU OPENSTREETMAP

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN LATTA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# APLIKACE PRO TVORBU MAP Z DAT PROJEKTU OPENSTREETMAP

MAP CREATION APPLICATION FROM OPENSTREETMAP PROJECT DATA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN LATTA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ NOVOTNÝ

BRNO 2013

## **Abstrakt**

Tato práce se zabývá vývojem aplikace, která umožňuje vytvářet mapy z dat projektu OpenStreetMap a měnit jejich vzhled. Práce obsahuje teoretický úvod do kartografie, informace o projektu OpenStreetMap a vykreslovací knihovně Mapnik. Hlavní náplní textu je návrh a popis implementace samotné aplikace včetně závěrečného testování a vyhodnocení.

## **Abstract**

This bachelor's thesis deals with the development of application that allows you to create maps from OpenStreetMap project data and change their appearance. The work contains an introduction to cartography, information about OpenStreetMap project and rendering library Mapnik. The main focus of the text is a description of the design and implementation of the application itself, including the final testing and evaluation.

## **Klíčová slova**

OpenStreetMap, vykreslování map, XML stylesheet, Mapnik, kartografie, Qt aplikace.

## **Keywords**

OpenStreetMap, map rendering, XML stylesheet, Mapnik, cartography, Qt application.

## **Citace**

Martin Latta: Aplikace pro tvorbu map z dat projektu OpenStreetMap, bakalářská práce, Brno, FIT VUT v Brně, 2013

# Aplikace pro tvorbu map z dat projektu OpenStreet-Map

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Tomáše Novotného . Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Latta  
15. května 2013

## Poděkování

Chtěl bych poděkovat svému vedoucímu panu Ing. Tomáši Novotnému za to, že mi poskytoval odbornou pomoc při řešení této bakalářské práce.

© Martin Latta, 2013.

*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

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Mapy</b>	<b>4</b>
2.1	Mapy v průběhu historie	4
2.2	Geoinformatika	4
2.2.1	GIS	5
2.3	Klasifikace map	5
2.4	Prvky map	6
2.4.1	Mapové měřítko	6
2.4.2	Kartografické zobrazení	6
2.4.3	Mapové symboly	7
2.5	Zkreslení map	7
2.6	Souřadnicové systémy	8
2.6.1	WGS-84	8
2.6.2	UTM	8
2.6.3	S-JTSK	8
<b>3</b>	<b>OpenStreetMap</b>	<b>9</b>
3.1	Formát dat	9
3.2	OpenStreetMap API	10
3.3	Vykreslovací a stylovací nástroje	10
3.3.1	Mapnik	10
3.3.2	TileMill	11
3.3.3	Kartograph	11
3.3.4	Osmarender	11
3.3.5	MapCSS	11
3.3.6	Cartagen a GSS	12
3.3.7	MapOSMatic	12
<b>4</b>	<b>Návrh aplikace</b>	<b>13</b>
4.1	Účel aplikace	13
4.2	Výběr vykreslovacího nástroje	13
4.3	Výběr programovacího jazyka a vývojového prostředí	14
4.3.1	Qt framework	14
4.4	Mapnik XML stylesheet	15
4.4.1	Definice stylů v XML souboru	15
4.4.2	Definice vrstev v XML souboru	16
4.4.3	Referenční souřadnicový systém	18

<b>5 Implementace</b>	<b>19</b>
5.1 Použité nástroje a třídy	19
5.2 Třída MainWindow	19
5.2.1 Načtení lokální mapy	20
5.2.2 Stahování dat ze serveru OpenStreetMap	20
5.2.3 Práce se styly	20
5.2.4 Nastavení souřadnicové mřížky	20
5.2.5 Nastavení velikosti mapy	20
5.2.6 Vykreslení mapy	21
5.2.7 Export mapy	21
5.3 Třída Render	22
5.3.1 Generování souřadnicové mřížky	22
5.3.2 Korekce měřítka	22
5.4 Třída StyleSheet	23
5.5 Třída StyleXmlParser	23
5.6 Ostatní dialogové třídy	24
5.6.1 Třída EditStyleDialog	24
5.6.2 Třída EditLayerDialog	24
5.6.3 Třída EditStylesheetDialog	24
5.6.4 Třída EditGridDialog	24
5.6.5 Třída EditMapSizeDialog	24
5.6.6 Třída OptionsDialog	25
5.6.7 Třída ExportMapDialog	25
<b>6 Testování</b>	<b>29</b>
6.1 Mapa s jednoduchým stylem	29
6.1.1 Přidání vrstevnic	29
6.2 Standardní OpenStreetMap vzhled	30
6.3 Mapa rozsáhlejší oblasti	31
<b>7 Závěr</b>	<b>33</b>
7.1 Možná rozšíření	33
<b>A Ukázka kódu deklarace</b>	<b>35</b>
<b>B Ukázka testovacích map</b>	<b>36</b>
<b>C Obsah CD</b>	<b>40</b>

# Kapitola 1

## Úvod

Mapy jsou již od nepaměti velmi užitečným nástrojem. Lidé se snažili svět kolem sebe zobrazit na nějaké médium, což jim následně pomáhalo v různých sférách života, jako hospodářství, vojenství, při orientaci a navigaci. S vývojem civilizace se postupně vyvíjela i kartografie, vědní obor zabývající se tvorbou map. V současnosti se bez map prakticky neobejdeme, jejich využití můžeme najít v mnoha oborech lidské činnosti. Vývoj map značně usnadnil nástup výpočetní techniky. Spojením kartografie a informačních technologií vzniká vědní obor nazývaný se geoinformatika.

Dnes je na Internetu k dispozici mnoho on-line mapových aplikací umožňující vyhledávat místa a trasy, nebo jinak manipulovat s mapou, málokteré ale nabízejí svá mapová data volně k dispozici. Mezi zdroje, ze kterých lze otevřeně čerpat geografická data, patří i projekt OpenStreetMap [3].

Aplikace pro tvorbu map, která je výsledkem této práce, využívá jako zdroj dat právě volně dostupné podkladové mapy z projektu OpenStreetMap a vykreslovací nástroj Mapnik k jejich zobrazování. Uživatel má možnost upravit vzhled map podle svých požadavků.

V následující kapitole bude vysvětleno, co to vlastně mapy jsou, co obsahují a jak se dělí. Dále jak se vyvíjela kartografická tvorba a jaký vliv mají v současné době informační technologie na kartografii. V třetí kapitole je čtenář seznámen s projektem OpenStreetMap a některými vybranými nástroji a aplikacemi, které pracují s daty tohoto projektu. Další kapitola představí samotnou Aplikaci pro tvorbu map a pátá kapitola rozebere její implementační detaily. Šestá kapitola se zabývá testováním aplikace a závěrečná kapitola shrne dosažené výsledky a výstupy.

## Kapitola 2

# Mapy

Každý má určitou představu o tom, co jsou to mapy. Dá se říci, že je to model reálného světa, přičemž některé skutečnosti mohou být zanedbány, jiné naopak zdůrazněny. Jedna z mnoha definic nám říká, že mapa je zmenšený a generalizovaný obraz Země (nebo jiných těles) převedený do roviny pomocí matematicky definovaných vztahů (kartografické zobrazení) vyjadřující pomocí smluvených znaků rozmístění, vlastnosti a vztahy přírodních, socioekonomických a technických objektů a jevů [6].

### 2.1 Mapy v průběhu historie

Počátky vývoje kartografie lze objevit už v pravěku ve formě různých náčrtků a kreseb. Nejstarší dochované mapy jsou z dob Mezopotámie, Babylonu a Egypta. Prvotní rozvoj přírodních věd, matematiky a astronomie přispěl i k vývoji kartografie v antickém Řecku a Římě, na východě pak především v Číně a Indii. Ve středověku se v Evropě kartografie z počátku příliš nerozvíjela, naopak v Asii (hlavně v Číně) je zaznamenán značný vývoj. Na Blízkém východě se v této době kartografie rovněž velmi rozvíjela.

Velký zlom přišel s nástupem renesance a zaoceánskými plavbami. Vznikají první atlasy a soubory map. Od 18. století se čím dál svižnější technický pokrok projevoval i ve tvorbě map. Probíhala rozsáhlá státní vojenská mapování, tedy snaha kompletně a detailně znázornit území daných států. Mapy jsou čím dál víc precizní a vznikají velká kartografická díla. Začíná se rozlišovat praktická a teoretická kartografie.

20. století přineslo řadu změn - ofsetový tisk, fotogrammetrie (rozlišování tvarů, měření rozměrů a polohy objektů z fotografií), využívání výpočetní techniky. Díky kosmickým programům je možné mapovat Zemi i z oběžné dráhy, tzv. metoda DPZ – dálkový průzkum Země (satelitní, radarové snímky). Data se získávají se špičkovou přesností, rozmáhá se globální navigace GPS, vliv má i rozvoj Internetu a geoinformatiky [6].

### 2.2 Geoinformatika

Masivní rozšíření počítačových technologií v druhé polovině 20. století se projevilo i v kartografii. Mapy již nejsou pouze v tištěné formě, ale jsou uloženy a vizualizovány v počítačích. Mapy se stávají dynamické a interaktivní. Spojením kartografie s informačními technologiemi vzniká vědní obor geoinformatika. Je to aplikovaná informatika zaměřená na řešení různorodých geografických problémů s důrazem na geografickou polohu objektů [6].



Jedna z definic nám říká, že geoinformatika je vědecký a technický interdisciplinární obor, zabývající se zpracováním a využíváním geodat a geoinformací pro potřeby rozhodování, plánování a správy zdrojů [9].

Geoinformatika v sobě zahrnuje mnoho oblastí výzkumu: získávání digitálních geoinformací přímo z terénu a nebo z dálky, navigační systémy (GPS), analýza a posuzování údajů z dálkového průzkumu Země, geografické informační systémy (GIS), digitální kartografie, vývoj a aplikace geostatistických metod, simulační a prognostické modely, systémy pro podporu rozhodování, trojrozměrná (3D) vizualizace a virtuální realita [4].

### 2.2.1 GIS

Geografické informační systémy [4] hrají v geoinformatice významnou roli. Jsou to informační systémy pracující s geografickou informací. Slouží k získávání, ukládání, zpracování, analýze, modelování a zobrazování geoinformací, které popisují geometrii, topologii, tematiku a dynamiku *geoobjektů*. Každý geoobjekt je pak určen identifikací, prostorovou polohou, atributy, případně i jeho změnami v čase. Geometrie geoobjektů popisuje jejich polohu, tvar a velikost. Topologie geoobjektů určuje jejich vzájemné logické vazby.

Výhoda oproti klasickým tištěným mapám je, že GIS striktně odděluje ukládání geografických dat a jejich prezentaci. Data je tak mnohem jednodušší průběžně aktualizovat. K ukládání geodat se často používají databáze, v současné době především postrelační databáze<sup>1</sup>.

*Datový model* určuje, jakým způsobem jsou geodata uložena. Může být buď *rastrový*, nebo *vektorový*. Rastrový model rozděluje prostor na množinu diskrétních buněk, z nichž každá nese určitou informaci. Nejčastěji se jedná o číslo vyjadřující buď kvantitativní, nebo kvalitativní informaci. Prostor je dělen typicky na pravidelné a stejně velké buňky, ale lze se setkat i s nepravidelným dělením. Rastrová data se nejvíce používají pro vyjádření terénu, fyzikálních veličin, osvětlení, nebo se jedná o letecké fotografie. Hlavní nevýhodou je poměrně velká paměťová náročnost na ukládaná data. Vektorový model uchovává přesnou informaci o poloze objektů a jevů. Vektorová data reprezentují body, čáry (spojení dvou bodů), linie (sekvence čar), polygony (uzavřené linie). K informaci o poloze objektů k nim lze přidat i informaci o vzájemné topologii například formou grafů.

Data, ať už reprezentovaná vektorově či rastrově, mohou být členěna do vrstev podle určitých kritérií, čímž je zajištěna jejich nezávislost a případně možnost jejich skládání a kombinování. Tento přístup se nazývá *vrstevový*. Dalším způsobem je *objektový* přístup, který vychází z principů objektově orientovaného programování. Každý objekt obsahuje svoji geometrii, topologii, atributy a metody. Je možné objekty sdružovat do tříd a vytvářet hierarchické vztahy pomocí dědičnosti [5].

## 2.3 Klasifikace map

Mapy lze třídit podle mnoha kritérií [12]:

- Podle obsahu - topografické, všeobecně zeměpisné, tematické, katastrální, účelové.
- Podle rozsahu zobrazeného území - astronomické mapy (mapy hvězdné oblohy nebo vesmírných těles), mapy Země (celý svět, polokoule, světadíly, státy, regiony, detailní plány měst atd.).

---

<sup>1</sup>Oproti relačním databázím zde platí, že každý záznam má jinou délku a obecně jiný datový typ, takže jsou vhodnější k ukládání geografických dat [4].

- Podle času platnosti mapy - statické, dynamické (zachycují vývoj v čase), retrospektivní (rekonstrukce jevů v minulosti), prognostické (odhad jevu do budoucnosti), genetické (vznik a vývoj jevů).
- Podle účelu - státní správa, věda, kultura, vojenské, orientační, propagační, sportovní, školní, dopravní mapy apod.
- Podle měřítka - malé, střední, velké. Měřítka udávají poměr zmenšení vzhledem k rozměrům v reálném světě. U menších měřítek se může značně promítnout zakřivení Země.
- Podle formy záznamu - digitální, analogové, obrazové (foto-mapy), transparentní (určené pro projekci), reliéfní, tyflomapy (pro slepé), anaglyfové (stereoskopické) mapy.
- Podle způsobu vzniku - původní (vytvořené z dat získaných v terénu), odvozené.

## 2.4 Prvky map

Mapové dílo obsahuje tři základní náležitosti. Jsou to měřítko, kartografické zobrazení (projekce) a symbolizace [7].

### 2.4.1 Mapové měřítko

Mapové měřítko nám určuje, o kolik je mapa menší než realita. Lze ho uvést jako číslo udávající podíl, slovní popis, či v grafické podobě. Číselné měřítko vyjadřuje vztah mezi vzdáleností na mapě a vzdáleností v reálném světě, přičemž jednotky těchto hodnot musí být stejné (a tím pádem ani nemusí být uvedeny). Mapy velkého (například 1:1 000) měřítka jsou přesnější, než mapy malého měřítka (například 1:500 000).

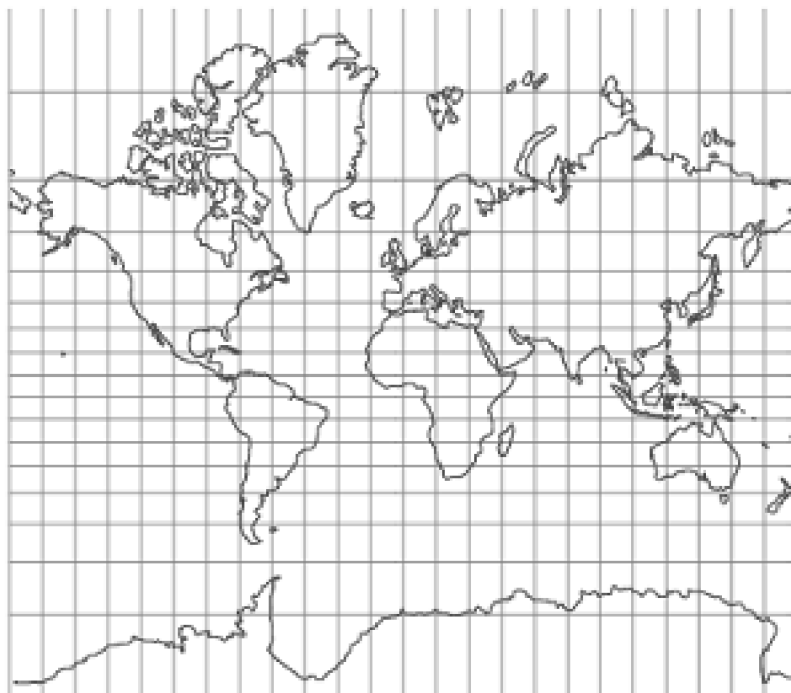
### 2.4.2 Kartografické zobrazení

Kartografické (mapové) zobrazení, nebo také kartografická projekce, je matematický postup používaný k převodu zeměpisných souřadnic na souřadnice rovinné [6]. Tomuto vztahu se říká zobrazovací rovnice. Existuje mnoho typů zobrazení, jež lze dělit podle tvaru nebo polohy zobrazovací plochy, vlastností zkreslení, případně podle tvaru zobrazovacích rovnic.

Základní tři zobrazovací plochy jsou: válcová, kuželová, nebo azimutální. Válcová projekce se používá u celosvětových map, kuželová u rozsáhlých oblastí, jako jsou kontinenty nebo oceány. Azimutální zobrazení se obvykle využívá pro mapy polárních oblastí. Další typy ploch mohou být: nepravé, mnohokuželové, či různě složené a další. Všechny tyto plochy se dotýkají povrchu referenční koule v určitém bodě (nebo bodech). V bodě dotyku je poté zkreslení minimální.

Zkreslení mapového zobrazení se rozlišují podle toho, jakým způsobem ovlivňují údaje na mapě: konformní nezkrusuje úhly, ekvivalentní zachovává poměr ploch, ekvidistantní nezkrusuje vzdálenosti a kompenzační (vyrovnávací) je kompromisní zobrazení [6].

Příkladem zobrazení může být *Mercatorovo zobrazení*, které je válcové a konformní. Válec se dotýká rovníku glóbu a proto se zkreslení zvětšuje směrem k pólům. Některé internetové mapové služby používají tuto projekci (například OpenStreetMap [3], Google Maps [2]).



Obrázek 2.1: Ukázka Mercatorova zobrazení [10].

### 2.4.3 Mapové symboly

Mapové symboly a znaky nesou podstatnou část informací v mapě. Jejich množství a použití v mapě se může velmi lišit podle toho, o jaký typ mapy se jedná. Výběr, vzhled a organizace jednotlivých symbolů bývá u odlišných typů map často standardizována. Znaky na dvojrozměrných mapách jsou bodové, liniové a plošné. Většina map používá kombinace těchto tří základních prvků. Důležitým požadavkem je kontrast symbolů. Ty se mohou lišit svojí velikostí, tvarem, texturou, orientací, barvou, či odstínem šedi [7].

Symboly map lze také tematicky dělit do skupin následovně: fyzickogeografické představující vodstvo, reliéf (kóty, vrstevnice), porosty, půdy, podložní horniny nebo jiné přírodní jevy. Socioekonomické zobrazují města, dopravní sítě, průmyslové a zemědělské objekty, hranice, či jiné socioekonomické jevy. Neméně důležitou roli mají i doplňkové a pomocné prvky jako název, legenda, popis a informace o mapě, tiráž atd.

## 2.5 Zkreslení map

Mapové zobrazení transformuje zakřivený trojrozměrný povrch Země do dvojrozměrné plochy mapy. To samo o sobě nelze prakticky provést, aniž by nedocházelo k určité deformaci a zkreslení jednotlivých rozměrů na mapě (výjimkou je glóbus, ale u toho jsme omezeni určitou velikostí). Čím menší má mapa měřítko, tím větší je její výsledné zkreslení. U takových map, které zabírají rozsáhlá území, nemusí dokonce být měřítko konstantní a může se lišit podle pozice na mapě. Mapy s velkým měřítkem zabírají pouze malý výsek povrchu Země, takže zde bývá zkreslení zanedbatelné.

## 2.6 Souřadnicové systémy

Pro určení polohy je nejdříve potřeba definovat určitý *souřadnicový systém* [4].

Země nemá přesně matematicky popsateľný tvar. Asi nejlépe jej popisuje *geoid*, ale ten má velice složité matematické vyjádření. Pro kartografické účely se využívají jednodušší modely. Zemský povrch je nahrazen *rotačným elipsoidem*<sup>2</sup>, také nazývaný referenční (náhradní) elipsoid. Mezi známé patří například Besselův elipsoid (pro území ČR), NAD83 (území Severní Ameriky), WGS-84 (celosvětový elipsoid), Krasovského elipsoid (Rusko) a další. Pro každý referenční elipsoid je definovaná vlastní souřadnicová soustava.

Souřadná soustava nám umožňuje přesně určit polohu a měřit vzdálenosti mezi objekty v konkrétní jednotce.

### 2.6.1 WGS-84

*World Geodetic System 1984* (WGS-84) je světově uznávaný geodetický standard z roku 1984 neboli systém šířka-délka, používající stejnojmenný referenční elipsoid. Zeměpisná šířka bodu je dána jako úhel, který svírá s rovinou rovníku. Zeměpisná délka je úhel, který svírá s předem definovaným nultým poledníkem (Greenwichský poledník).

### 2.6.2 UTM

*Universal Transverse Mercator* (UTM), neboli Univerzální transverzální Mercatorův systém souřadnic. Princip spočívá v rozdělení povrchu elipsoidu do zón podle poledníků a rovnoběžek a jejich zobrazení do pravoúhlé soustavy. V rámci jedné zóny lze tak jednoduše měřit vzdálenosti, souřadnice platí pouze v konkrétní zóně.

Stejně jako WGS-84 je UTM používán hlavně v globálních polohovacích systémech jako Navstar GPS.

### 2.6.3 S-JTSK

*Souřadnicový systém jednotné trigonometrické sítě katastrální* (S-JTSK) je základní souřadnicový systém používaný na území České republiky a Slovenska. Referenční těleso je Besselův elipsoid. Jedná se o pravoúhlý systém, jednotkou je metr, chyba způsobená projekcí je zanedbatelná.

---

<sup>2</sup>Elipsoid je matematicky definován hlavní a vedlejší poloosou, 1. a 2. excentricitou a zploštěním [6]. Tyto parametry se liší podle použitého druhu elipsoidu a požadované přesnosti. Střed elipsoidu leží ve středu Země a svým natočením se snaží co nejlépe napodobit tvar geoidu.

## Kapitola 3

# OpenStreetMap

OpenStreetMap [3] je projekt, který založil Steve Coast v červenci 2004. Klade si za cíl vytvořit volně šiřitelné mapy celého světa a jejich následnou vizualizaci do různých podob. Geodata jsou získávána převážně z přijímačů GPS nebo z jiných volně dostupných zdrojů. Na vývoji se může podílet jakýkoliv registrovaný uživatel, uchovává se historie změn (stejně jako například u Wikipedie). Geografická data jsou šiřitelná pod licencí *Open Data Commons Open Database License* (ODbL)<sup>1</sup>.

### 3.1 Formát dat

Geografická data jsou uchována ve specifickém souborovém formátu založeném na XML – *OpenStreetMap XML file format* (dále jen OSM XML). Referenční souřadnicový systém je WGS-84 (viz 2.6.1), data jsou uložena v projekci *Spherical Mercator* vycházející z Mercatorova zobrazení (viz 2.4.2).

Data jsou uložena v centrální databázi, rozlišují se následující datová primitiva:

- Uzly (**nodes**) – body lokalizované dvěma souřadnicemi v daném referenčním systému, třetí souřadnice (nadmořská výška) je nepovinný atribut. Jsou používány hlavně k znázornění bezrozměrných prvků, jako například body zájmu, vrcholy hor, stromy, křižovatky atd.
- Cesty (**ways**) – uspořádaný seznam uzlů, reprezentující liniové nebo plošné prvky. Příkladem linií jsou silnice, řeky, ploty, elektrické vedení. Oblasti mohou být lesy, jezera, parky, budovy apod. Polylinie může být otevřená a nebo uzavřená v případě totožného počátečního a koncového bodu. Uzavřená polylinie s výplní je polygon (**area**). Cesty se skládají z minimálně dvou uzlů a maximálně z 2 000. Delší cesty přesahující 2 000 uzlů se proto musí rozdělit.
- Relace (**relations**) – reprezentují logickou nebo geografickou vazbu mezi existujícími uzly a cestami. Nepovinným atributem je **role**, který popisuje roli elementu v relaci. OpenStreetMap obsahuje mnoho typů relací, za zmínku stojí například relace: **multipolygon** spojuje polygony do velkých oblastí, **route** určuje typ dopravní cesty, **boundary** definuje administrativní hranice oblastí.

---

<sup>1</sup>Licence se týká geografických dat uložených v databázi. Dříve používaná licence *Creative Commons* je určena především pro autorská díla a příliš se nehodí pro databáze.

- Atributy (**tag**) – lze je přiřadit k uzlům, cestám a relacím ve formě <klíč>=<hodnota>. Klíč i hodnota mohou nabývat libovolné textové hodnoty. Lze tak definovat k elementům další pomocné informace jako jejich název, typ, případně fyzikální vlastnosti daného objektu.

Struktura OSM XML souboru je následující:

- XML hlavička zavádějící UTF-8 kódování znaků v souboru
- **osm** – element obsahující verzi API a generátor, který vygeneroval daný soubor (případě editační nástroj)
  - element **bounds** jehož atributy vymezují hranice mapové oblasti
  - blok uzlů (elementy **node**) obsahující identifikaci **id**, souřadnice v souřadném systému WGS-84, případně další informace (autor, datum vytvoření apod.)
    - volitelný seznam tagů pro každý uzel
  - blok cest (elementy **way**)
    - seznam referencí na uzly, ze kterých se cesta skládá (elementy **nd**)
    - volitelný seznam tagů pro každou cestu
  - blok relací (elementy **relation**)
    - seznam referencí na členy relace (elementy **member**)
    - volitelný seznam tagů pro každou relaci

## 3.2 OpenStreetMap API

Aktuální verze je API v0.6 nasazená v dubnu 2009. Editiční API je serverová komponenta zpracovávající HTTP dotazy přicházející na server. Zpracovává požadavky na vytvoření, čtení, aktualizování a smazání specifikovaných elementů mapy.

## 3.3 Vykrešlovací a stylovací nástroje

Existuje celá řada programů a nástrojů, které zpracovávají geografická data v odlišných formátech a vytváří tak různorodé mapy. Samotná geografická data často neobsahují informaci o jejich vzhledu. Výsledný vzhled mapy tak lze ovlivnit pomocí *stylovacího souboru* (stylesheet), který přesně popisuje, jak mají jednotlivé prvky mapy vypadat. Záleží na konkrétním nástroji, které formáty dat a stylesheetů podporuje. Některé nástroje jsou komerčního charakteru, jiné zase volně dostupné (například pod licencí GPL). Pár z nich si zde představíme, a to především ty, které mají spojitost s OpenStreetMap.

### 3.3.1 Mapnik

Mapnik [8] je volně dostupná knihovna pro vývoj aplikací se zaměřením na mapy. Nabízí velice precizní vykreslovací funkce pro vytváření uživatelsky přívětivých map z různých geografických dat. Je napsaný v C++, existuje i verze pro Python a je multiplatformní (Windows, Mac OS, Linux). Jako zdrojová data podporuje mnoho formátů, jako ESRI

shapefile<sup>2</sup>, rastr TIFF, PostGIS databáze a další. Pro nás je však nejpodstatnější podpora OpenStreetMap dat.

K definici stylů map využívá XML soubory (tzv. *Mapnik XML stylesheet*) obsahující dvě části. Za prvé to jsou vrstvy (*layers*) obsahující popis vrstev v pořadí, v jakém se budou renderovat – první uvedené se první renderují. Za druhé to jsou styly (*styles*) obsahující pojmenovaná pravidla, která umožňují komplexní nastavení všech grafických aspektů mapy.

### 3.3.2 TileMill

TileMill<sup>3</sup> je prostředí vyvinuté firmou MapBox pro vytváření interaktivních map. K renderování používá Mapnik, k definici stylů využívá stylovací jazyk *CartoCSS*, založený na CSS. Editace stylu se provádí ručně přepisováním CSS stylů. K vzájemnému převodu mezi *CartoCSS* a Mapnik XML slouží preprocesor *Carto* vyvinutý rovněž firmou MapBox.

### 3.3.3 Kartograph

Kartograph<sup>4</sup> je jednoduchý odlehčený framework pro vytváření interaktivních mapových aplikací. Knihovna Kartograph obsahuje dvě části. Jedna ke generování kompaktních map ve vektorovém formátu SVG, druhá k vytváření interaktivních map, které fungují pod celou řadou webových prohlížečů. Jako zdroj dat akceptuje ESRI shapefile, případně i data z OpenStreetMap.

### 3.3.4 Osmarender

Nástroj Osmarender<sup>5</sup> byl vytvořen pouze za účelem renderovat mapy z OpenStreetMap dat. Nejedná se o klasickou aplikaci, ale spíše o množinu souborů, které umožňují provést XSLT transformaci vstupních OSM XML dat do výstupního souboru ve vektorovém formátu SVG.

Vstupem je soubor pravidel neboli *rule file*, který se může lišit podle úrovně požadovaného přiblížení, dále soubor se styly nazvaný *osmarender.xsl*, případně složku obsahující symboly, které chceme rovněž do mapy vykreslit. Stylový soubor obsahuje nastavení a pravidla vykreslování zapsané ve formátu XML, která určují, jaký způsobem budou prvky vykresleny. Také obsahuje část s CSS styly definující vzhled dat. Nezbytný je soubor *data.osm*, který obsahuje mapová data ve formátu OSM XML.

K samotnému procesu renderování je ještě potřeba *XSLT procesor* (program provádějící XSL transformaci), mezi podporované patří například Xalan-J, XRay2, XML starlet, MSXML, xsltproc a další.

Až do března 2012 Osmarender fungoval jako jeden z hlavních vykreslovacích nástrojů pro webové mapy OpenStreetMap. Poté byl vývoj projektu pozastaven.

### 3.3.5 MapCSS

MapCSS<sup>6</sup> je jazyk k definici stylů vytvořený speciálně pro OpenStreetMap. Podobně jako *CartoCSS* (viz. 3.3.2) je založen na bázi CSS. Soubor se styly je tvořen posloupností pravidel, které mají dvě části: jeden či více selektorů k určení, které mapové elementy pravidlo

---

<sup>2</sup>Esri Shapefile je otevřený datový formát pro ukládání vektorových prostorových dat pro geografické informační systémy vyvinutý firmou ESRI [11].

<sup>3</sup>Webové stránky: <http://mapbox.com/tilemill/>

<sup>4</sup>Webové stránky: <http://kartograph.org/>

<sup>5</sup>Webové stránky: <http://wiki.openstreetmap.org/wiki/Osmarender>

<sup>6</sup>Webové stránky: <http://www.mapcss.org/>

ovlivní a jedna či více deklarácí, které říkají, jaké styly se mají aplikovat na odpovídající elementy.

### 3.3.6 Cartagen a GSS

Cartagen<sup>7</sup> je vektorově založený framework k renderování map v nativním HTML 5. Je napsaný v JavaScriptu a běží na klientském počítači. Nabízí možnost renderovat data i z OpenStreetMap. Mapy jsou jednoduše stylovány pomocí GSS (*Geo Style Sheets*), což je také skriptovací jazyk. Cartagen se hodí pro tvorbu dynamických map měnících se v reálném čase.

### 3.3.7 MapOSMatic

MapOSMatic<sup>8</sup> je volně dostupný webový generátor plánů měst. Výstupem je plakát s mapou zvoleného města, která obsahuje legendu, mřížku a vypsaný seznam ulic a významných objektů. Na výběr je mnoho výstupních formátů obrázku, ale jen několik základních OpenStreetMap stylů.

---

<sup>7</sup>Webové stránky: <http://cartagen.org/>

<sup>8</sup>Webové stránky: <http://www.maposmatic.org/>



## Kapitola 4

# Návrh aplikace

Kapitola se zabývá návrhem aplikace – co má program dělat, jak bude probíhat interakce s uživatelem a v neposlední řadě zhodnocení, výběr a následný popis externích knihoven a použitých technologií.

### 4.1 Účel aplikace

Aplikace umožňuje uživateli vytvářet mapy z dat projektu OpenStreetMap. Vzhled mapy lze měnit podle zadaných parametrů. Kromě editace vzhledu dat umožňuje aplikace také přidávat další geografické prvky jako vrstevnice, kilometrovou mřížku či další mapová a textová data i z jiných zdrojů, než jen z OpenStreetMap.

Jedním z cílů aplikace je vytvoření jednoduchého rozhraní, aby uživateli umožnilo efektivně pracovat s mapou. Jedná se tedy o desktopovou aplikaci obsahující hlavní menu, nástrojovou lištu, stavový řádek a hlavní okno. Aplikace musí obsahovat intuitivní nástroje, kterými lze upravovat vzhled mapy. Největší část okna je rezervována pro samotné zobrazení mapy.

Vykreslení mapy se neprovede okamžitě, ale s určitou časovou reží v závislosti na velikosti mapy. Během renderování se hlavní okno aplikace jeví „zamrzlé“ a uživatel nemůže nijak pracovat. Tomuto se lze vyhnout použitím vláken, kdy proces vykreslování mapy běží paralelně s hlavním oknem a uživatel může bez problému dále pracovat s aplikací.

Výsledná mapa se dá exportovat do různých rastrových formátů. Do mapy jsou přidány i další prvky jako je nadpis, měřítko nebo další textové informace.

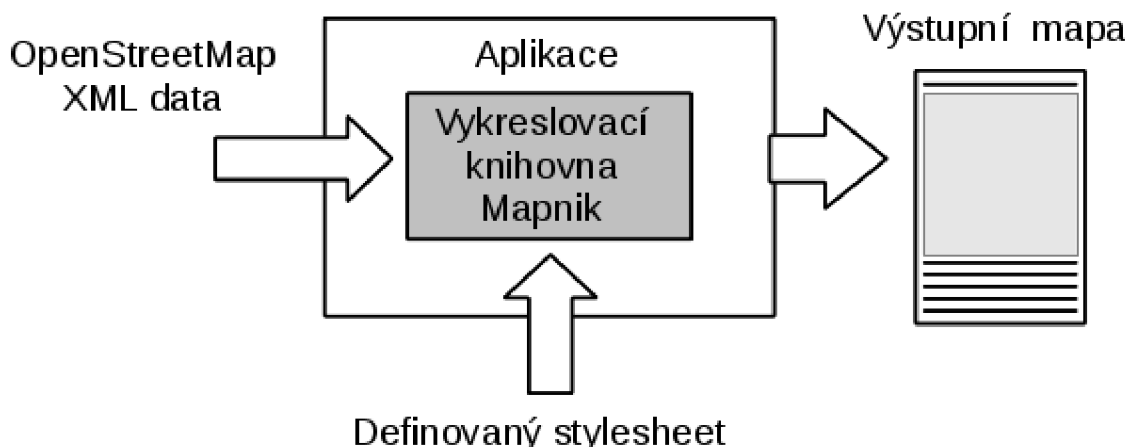
Koncept aplikace je znázorněn na obrázku [4.1](#).

### 4.2 Výběr vykreslovacího nástroje

Smyslem této práce není vytvořit nový vykreslovací nástroj. Je tedy důležité vybrat některou vhodnou knihovnu z již existujících a volně dostupných.

Jako první se nabízí nástroj Osmarender [3.3.4](#). Je vytvořen pouze pro vykreslování OpenStreetMap dat, což na jednu stranu je základní požadavek, ale stejně tak by jsme tím přišli o možnost přidávat další data z jiných zdrojů, což je potřeba například u vykreslování vrstevnic. Navíc jeho vývoj byl zastaven, takže nereflektuje změny v samotné OpenStreetMap databázi.

Další varianta je framework Cartagen [3.3.6](#), který bohužel nelze použít uvnitř aplikace jako vykreslovací knihovna.



Obrázek 4.1: Koncept aplikace.

Daleko vhodnější je použití knihovny Mapnik [3.3.1](#). Podporuje různorodé vstupní formáty dat a je používán i samotnou webovou aplikací OpenStreetMap, tudíž je možné generovat mapy se vzhledem odpovídajícím základnímu stylu OpenStreetMap. Mapnik pracuje s relativně složitým a komplikovaným souborem pro definici stylů (více v sekci [4.4](#)). Aplikace musí nabídnout jednoduchou („user friendly“) práci s tímto stylesheetem.

### 4.3 Výběr programovacího jazyka a vývojového prostředí

Zvolením knihovny Mapnik se výběr programovacího jazyka zúžil na C++ a Python. Zdrojový kód Mapniku je napsaný v jazyce C++, proto bude lepší zvolit tento jazyk. Implementaci uživatelského rozhraní nám usnadní použití multiplatformní knihovny Qt, která je volně dostupná pod licencí GPL.

#### 4.3.1 Qt framework

Qt [\[1\]](#) je především knihovna jazyka C++ určená k vytváření *GUI aplikací*. Vytvořené grafické aplikace jsou nezávislé na operačním systému. Volně dostupné je také vývojové prostředí *Qt Creator IDE* obsahující editor kódu a *Qt Designer* určený k tvorbě grafického rozhraní.

Knihovna nabízí velkou škálu knihovnických funkcí a tříd, například pro zpracování XML, pro síťovou komunikaci přes HTTP protokol, podpora pro více vláknové aplikace, algoritmy pro práci s bitmapami a další. Knihovna je velice dobře zdokumentovaná včetně mnoha příkladů užití.

Specialitou Qt je rozšíření standardu C++ o *signály* a *sloty*. Fungují na principu zasílání zpráv mezi objekty. Slot je speciální metoda, která se spustí při obdržení nějaké zprávy – signálu. Uplatňují se především u grafických prvků při jejich interakci s uživateli.

## 4.4 Mapnik XML stylesheet

Před samotnou implementací je potřeba znát strukturu souboru s definicemi stylů, který Mapnik očekává na svém vstupu. Soubor obsahuje definice vzhledu jednotlivých částí mapy a uvádí zdroje geografických dat pro jednotlivé vrstvy. Aby nemusel uživatel editovat tento soubor přímo, tak aplikace nabízí rozhraní, kde je možné pomocí formulářů zadávat a měnit hodnoty.

Struktura souboru „Mapnik XML stylesheet“ je následující:

- XML hlavička zavádějící UTF-8 kódování znaků v souboru
- **Map** – kořenový element s volitelnými atributy: **background-color** – nastavuje barvu pozadí, **background-image** – nastavuje obrázek na pozadí, **font-directory** – adresa obsahující vlastní písma, **srs** – výchozí referenční souřadnicový systém (4.4.3), **maximum-extent** – maximální rozsah mapy v zadané projekci
  - **Style** – element k definování jednotlivých stylů s atributem **name** k identifikaci stylu (více v sekci 4.4.1)
    - seznam elementů **Rule**
  - **Layer** – element definující jednotlivé vrstvy mapy s atributy:
    - seznam elementů **StyleName** odkazující se na dříve definovaný styl
    - element **Datasource** určující zdroj dat vrstvy (více v sekci 4.4.2)
  - **FontSet** – nastavení používaného písma

### 4.4.1 Definice stylů v XML souboru

Element **Style** slouží k přesné definici vzhledu jednotlivých prvků mapy. Jediným atributem je atribut **name** udávající název stylu a díky kterému se lze později na styl odkazovat u jednotlivých vrstev mapy. Každý element **Style** může obsahovat libovolné množství pravidel – elementy **Rule**. Jednotlivé elementy **Rule** vytváří pravidla. Pravidlo obsahuje popis vzhledu určitého typu prvku v mapě. Definice vzhledu se provádí výběrem a nastavením atributů některých z následujících podelementů<sup>1</sup>:

- **Filter** – pomocí relačních operátorů a operátorů *and*, *or* a *not* lze konstruovat filtry k specifitějšímu výběru objektů, na které se bude dané pravidlo vztahovat
- **ElseFilter** – pravidlo, kde je tento element uveden, je vyhodnoceno v případě, že žádné jiné předchozí pravidlo nevyhovuje aktuálnímu objektu
- **MinScaleDenominator** – omezuje minimální měřítko mapy pro použití pravidla
- **MaxScaleDenominator** – omezuje maximální měřítko mapy pro použití pravidla
- **PointSymbolizer** – specifikuje vykreslení grafického symbolu na konkrétní bod
- **LinePatternSymbolizer** – určuje symbol (soubor s obrázkem), který se vykreslí jako opakující se vzor na vybranou linii

<sup>1</sup>Detailed popis všech atributů u jednotlivých podelementů lze najít na webu: <https://github.com/mapnik/mapnik/wiki/XMLConfigReference>

- **PolygonPatternSymbolizer** – určuje bitmapový obrázek, který se použije jako textura pro vybranou plochu
- **TextSymbolizer** – nastavuje formát textových popisků v mapě
- **ShieldSymbolizer** – nastavuje formát textových popisků včetně obrázku na pozadí textu
- **LineStyleSymbolizer** – definuje vzhled linie (barva, tloušťka, průhlednost atd.)
- **PolygonSymbolizer** – definuje výplň vybraných polygonů
- **BuildingSymbolizer** – lze použít k vytváření pseudo 3D efektu polygonů (např. budov)
- **MarkersSymbolizer** – podobně jako **LinePatternSymbolizer** nastavuje symbol přiřazený k určité linii s více možnostmi nastavení
- **DebugSymbolizer** – speciální symbol určený k detekci kolizí

Elementy obsahující v názvu slovo „Symbolizer“ (kromě posledního uvedeného) definují vzhled prvků pomocí údajů uvedených jako atributy elementu. Pokud například styl obsahuje pravidlo:

```
<Rule>
  <MaxScaleDenominator>50000</MaxScaleDenominator>
  <LineStyleSymbolizer stroke="black" stroke-opacity="0.8" stroke-width="7.5"/>
</Rule>
```

znamená to, že všechny linie dané vrstvy se zobrazí s černou barvou, s 80% průhledností a tloušťkou čáry 7,5 pixelů. Zároveň tohle pravidlo platí pouze pro mapy s maximálním měřítkem 1:50 000.

#### 4.4.2 Definice vrstev v XML souboru

Každou vrstvu definuje element **Layer**. Mezi nastavitelné atributy elementu patří: **name** – název vrstvy, **srs** – určuje referenční souřadnicový systém pro danou vrstvu (4.4.3). Parametr **status** – „on“ nebo „off“ indikuje, jestli bude vrstva vykreslena, **abstract** a **title** obsahují dodatečné textové informace. Další je atribut **estimate\_extent** s hodnotami „true“ nebo „false“. Pokud je nastaven na „false“, je potřeba ručně nastavit rozsah mapy pomocí atributu **extent**, v opačném případě se rozsah mapy odhadne automaticky. Hodnoty **minzoom** a **maxzoom** určují, při jakém nejmenším resp. největším přiblížení se vrstva vykreslí.

Následuje seznam podelementů **StyleName** s odkazem na některý dříve definovaný styl. Při použití více stylů se aplikují podle pořadí v jakém jsou uvedeny. Při nesprávném použití více stylů může docházet ke kolizím a nedefinovanému chování.

U každé vrstvy mapy lze nastavit, z jakého zdroje budou čerpána mapová data. Pomocí atributu **type** určíme zdroj, který může nabývat hodnot „osm“, „postgis“, „shape“, „raster“, „gdal“ a „ogr“.

## OSM

Typ pro OpenStreetMap data. Můžeme použít soubor s daty uložený lokálně, nebo lze zadat URL poskytující OSM XML data, ze které se stáhnou.

- **file** (string) – lokální soubor s daty ve formátu OSM XML
- **url** (string) – URL adresa zdroje
- **bbox** (string) – ohraničující box k načtení dat z URL – řetězec obsahuje souřadnice ve formátu: **minlat**, **minlon**, **maxlat**, **maxlon**
- **parser** (string) – XML parser, musí obsahovat hodnotu „libxml2“, protože libxml2 je v současnosti jediný podporovaný XML parser

Atribut **file** nelze použít zároveň s atributy **url** a **bbox**.

## PostGIS

Pokud chceme jako zdroj použít postrelační databázi *PostgreSQL*<sup>2</sup> s geografickou nadstavbou *PostGIS*<sup>3</sup>, je potřeba OSM XML data importovat do databáze pomocí utility *osm2pgsql*<sup>4</sup>. K datům lze poté přistupovat pomocí následujících paramterů vrstvy:

- **host** (string) – síťové jméno hosta s databázovým serverem (např. „localhost“)
- **port** (integer) – označení portu (výchozí hodnota je „5432“)
- **dbname** (string) – jméno databáze (typicky „gis“)
- **user** (string) – uživatelské jméno použité k připojení
- **password** (string) – uživatelské heslo
- **table** (string) – název tabulky obsahující data, lze použít i SQL dotazy pro specifitější výběr dat

## ShapeFile

Zdroj dat může být nastaven na ESRI shapefile, parametry jsou: **file** – soubor ve formátu ESRI shapefile (.shp) a **encoding** – kódování souboru, výchozí hodnota je UTF-8.

## Raster

Zdrojem dat je obrázek ve formátu TIFF. Jediný parametr je **file** k lokalizaci souboru.

---

<sup>2</sup>Webové stránky: <http://www.postgresql.org/>

<sup>3</sup>Webové stránky: <http://postgis.net/>

<sup>4</sup>Webové stránky: <http://wiki.openstreetmap.org/wiki/Osm2pgsql>

## GDAL a OGR

Podpora knihovny *Geospatial Data Abstraction Library*<sup>5</sup> (GDAL) a její rozšiřující součástí *OGR Simple Feature Library* umožňuje zpracování mnoha geografických rastrových i vektorových formátů, jako například KML, gif, bmp, PostGIS, Oracle Spatial, ESRI Shapefile a další<sup>6</sup>. Parametry u obou typů jsou: `file` – soubor ve formátu podporovaném knihovnou GDAL/OGR, `base` – obsahuje cestu, kde jsou umístěny soubory. U typu OGR jsou pak ještě navíc parametry `layer` – vybere určitou vrstvu podle jména a `layer_by_index` – vybere určitou vrstvu podle jejího indexu.

### 4.4.3 Referenční souřadnicový systém

Mapnik používá k definici souřadnicového systému zápis ve formě *Proj.4 řetězce*<sup>7</sup>, který obsahuje parametry projekce.

Pro OSM XML data je nutné použít referenční souřadnicový systém *EPSG:4326* [2], jehož Proj.4 řetězec vypadá následovně:

```
srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
```

Při použití PostGIS databáze obsahující OpenStreetMap data je nezbytné použít referenční systém *SR-ORG:7483* (používá se i označení *EPSG:900913*, nebo také *Google projection* [2]):

```
srs="+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0  
+y_0=0 +k=1.0 +units=m +nadgrids=@null +wktext +no_defs"
```

---

<sup>5</sup>Webové stránky: <http://www.gdal.org/>

<sup>6</sup>Kompletní seznam podporovaných formátů je zde: [http://www.gdal.org/formats\\_list.html](http://www.gdal.org/formats_list.html)

<sup>7</sup>Knihovna provádějící konverze mezi různými kartografickými projekcemi. Webové stránky: <http://trac.osgeo.org/proj/>

# Kapitola 5

## Implementace

Implementace zabírá nemalou část práce. V této kapitole jsou probrány implementační detaily projektu. Kapitola je členěna do sekcí, které se postupně věnují jednotlivým třídám vytvořených v aplikaci.

### 5.1 Použité nástroje a třídy

Aplikace je napsaná v jazyce C++ s využitím multiplatformní knihovny Qt verze 4.8 [1]. Jako šablona je zvolena *Qt Gui Application* používaná pro standardní desktopové aplikace. Implementace dodržuje základní principy objektově orientovaného programování.

V aplikaci jsou implementovány následující třídy:

- `MainWindow` – hlavní okno aplikace
- `Render` – třída zajišťující vykreslování pomocí knihovny Mapnik
- `StyleSheet` – vnitřní reprezentace souboru *Mapnik XML stylesheet*, se kterou aplikace pracuje jednodušeji, než se samotným XML dokumentem
- `StyleXmlParser` – slouží k převodu XML souboru na objekt třídy `StyleSheet` a zpět
- pomocné dialogové třídy s formuláři – `EditStyleDialog`, `EditStyleMarkersDialog`, `EditStyleShieldDialog`, `EditStyleTextDialog`, `EditLayerDialog`, `EditStylesheetDialog`, `EditGridDialog`, `EditMapSizeDialog`, `ExportMapDialog` a `OptionsDialog`

### 5.2 Třída MainWindow

Třída představuje hlavní okno aplikace a zajišťuje interakci s uživatelem. V konstruktoru se vytvoří kompletní grafické rozhraní aplikace – hlavní menu, nástrojová lišta, stavový řádek, hlavní okno pro zobrazení mapy a postranní panel pro editaci stylu mapy. Vytvoří se objekty tříd `StyleSheet` a `StyleXmlParser`. Načte se nastavení aplikace a výchozí hodnoty proměnných, které si aplikace uchovává pomocí Qt třídy `QSettings`. Grafické rozvržení hlavního okna je vidět na obrázku 5.1.

### 5.2.1 Načtení lokální mapy

OSM XML data lze stáhnout buď přímo ze stránek aplikace OpenStreetMap (záložka „Export“), nebo pomocí nějaké aplikace pracující s OpenStreetMap daty. Při otevření tohoto souboru v aplikaci se vytvoří lokální kopie (s výchozím názvem `tmp_map.osm`) a aplikace dále pracuje jen s touto kopií.

### 5.2.2 Stahování dat ze serveru OpenStreetMap

Druhá možnost, jak získat OSM XML data, je zaslání přímého požadavku na OpenStreetMap server pomocí rozhraní v hlavním okně aplikace. Po kliknutí na tlačítko „Download data“ se ze vstupních boxů umístěných v nástrojové liště, kde uživatel může specifikovat ohraničení mapy v zeměpisných souřadnicích (tzv. *bbox*, či *bounding box*), získají proměnné `minlat`, `minlon`, `maxlat` a `maxlon`. První dvě hodnoty definují levý dolní okraj a další dvě pravý horní okraj mapy. Použijí se při zaslání požadavku na server. To se provede pomocí Qt třídy `QHttp` implementující metodu `get`:

```
GET www.openstreetmap.org/api/0.6/map?bbox=minlon,minlat,maxlon,maxlat
```

Informace o přenosu dat se zobrazují na stavový řádek. Pokud je zadané ohraničení příliš velké (počet elementů v požadované oblasti přesahuje 50 000), uživatel je informován o problému s doporučením zadat menší oblast. Při úspěšném dokončení přenosu jsou data uložena do souboru `tmp_map.osm` a pokud je i nastaven stylesheet, je možné vykreslit mapu.

### 5.2.3 Práce se styly

Pro vykreslení mapy je potřeba otevřít nebo vytvořit nový stylesheet. V případě otevření již existujícího si aplikace vytvoří kopii s výchozím názvem `tmp_osm.xml` a dále pracuje s touto kopií. Při vytváření nového stylesheetu se zadává název, který má význam při následném ukládání do souboru. Nástroje pro editaci stylesheetu jsou k dispozici v postranním panelu. Ten obsahuje dvě sekce – seznam stylů a seznam vrstev. Jednotlivé styly a vrstvy lze editovat (tlačítko „Edit“, nebo dvojitým kliknutím na příslušnou položku), přidávat nové, odebírat nebo měnit jejich pořadí (na výsledný vzhled mapy má vliv pouze pořadí jednotlivých vrstev, pořadí stylů vliv nemá). Editaci stylu zajišťuje dialogová třída `EditStyleDialog` 5.6.1 a editaci vrstvy dialog `EditLayerDialog` 5.6.2. Kromě stylů a vrstev lze editovat i samotný stylesheet pomocí třídy `EditStylesheetDialog` 5.6.3. Stylesheet je možné uložit na disk ve formátu Mapnik XML.

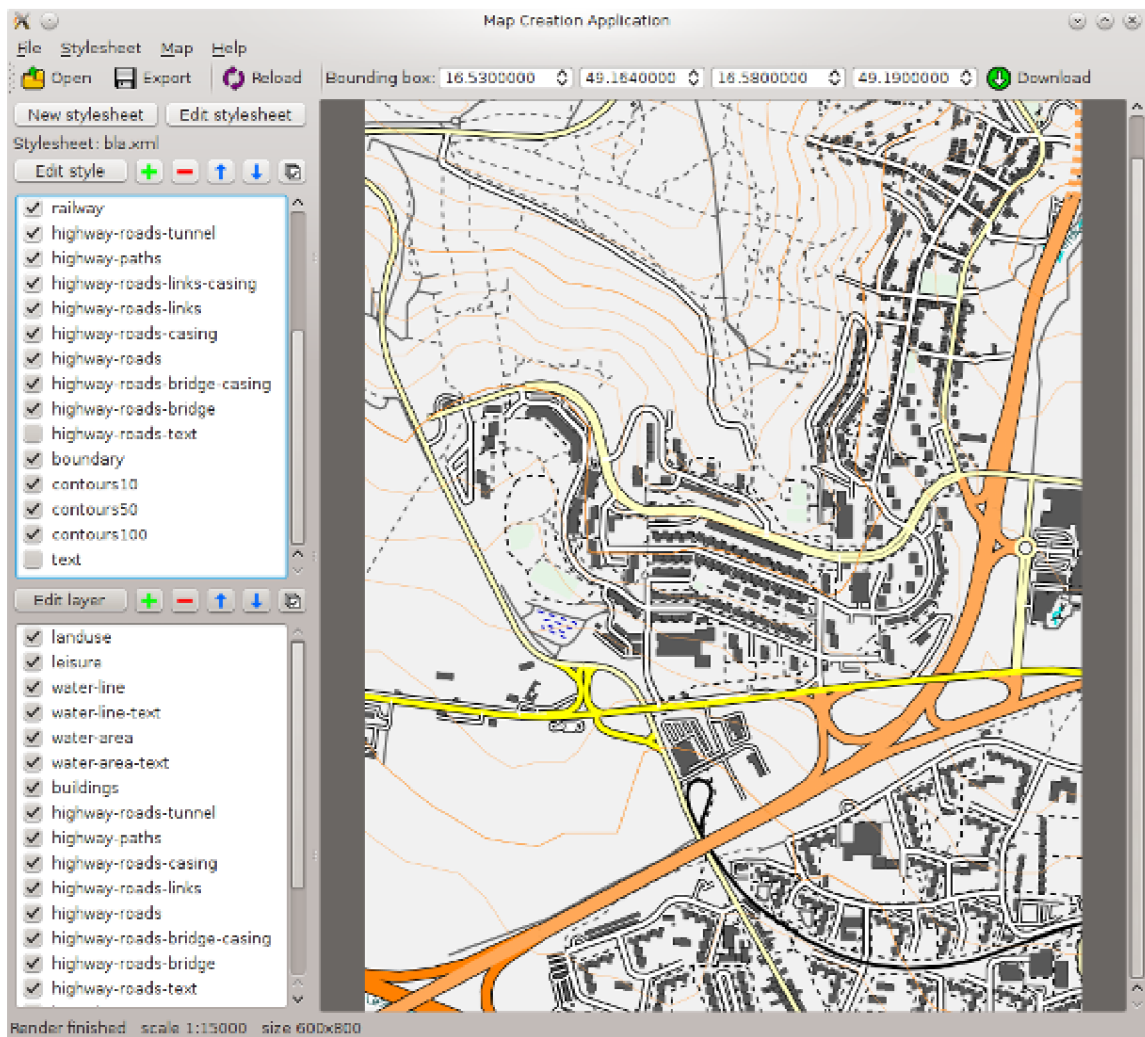
### 5.2.4 Nastavení souřadnicové mřížky

Uživatel může nastavit zobrazení mřížky poledníků a rovnoběžek. Parametry mřížky se získají z formuláře – viz sekce o dialogové třídě `EditGridDialog` 5.6.4.

### 5.2.5 Nastavení velikosti mapy

Nastavení velikosti vykreslené bitmapy se provádí v dialogu `EditMapSizeDialog` 5.6.5.





Obrázek 5.1: Hlavní okno aplikace s ukázkou vykreslené mapy.

### 5.2.6 Vykreslení mapy

Mapa se vykresluje pouze na požadavek uživatele a ne po každé provedené změně. Proces vykreslování běží v samostatném vlákně nezávisle na hlavním programu. Na stavový řádek se vypisují informace o průběh vykreslování. Jakmile je mapa vykreslena, spustí se obslužná funkce `renderFinished()` a ta zobrazí získanou bitmapu do hlavního okna aplikace. Při neúspěšném vykreslení provádí obsluhu funkce `renderFailed()`, která uživatele informuje o vzniklé chybě. Samotnému procesu vykreslování se věnuje sekce 5.3.

### 5.2.7 Export mapy

Jakmile je mapa úspěšně vykreslena, zobrazí se na hlavní plochu aplikace. Nyní je možné mapu uložit. Detailnímu popisu exportování mapy se věnuje kapitola `ExportMapDialog` 5.6.7.

## 5.3 Třída Render

Třída `Render` je zděděná z Qt třídy `QThread`. Vykreslování mapy se tak provádí v odlišném vlákně, než je hlavní okno aplikace. Vlákno se spouští po předání vykreslovacích parametrů – požadovaná velikost bitmapy, ohraničující box (4 zeměpisné souřadnice), referenční souřadnicový systém, parametry sítě souřadnic, výchozí názvy souborů.

Tato třída využívá externí knihovnu `Mapnik`. Práci s `Mapnikem` přibližuje následující pseudokód:

```
// mapa s parametry šířka, výška a referenční souřadnicový systém
Map map(width, height, srs);
...
// načtení souboru Mapnik XML stylesheet
load_map(map, "style.xml");
// nastavení ohraničující oblasti
map.zoom_to_box(minlon, minlat, maxlon, maxlat);
// vykreslení mapy do výstupní bitmapy
render(map, output_image);
```

Po úspěšně dokončení je hlavní okno informováno pomocí signálu `renFinished()` a je mu předána bitmapa obsahující vykreslenou mapu se zvolenými parametry.

### 5.3.1 Generování souřadnicové mřížky

Pokud uživatel aktivoval generování mřížky a zadal příslušné parametry, tak se ještě před vykreslením mapy vygeneruje mřížka (funkce `generateGrid()`). Vytvoří se nový soubor `grid.osm`, do kterého jsou uloženy linie představující síť souřadnic ve formátu OSM XML. Oblast je vymezená hodnotami proměnných `minlat`, `minlon`, `maxlat` a `maxlon`. Mřížka se tedy generuje pouze pro dané ohraničení mapy a ne zbytečně mimo oblast mapy.

Hodnoty ohraničujících proměnných mohou být zadané i s přesností na desetiny vteřiny, proto je potřeba tyto hodnoty vhodným způsobem zaokrouhlit, aby se mřížka generovala pouze na celé násobky stupňů, minut či vteřin (záleží na zadané šířce mřížky). Generátor provede zaokrouhlení na celé minuty odříznutím vteřinové části. Algoritmus bere v potaz i možné záporné zeměpisné souřadnice.

Rozteč mřížky uživatel zadává ve formě tří hodnot – stupně, minuty a vteřiny. `Mapnik` i `OpenStreetMap` používají pro zeměpisné souřadnice desetinná čísla, je tedy potřeba hodnoty převést [10]:

```
double step = degrees + minutes/60.0 + seconds/3600.0;
```

Hodnota `step` se poté použije jako rozstup mezi vygenerovanými horizontálními a vertikálními liniemi.

### 5.3.2 Korekce měřítka

Měřítko vygenerované mapy se odvíjí od zadané velikosti a ohraničujícího boxu, tudíž ve většině případů je výsledné měřítko libovolné číslo. Standardně se ovšem měřítka udávají číslem s první číslicí libovolnou (případně pár prvních číslic libovolných), zbytek čísla však tvoří nuly. `Mapnik` nenabízí možnost vykreslit mapu pro požadované měřítko, ale je možné mapu přiblížit zadáním *faktoru*. Faktor je desetinné číslo udávající přiblížení, získáme ho

poměrem požadovaného měřítka ku původnímu měřítku. Nové měřítko získáme nahrazením všech čísel nulami kromě prvního. Pokud je druhá číslice  $\geq 5$ , nahradíme i druhou číslici číslem 5:

```
double old = map_scale;
int new = ponech_pouze_prvni_cislici( map_scale );
if (old.druha_cislice >= 5)
    new.druha_cislice = 5;
map.zoom( new / old );
```

## 5.4 Třída StyleSheet

Třída `StyleSheet` definuje vlastní zjednodušenou reprezentaci souboru *Mapnik XML stylesheet* 4.4. Tento XML soubor má složitou strukturu a velké množství různě zanořených elementů, takže práce přímo s ním by byla velice neefektivní. Proto je potřeba soubor nejdříve převést do vlastní datové reprezentace (převodem se zabývá následující sekce 5.5). Na základě uspořádání elementů v souboru jsou deklarovány nové datové struktury a operace nad nimi. Základem jsou dva uspořádané seznamy:

```
QList<TStyleElement> styleList;
QList<TLayerElement> layerList;
```

kde `TStyleElement` je datový typ reprezentující jednotlivé styly a `TLayerElement` je datový typ pro jednotlivé vrstvy. Celá deklarace je uvedena v příloze A.

S proměnnými `styleList` a `layerList` lze následně pracovat pomocí metod pro vkládání, modifikování, odebírání prvků a nebo zjištění přítomnosti prvku. Jelikož se jedná o uspořádané seznamy, jsou na místě i metody pro přesun jednotlivých položek.

## 5.5 Třída StyleXmlParser

Pomocí Qt třídy `QDomDocument` lze zpracovávat data ve formátu XML. Stylesheet lze otevřít již existující, nebo vytvořit nový. Soubor se styly se načte voláním metody `readFile()`, kde se testuje existence souboru `tmp_osm.xml`. Předpokládá se, že před voláním této metody byl soubor `tmp_osm.xml` již vytvořen a to buď zcela nový (metoda `createFile()`) a nebo jako kopie uživatelem zadaného souboru. Program pracuje pouze s kopií souboru proto, aby nedošlo k destrukci původního souboru. Po otevření souboru se uloží kořenový element a následuje zpracování jednotlivých podelementů.

Metoda `parseStyleSheetElements()` převede celý dokument do reprezentace formou třídy `StyleSheet`. Od kořenového elementu se prochází postupně po jeho jednotlivých potomcích. Podle typu elementu se ukládá buď nový font, styl nebo vrstva. Zároveň se přidávají položky do postranního okna aplikace, kde jsou vidět jednotlivé styly a vrstvy.

Naopak metoda `updateStyleSheet()` převádí z vnitřní reprezentace dat formou třídy `StyleSheet` zpět na XML data uložená v třídě `QDomDocument`.

K uložení XML souboru slouží metoda `writeFile()`. Při jejím spuštění se převedou data do XML pomocí zmiňované metody `updateStyleSheet()` a získaný `QDomDocument` se jednoduše uloží jako XML soubor se zadaným jménem.

## 5.6 Ostatní dialogové třídy

Tyto dialogové třídy slouží k získávání dat od uživatele pomocí formulářů. Uživatel zde může zadávat a měnit celou řadu položek a tato data je často potřeba předat zpět třídě `MainWindow`, kde se dále zpracují.

### 5.6.1 Třída `EditStyleDialog`

Dialog sloužící k editaci informací o vybraném stylu. Obsahuje seznam pravidel, umožňuje přidávání, mazání a editování jednotlivých pravidel. Ve formuláři pak lze přidávat a upravovat informace týkající se právě vybraného pravidla. Popis všech parametrů se nachází v sekci 4.4.1. Všechny nastavitelné parametry stylu se nevejdou do jednoho dialogu, proto jsou zde následující tři pomocné dialogy:

- Dialogová třída `EditStyleTextDialog` – uživatel zadává a mění informace o elementu `TextSymbolizer` vztahené k právě vybranému pravidlu.
- Dialogová třída `EditStyleShieldDialog` – uživatel zadává a mění informace o elementu `ShieldSymbolizer`
- Dialogová třída `EditStyleMarkersDialog` – uživatel zadává a mění informace o elementu `MarkersSymbolizer`

Vzhled dialogu pro úpravu stylů je ukázán na obrázku 5.2.

### 5.6.2 Třída `EditLayerDialog`

Dialog sloužící k úpravě informací o vybrané vrstvě. U každé vrstvy je možné nastavit jednotlivé parametry detailně popsané v části 4.4.2. Výběr referenčního souřadnicového systému usnadňuje nabídka s nejčastěji používanými `srs` řetězci (viz 4.4.3).

### 5.6.3 Třída `EditStylesheetDialog`

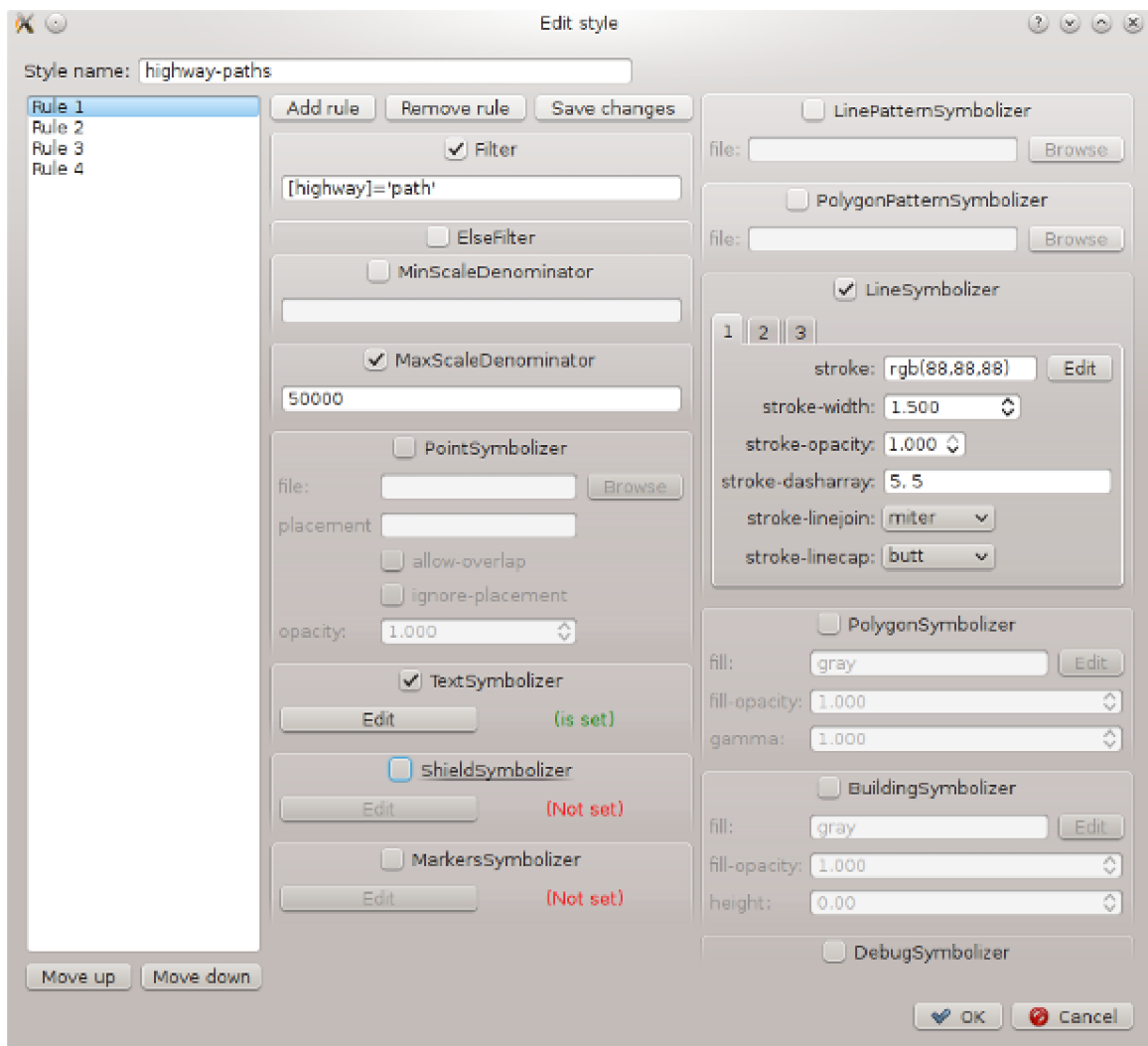
Dialog, kde uživatel může měnit informace týkající se samotného stylesheetu a mapy obecně. V tomto dialogu se nastavuje barva pozadí mapy, referenční souřadnicový systém (opět s nabídkou nejčastěji používaných), obrázek na pozadí mapy, složka s fonty, souřadnice udávající maximální rozsah mapy a použité druhy písma.

### 5.6.4 Třída `EditGridDialog`

V tomto dialogu se nastavuje zobrazení souřadnicové mřížky. Při výchozím nastavení je vypnutá, spuštěním dialogu se aktivuje. V dialogu je možné nastavit barvu mřížky, tloušťku a průhlednost čáry a vzdálenost mezi jednotlivými linkami ve stupních (výchozí hodnota je 1 minuta). Dialog slouží pouze k získání vstupních informací o vlastnostech souřadnicové mřížky, samotná mřížka se generuje až při vykreslování (viz sekce 5.3).

### 5.6.5 Třída `EditMapSizeDialog`

Zde uživatel zadává vstupní informace pro `Mapnik` týkající se velikosti vykreslené bitmapy. Proměnné jsou šířka a výška v pixelech a proměnná `aspect_fix_mode`, která definuje, jakým způsobem se bude řešit rozdílný poměr velikosti bitmapy a ohraničujícího boxu. Na výběr je z několika možností, např. ořez bitmapy, rozšíření bitmapy, ořez boxu atd.



Obrázek 5.2: Dialog pro editaci stylu. Tento konkrétní styl upravuje vzhled cest pro pěší.

### 5.6.6 Třída OptionsDialog

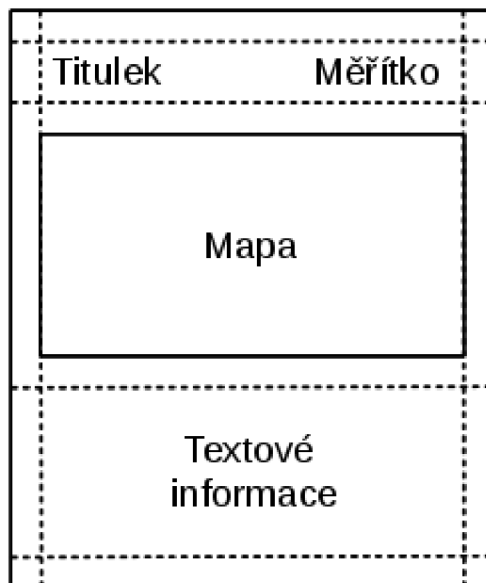
Zde se mění některá nastavení aplikace – výchozí názvy souboru se stylem a souboru s mapovými daty, výchozí URL adresa, ze které lze stáhnout OpenStreetMap data a další.

### 5.6.7 Třída ExportMapDialog

V tomto dialogu se blíže specifikuje, co všechno bude obsahovat obrázek na výstupu. Dialog obsahuje náhled výsledné bitmapy. Provedené změny se ihned reflektují do náhledu. Lze měnit text titulku mapy, měnit velikost textu nadpisu a měřítka, měnit velikost ohraničujícího plátna (výchozí poměr stran je  $1 : \sqrt{2}$ <sup>1</sup>, poměr stran lze prohodit), měnit velikost okraje. Stejná hodnota jako pro okraj je použita i pro rozestupy mezi jednotlivými prvky. Dále lze specifikovat výstupní formát obrázku (.png, .jpg, .bmp). Nastavit lze i velikost

<sup>1</sup>Odpovídá poměru stran běžného papíru.

boxu obsahujícího doplňující textové informace. Rozložení všech prvků je znázorněno na obrázku 5.3.



Obrázek 5.3: Schéma rozvržení elementů ve výsledném rastru.

Velikost exportovaného obrázku se ovlivní buď změnou velikosti plátna, nebo změnou měřítka. K oběma hodnotám je při každé změně dopočítána hodnota rozlišení *dots per inch* (DPI). Vychází se z toho, že Mapnik používá hodnotu DPI= 90.7 [8].

Obrázek 5.4 ukazuje, jak vypadá dialog pro export mapy.

### Načítání doplňujících textových informací

Dialog pro export mapy umožňuje přidat i dodatečné textové informace z volně dostupných informačních zdrojů. Nabízí se dvě možnosti – buď jsou v mapě k dispozici přímo odkazy na články z Wikipedie<sup>2</sup>, nebo se z mapy extrahují geografické názvy a k nim se aplikace pokusí dohledat informace.

- První varianta počítá s tím, že OSM XML data obsahují URL adresy v některém z následujících formátů:

```
<tag k="wikipedia" v="http://XX.wikipedia.org/wiki/NÁZEV_ČLÁNKU"/>  
<tag k="wikipedia" v="XX:NÁZEV_ČLÁNKU"/>  
<tag k="wikipedia:XX" v="NÁZEV_ČLÁNKU"/>
```

V druhém a třetím případě jsou uvedeny jen názvy článků, z nichž lze jednoduše pomocí běžných operací s řetězci vytvořit řetězce s URL adresou. Znaky XX nahrazují doménu třetího řádu určující, v jakém jazyce jsou informace.

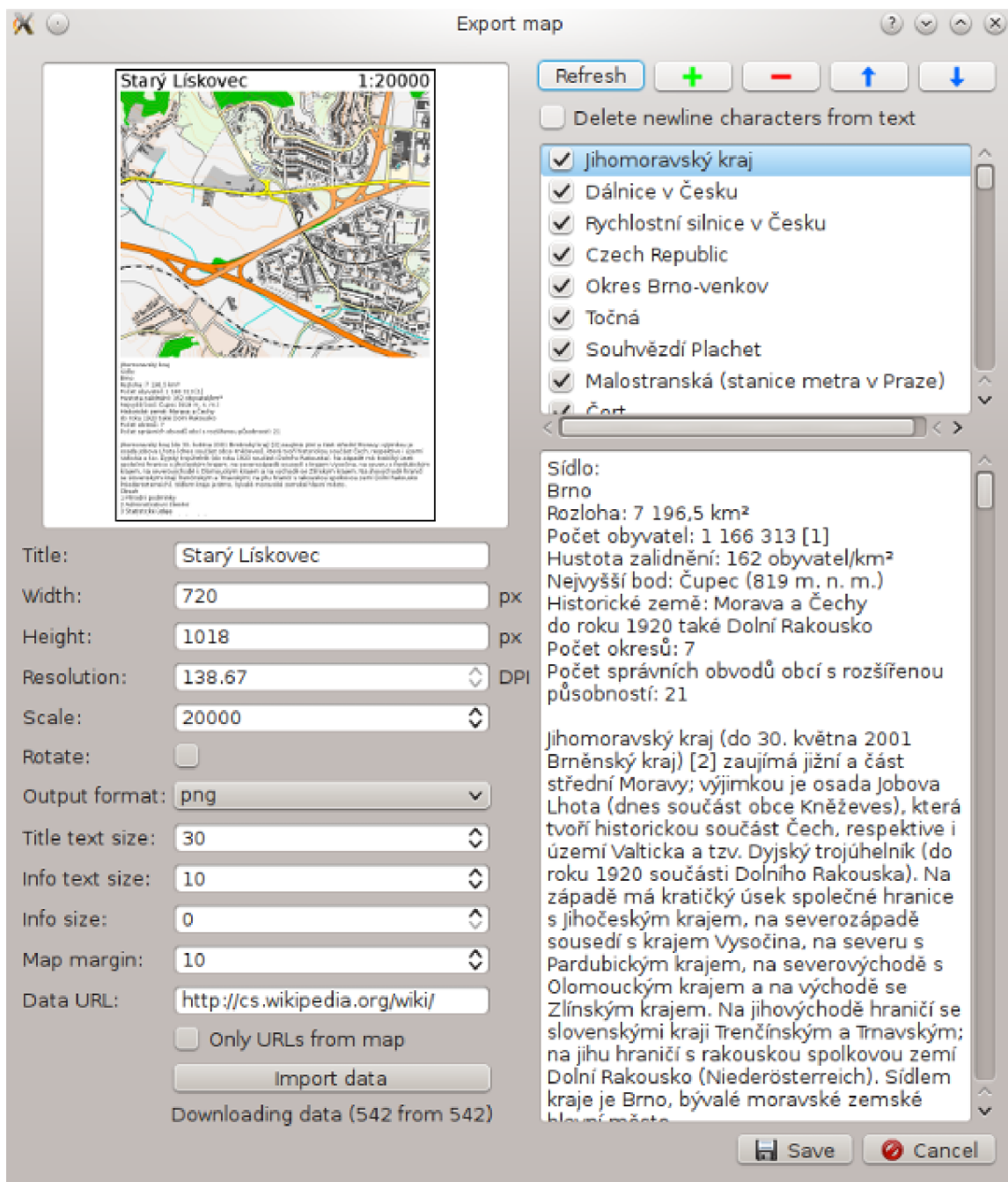
- U druhé varianty jsou z OSM XML dat získány všechny místní názvy. Předpokladem je jejich uložení v elementech ve formátu:

<sup>2</sup>Webové stránky: <http://www.wikipedia.org/>

```
<tag k="name" v="NÁZEV"/>
```

Regulárním výrazem získáme řetězec **NÁZEV** a jeho konkatencí s uživatelem zadanou adresou se vytvoří seznam potenciálních URL adres.

Uživatel může použít buď obě metody, nebo pouze první. V obou případech se posílá pro všechny URL adresy HTTP požadavek pomocí Qt třídy `QHttp`. Při pozitivní odpovědi (návrátový kód 200) se z přijaté HTML stránky ořízne hlavička, zápatí a další redundantní informace. Každý úspěšně stažený článek se převede do čistého textu bez HTML tagů a uloží se jako nová položka do seznamu s informacemi. Na pravé straně dialogu je seznam jednotlivých položek. Text se získanými informacemi náležící k aktuálně vybrané položce se zobrazí v editačním okně, kde jej lze libovolně editovat. Položky lze přidávat nové, mazat nebo měnit jejich pořadí. Kliknutím na tlačítko „Refresh“ se aktualizuje náhled mapky spolu s textovými informacemi. Do mapy se vloží text tak, jak jdou jednotlivé textové položky za sebou. Jednotlivé textové bloky jsou odděleny prázdným řádkem a volitelně lze v každém bloku odstranit znaky „nový řádek“.



Obrázek 5.4: Dialog s exportem mapy.



## Kapitola 6

# Testování

Výslednou aplikaci bylo potřeba otestovat a ověřit její funkčnost. Výstupem testů jsou tři mapy, každá je vytvořena s jiným stylem. Pokud není upřesněno jinak, tak testování proběhlo pod operačním systémem *Fedora release 18 (Spherical Cow)*, *KDE 4.10.2* (včetně screenshotů aplikace použitých v předchozích kapitolách).

### 6.1 Mapa s jednoduchým stylem

Výstupem prvního testu je mapa s jednoduchým stylem vhodná pro běžnou orientaci po městě. Součástí testu je i ukázka importování vrstevnic do mapy. Vytvořený stylesheet se hodí pro mapy s měřítkem řádově 1:10 000. Mapa obsahuje pouze málo syté barvy, takže je úspornější pro tisk.

Zdrojový OSM XML soubor lze získat přímo z webu <http://www.openstreetmap.org/> v záložce *Export* a výběrem požadované oblasti nebo pomocí externí aplikace – v tomto testu byl použit program JOSM<sup>1</sup>.

#### 6.1.1 Přidání vrstevnic

K vytvoření vrstevnic byla použita data projektu *Shuttle Radar Topography Mission*<sup>2</sup> (SRTM). Topografická SRTM data jsou volně k dispozici ze stránek NASA: [http://dds.cr.usgs.gov/srtm/version2\\_1/SRTM3/](http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/). Každý soubor zabírá oblast 1 stupeň čtvereční a má rozlišení až 3 úhlové vteřiny – to odpovídá přibližně vzdálenosti 90 metrů. Protože vykreslujeme mapu části Brna ležící přibližně na 49. severní rovnoběžce a 16. východním poledníku, potřebujeme stáhnout soubor:

```
http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/Eurasia/N49E016.hgt.zip
```

Soubor ve formátu `.hgt` obsahuje matici o velikosti  $1201 \times 1201$ , kde každá buňka nese číselnou informaci o nadmořské výšce. Mapnik neumí s tímto formátem pracovat, programem `gdal_contour`<sup>3</sup> jej však lze převést do vektorového formátu ESRI shapefile:

<sup>1</sup>Program umožňující editaci OpenStreetMap dat. Webové stránky: <http://josm.openstreetmap.de/>

<sup>2</sup>Jedná se o program NASA, který si kladl za cíl vytvořit komplexní a detailní výškovou mapu Země. Webové stránky: <http://www2.jpl.nasa.gov/srtm/>

<sup>3</sup>Program je součástí knihovny GDAL, která je potřebná pro instalaci Mapniku. Webové stránky: <http://www.gdal.org/>

```
gdal_contour -i 10 -snodata 32767 -a height N49E016.hgt N49E016c10.shp
gdal_contour -i 50 -snodata 32767 -a height N49E016.hgt N49E016c50.shp
gdal_contour -i 100 -snodata 32767 -a height N49E016.hgt N49E016c100.shp
```

Parametr `-i` definuje rozestupy vrstevnic v metrech, parametrem `-snodata` se odstraní nepřesnosti způsobené převodem a parametr `-a` přidá do souboru textové informace o nadmořské výšce vrstevnice. Jsou tedy vytvořené tři soubory, každý s jiným intervalem vrstevnic. Do stylesheetu je v aplikaci potřeba přidat pro každý soubor jednu vrstvu včetně stylu. Pokračuje se nastavením vzhledu linií a textu, ve vrstvě se vybere zdroj dat „shape“ s cestou k souboru. V Mapnik XML souboru vypadá definice stylu pro vrstevnice s intervalem 10 m následovně:

```
<Style name="contours10">
  <Rule>
    <LineStyle stroke="rgb(255,168,88)" stroke-width="0.5"/>
    <TextSymbolizer size="10" fontset-name="book-fonts">[height]
  </TextSymbolizer>
  </Rule>
</Style>
<Layer name="srtm10" srs="+proj=latlong +datum=WGS84">
  <StyleName>contours10</StyleName>
  <Datasource>
    <Parameter name="type">shape</Parameter>
    <Parameter name="file">srtm/N49E016c10.shp</Parameter>
  </Datasource>
</Layer>
```

Stylesheet je připravený a mapu je možné vykreslit. Při exportování byly doplněny navíc textové informace. Výsledná mapa je exportována do formátu `.jpg`, má měřítko 1:15 000, rozměry 1120 × 1583 pixelů, rozlišení 137 DPI a velikost 495.5 kB.

Výslednou mapu je možné prohlédnout v příloze [B.1](#) nebo na přiloženém CD pod názvem `Test_Map_01.jpg`.

## 6.2 Standardní OpenStreetMap vzhled

Tato mapa byla vytvořena na počítači s operačním systémem *Ubuntu 10.04 LTS - the Lucid Lynx* a s použitými nástroji: databázový systém PostgreSQL – verze 8.4.17 a jeho geografická nadstavba PostGIS – verze 1.4.0 a program `osm2pgsql` – verze 0.83.0. Vykreslení mapy se standardním OpenStreetMap stylem vyžaduje mít vytvořenou a spuštěnou databázi a následné importování OSM XML dat. Pro účel testování byla vytvořena databáze s názvem „gis“ a uživatel „username“ s právem přístupu do databáze.

Standardní OpenStreetMap stylesheet je k dispozici v oficiálním OpenStreetMap repozitáři<sup>4</sup>. Po stáhnutí bylo potřeba spustit skript `get-coastlines.sh` k získání doplňujících geografických dat. Konfigurace stylesheetu se provedla skriptem `generate_xml.py`:

```
./generate_xml.py osm.xml myosm.xml --host localhost --port 5432 \
--dbname gis --user username
```

<sup>4</sup>Webové stránky: <https://github.com/openstreetmap/mapnik-stylesheets>

tímto se vytvořila kopie stylesheetu s názvem `myosm.xml`, ale přizpůsobená lokálnímu nastavení databáze. Následovalo importování požadovaných OpenStreetMap dat do databáze:

```
osm2pgsql -d gis -U username --slim map.osm
```

V aplikaci pak už bylo potřeba jen otevřít stylesheet `myosm.xml`, specifikovat ohraničující box (případně v předchozím kroku spustit `osm2pgsql` s parametrem `--bbox`) a spustit proces renderování. Před uložením byly přidány doplňující textové informace o zajímavých objektech na mapě.

Výsledná mapa je ve formátu `.png` má měřítko 1:27 000, rozměry 1290 × 1824 pixelů, rozlišení 164 DPI a velikost 2.417 MB a je k dispozici na příloženém CD pod názvem `Test_Map_02.png` nebo v příloze [B.2](#)

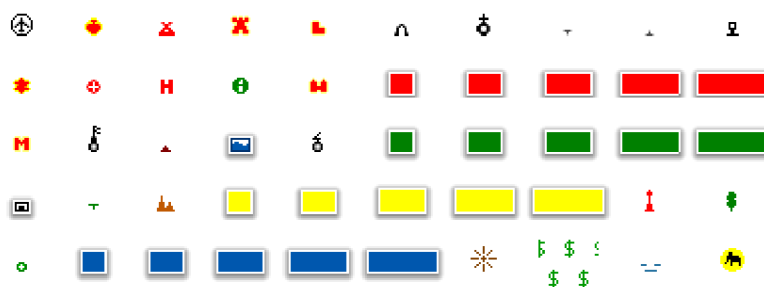
### 6.3 Mapa rozsáhlejší oblasti

Cílem toho testu bylo vytvoření mapy se vzhledem autoatlasu (s měřítkem řádově 1:100 000). Použití základní OpenStreetMap API je omezeno pouze na oblast s maximálním počtem prvků 50 000. Pro vytváření mapy větší oblasti je to značné omezení. Za tímto účelem byla tedy použita *OSM Extended API* (XAPI)<sup>5</sup>, konkrétně XAPI poskytovaná službou *Open MapQuest*<sup>6</sup>. V dialogu nastavení aplikace byla změněna URL adresa pro stahování dat na následující:

```
http://open.mapquestapi.com/xapi/api/0.6/*[bbox=16.35,49.08,16.62,49.2]
```

Získaná data zaujímají oblast na jihozápad od Brna v rozsahu přibližně 20 km čtverečních. Vykreslování této velké oblasti bylo poměrně časově náročné, každé spuštění renderování trvalo řádově několik sekund. Zde se osvědčila implementace vykreslování v samostatném vlákně.

K dosažení podobného vzhledu, jako autoatlas, byla vytvořena sada symbolů reprezentující vybrané bodové prvky zobrazené na obrázku [6.1](#).



Obrázek 6.1: Množina bodových symbolů.

<sup>5</sup>XAPI je modifikovaná verze standardní OSM API určená pouze pro čtení dat, takže lze přenos dat značně urychlit. Díky tomu lze stahovat mapy až do velikosti 10 stupňů čtverečních.

<sup>6</sup>Webové stránky: <http://open.mapquestapi.com/xapi/>

Nakonec byla do mapy přidána mřížka zeměpisných souřadnic s rozstupem 2 úhlové minuty. Exportovaná mapa ve formátu `.png` má měřítko 1:70 000, rozměry  $1948 \times 1377$  pixelů, rozlišení 172 DPI a velikost 2.272 MB.

Výslednou mapu je možné prohlédnout v příloze **B.3**, nebo na přiloženém CD pod názvem `Test_Map_03.png`.

# Kapitola 7

## Závěr

Hlavním cílem této bakalářské práce byla implementace aplikace, která umožňuje vytvářet a upravovat vzhled map z dat projektu OpenStreetMap. Součástí práce bylo rovněž nastudování teorie kartografické tvorby a průzkum již existujících aplikací. Aplikaci se podařilo vytvořit tak, aby splňovala všechny součásti zadání. Provedené testy proběhly úspěšně a pomohly odhalit některé nedostatky.

V porovnání s podobnými nástroji (TileMill 3.3.2, Osmarender 3.3.4) aplikace nabízí uživatelské rozhraní pro editaci stylů, takže není potřeba ručně upravovat rozsáhlé XML soubory. Výstup aplikace je podobný plakátu generovaného programem MapOSMatic 3.3.7, který je ale omezen jen na předem definované styly.

Přestože aplikace klade důraz na jednoduché rozhraní, pořád je pro efektivní používání aplikace nutné znát strukturu OpenStreetMap dat, především definované názvy a identifikátory prvků v mapě.

### 7.1 Možná rozšíření

Téma vykreslování map je velmi rozsáhlé a možností rozšíření je nespočetně. Jednoduchý návrh aplikace umožňuje snadné přidávání funkcionality.

Jako první možnost rozšíření se nabízí generování legendy, což je relativně důležitá součást mapy, ale aplikace ji nedokáže vytvořit. Co se týče přidávání vrstevnic, tak v současném stavu je možné přidávat topografická data pouze ručně jako novou vrstvu. Uživatel se musí postarat o převedení dat do formátu, který aplikace akceptuje. Tato činnost by šla zautomatizovat, kdyby program na základě souřadnic v mapě transparentně stáhl potřebná data a sám je konvertoval do potřebného formátu (podobně jak je tomu například u vytváření souřadnicové mřížky). Zajímavou možností rozšíření by byla možnost získávat mapová data větších oblastí. Standardně lze přes OpenStreetMap API stáhnout soubor pouze s omezeným počtem elementů. Jednou z možností, jak tento problém obejít, je stáhnout mapová data po částech a ty potom sloučit do jednoho souboru.

# Literatura

- [1] Qt Project. [online]. 2013 [citováno 14. dubna 2013].  
URL <http://qt-project.org/>
- [2] Butler, H.; Schmidt, C.; Springmeyer, D.; aj.: Spatial Reference. [online]. 2013 [citováno 5. května 2013].  
URL <http://spatialreference.org/>
- [3] Coast, S.: OpenStreetMap the Free Wiki World Map. [online]. 2013 [citováno 30. dubna 2013].  
URL <http://www.openstreetmap.org/>
- [4] Hrubý, M.: *Studijní opora k předmětu Geografické Informační Systémy*. FIT VUT v Brně, 2006.  
URL <http://perchta.fit.vutbr.cz/vyuka-gis/uploads/1/GIS-final2.pdf>
- [5] Jedlička, K.; Břehovský, M.; Šíma, J.: *Úvod do geografických informačních systémů*. Západočeská univerzita, Plzeň, 2003.
- [6] Kaplan, V.; Keprtová, K.; Konečný, M.; aj.: Multimediální učebnice Kartografie a Geoinformatiky. [online]. 2012 [citováno 5. ledna 2013].  
URL <http://oldgeogr.muni.cz/ucebnice/kartografie/obsah.php>
- [7] Monmonier, M.: *Proč mapy lžou*. Computer Press, 2000, ISBN 80-7226-238-6.
- [8] Pavlenko, A.: Mapnik C++/Python GIS Toolkit. [online]. 2011 [citováno 9. dubna 2013].  
URL <http://mapnik.org/>
- [9] Rapant, P.: *Geoinformatika a geoinformační technologie*. VŠB - Technická univerzita Ostrava, Hornicko-geologická fakulta, Institut geoinformatiky, 2006, ISBN 80-248-1264-9.
- [10] Weisstein, E.: Wolfram MathWorld: The Web's Most Extensive Mathematics Resource. [online]. 2013 [citováno 30. dubna 2013].  
URL <http://mathworld.wolfram.com/>
- [11] Zeiler, M.: *Modeling Our World: The ESRI Guide to Geodatabase Design*. ESRI Press, 1999, ISBN 18-791-0262-5.
- [12] Čapek, R.; Mikšovský, M.; Mucha, L.: *Geografická kartografie*. Státní pedagogické nakladatelství, 1992, ISBN 80-04-25153-6.

# Příloha A

## Ukázka kódu deklarace

```
// Reprezentace Mapnik XML souboru v třídě StyleSheet
class classRule {
public:
    classRule(); // konstruktor pro nastavení výchozích hodnot
    // následují jednotlivé položky pravidla
    QString filter;
    bool elseFilter;
    int maxScaleDenominator;
    int minScaleDenominator;
    QHash<QString, QString> pointSymbolizer;
    QString linePatternSymbolizer;
    QString polygonPatternSymbolizer;
    QHash<QString, QString> textSymbolizer;
    QHash<QString, QString> shieldSymbolizer;
    QHash<QString, QString> lineSymbolizer1;
    QHash<QString, QString> lineSymbolizer2;
    QHash<QString, QString> lineSymbolizer3;
    QHash<QString, QString> polygonSymbolizer;
    QHash<QString, QString> buildingSymbolizer;
    QHash<QString, QString> markersSymbolizer;
    bool debugSymbolizer;
};

typedef struct tstyleelement {
    QString name; // název stylu
    QList<classRule> rules; // uspořádaný seznam pravidel
} TStyleElement; // element Style

typedef struct tlayerelement {
    QHash<QString, QString> attributes; // atributy vrstvy
    QString name; // název vrstvy
    QList<QString> styleNames; // seznam použitých stylů
    QHash<QString, QString> datasources;
} TLayerElement; // element Layer
```

## Příloha B

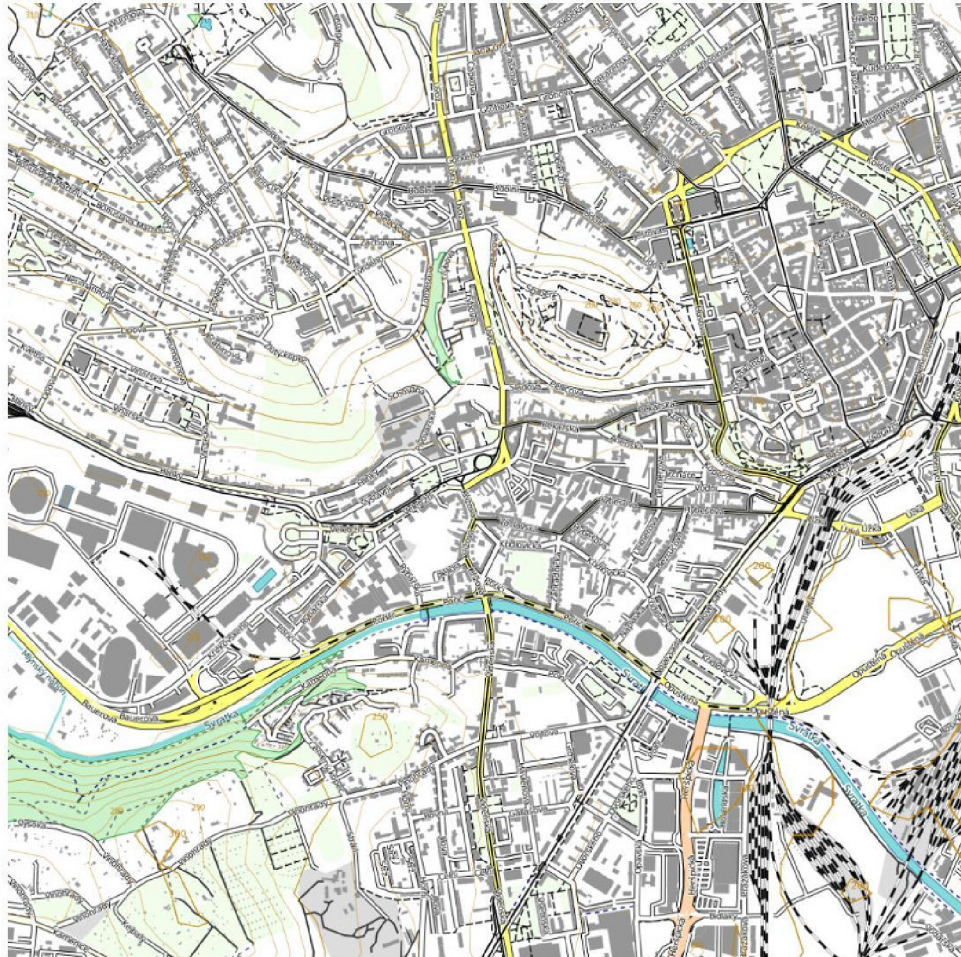
# Ukázka testovacích map

Následující mapy jsou zmenšené, takže jejich uvedené měřítko neodpovídá tomu, jak by se vytiskly v jejich původním rozlišení.



Brno

1:15000



Brno  
katastrální výměra: 230,22 km<sup>2</sup>  
počet obyvatel: 378 327 (1.1.2013)  
zeměp. souřadnice: 49° 12' s š, 16° 37' v d  
nadmořská výška: 190-479 m

Brno (německy Brunn) je počtem obyvatel i rozlohou druhé největší město v České republice, největší město na Moravě a bývalé hlavní město Moravy je sídlem jihomoravského kraje, v jehož centrální části tvoří samostatný okres Brno-město. Město leží na soutoku řek Svatavy a Svitavy a má zhruba 380 tisíc obyvatel, jeho regionální sídelní aglomerace pak má zhruba 780 tisíc obyvatel.

Město je významným střediskem vysokého školství s 33 fakultami třinácti univerzit a dalších vysokých škol s více než 89 000 studenty.

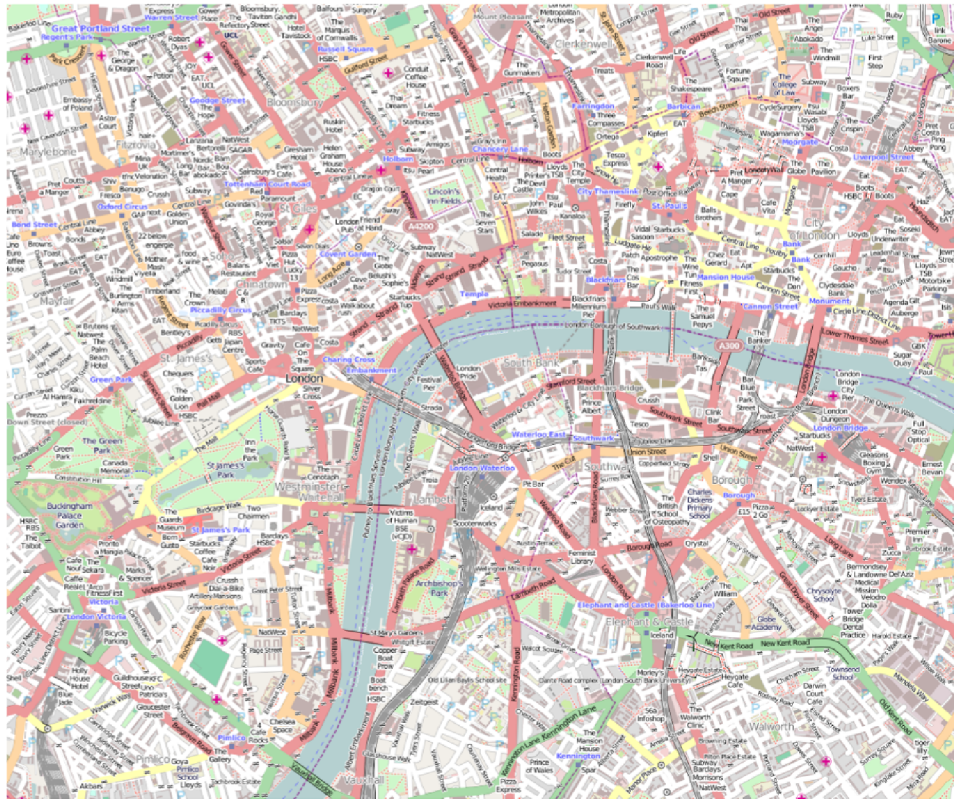
Na Brněnském výstavišti jsou tradičně konány velké mezinárodní výstavy a veletrhy. Rozsáhlý areál výstaviště započal svůj provoz již roku 1928 a dnes je považován také za jednu z kulturních památek města. Největší pořádanou událostí je Mezinárodní strojírenský veletrh. Brno proslulo i coby dějiště velkých motonostických závodů konaných na blízkém Masarykově okruhu, tato tradice sahá do třicátých let 20. století. K nejprestižnějším závodům patří Grand Prix ČR, součást seriálu Mistrovství světa silničních motocyklů [13]. Brno také každoročně hostí mezinárodní přehlídku ohňostrojů Ignis Brunensis, pořádanou od roku 1998. Navštěvnost této události se typicky pohybuje mezi jedním až dvěma sty tisíci návštěvníků každý den jejího konání.

K nejvýznamnějším dominantám města patří hrad a pevnost Špilberk na stejnojmenném kopci a katedrála svatého Petra a Pavla na vršku Petrov utvářející charakteristické panorama města a často vyobrazovaná jako jeho symbol. Druhým dochovaným hradem na území Brna je ve věří tyčící se nad Brněnskou přehradou. Další významnou památkou je funkcionalistická vila Tugendhat, která byla zapsána mezi světové dědictví UNESCO. Historické městské jádro bylo vyhlášeno městskou památkovou rezervací. K turisticky atraktivním lokalitám může patřit také začátek chráněné krajinné oblasti Moravský kras.

Obrázek B.1: Testovací mapa 01

# City of London

1:27000



## London

London is the capital city of England and the United Kingdom, and the largest city, urban zone and metropolitan area in the United Kingdom, and the European Union by most measures. Located on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium. London's ancient core, the City of London, largely retains its square-mile mediaeval boundaries. Since at least the 19th century, the name London has also referred to the metropolis developed around this core. The bulk of this conurbation forms the London region and the Greater London administrative area, governed by the elected Mayor of London and the London Assembly.

## Area

- Total 1,570 km<sup>2</sup> (607 sq mi)
- Population
- Total 8,173,194
- Density 5,205/km<sup>2</sup> (13,466/sq mi)

London is a popular centre for tourism, one of its prime industries, employing the equivalent of 350,000 full-time workers in 2003, while annual expenditure by tourists is around £15 billion. London attracts over 14 million international visitors per year, making it Europe's most visited city. London attracts 27 million overnight-stay visitors every year. In 2010 the ten most-visited attractions in London were:

British Museum, Tate Modern, National Gallery, Natural History Museum, London Eye, Science Museum, Victoria and Albert Museum, Madame Tussauds, National Maritime Museum, Tower of London

## Tower of London

Her Majesty's Royal Palace and Fortress, more commonly known as the Tower of London, is a historic castle on the north bank of the River Thames in central London, England. It lies within the London Borough of Tower Hamlets, separated from the eastern edge of the square mile of the City of London by the open space known as Tower Hill. It was founded towards the end of 1066 as part of the Norman Conquest of England.

## Buckingham Palace

Buckingham Palace is the official London residence and principal workplace of the British monarch.[1] Located in the City of Westminster, the palace is a setting for state occasions and royal hospitality. It has been a focus for the British people at times of national rejoicing and crisis.

## Tower Bridge

Tower Bridge (built 1886-1894) is a combined bascule and suspension bridge in London, over the River Thames. It is close to the Tower of London, from which it takes its name. The bridge consists of two towers tied together at the upper level by means of two horizontal walkways, designed to withstand the horizontal forces exerted by the suspended sections of the bridge on the landward sides of the towers.

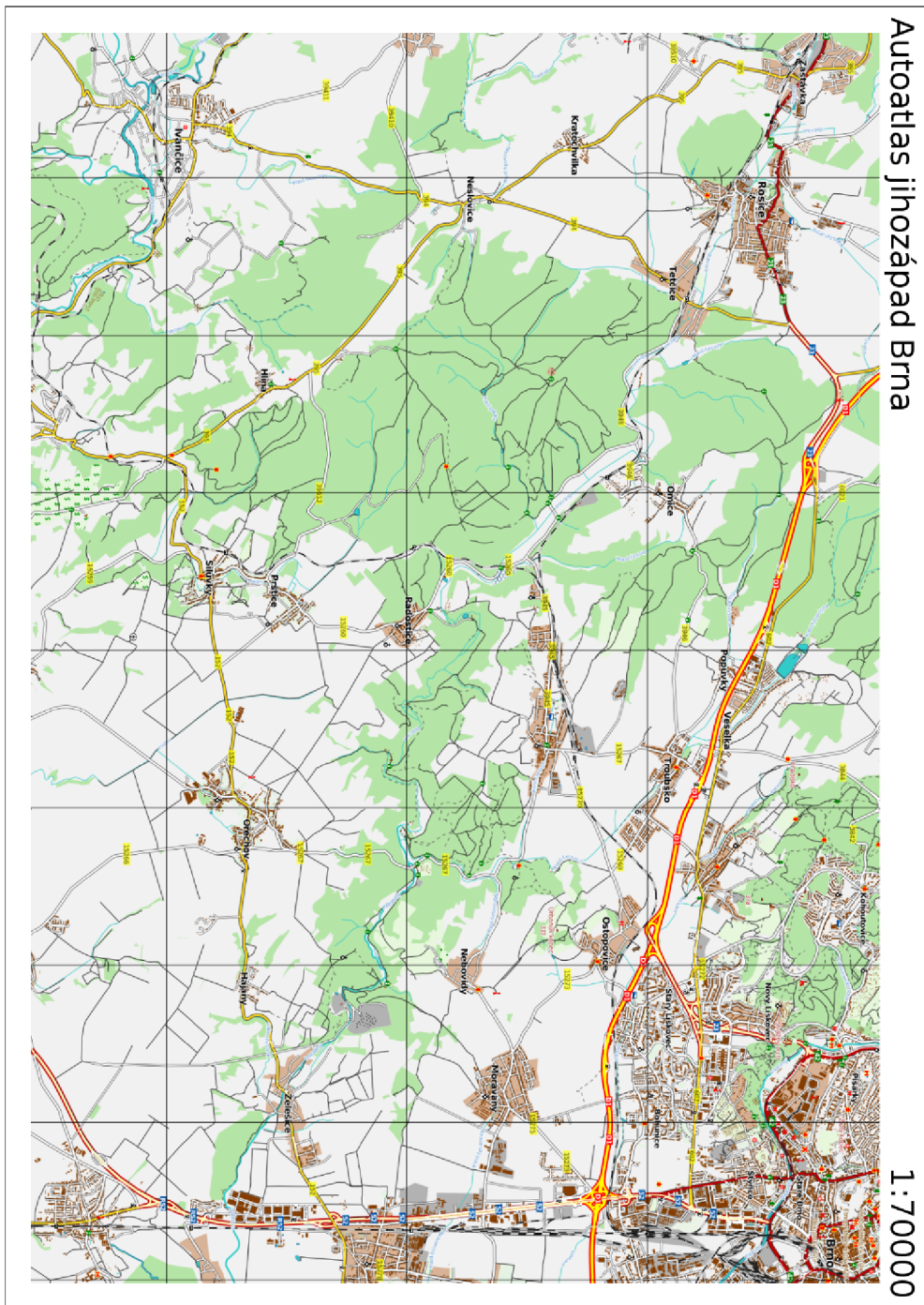
## Trafalgar Square

Trafalgar Square is a public space and tourist attraction in central London, United Kingdom, built around the area formerly known as Charing Cross. It is in the borough of the City of Westminster. At its centre is Nelson's Column, which is guarded by four lion statues at its base. There are a number of statues and sculptures in the square, with one plinth displaying changing pieces of contemporary art. The square is also used for political demonstrations and community gatherings, such as the celebration of New Year's Eve.

## Big Ben

Big Ben is the nickname for the great bell of the clock at the north end of the Palace of Westminster in London, and often extended to refer to the clock and the clock tower. The tower is now officially called the Elizabeth Tower, after being renamed to celebrate the Diamond Jubilee of Queen Elizabeth II. The Elizabeth Tower holds the largest four-faced chiming clock in the world and is the third-tallest free-standing clock tower. The tower was completed in 1858 and had its 150th anniversary on 31 May 2009, during which celebratory events took place. The Elizabeth Tower has become one of the most prominent symbols of both London and England and is often in the establishing shot of films set in the city.

Obrázek B.2: Testovací mapa 02



Obrázek B.3: Testovací mapa 03

# Příloha C

## Obsah CD

Na přiloženém CD se nachází následující adresáře a soubory:

- doc/ – dokumentace ve formátu doxygen
- examples/ – ukázky map a stylesheetů
- src/ – zdrojové kódy aplikace
- src-text/ – zdrojové kódy textu práce
- README.txt – informace o instalaci a spuštění aplikace