

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## IMPLEMENTACE GIS NÁSTROJE PRO MOBILNÍ POČÍTAČOVÁ ZAŘÍZENÍ

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VOJTĚCH KALČÍK

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

**IMPLEMENTACE GIS NÁSTROJE**  
**PRO MOBILNÍ POČÍTAČOVÁ ZAŘÍZENÍ**  
GIS APPLICATION TOOL FOR MOBILE PLATFORMS

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. VOJTĚCH KALČÍK**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. MARTIN HRUBÝ, Ph.D.**

BRNO 2013

## **Abstrakt**

Diplomová práce se zabývá návrhem GIS nástroje pro mobilní počítačová zařízení. Výsledný program bude přizpůsoben pro sběr geografických dat v terénu. Program bude pracovat se standardními formáty souborů pro GIS. Velký důraz je kladen na ovládání, které je na mobilních zařízeních s kapacitním displejem specifické. Program bude navržen tak, aby mohl běžet jak na smartphone, tak na tabletu. Součástí práce je vysvětlení principu mapování a popis geografických informačních systémů.

## **Abstract**

This master's thesis with draft of GIS tool for mobile platforms. The resulting program will be adapted for collection of geographic data in the field. The program will work with standard file formats for GIS. High emphasis is put on the control, which is particular for mobile devices with capacitive display. The program will be designed for both smartphone and tablet. Explanation of the mapping principle and description of geographic information systems are also parts of the thesis.

## **Klíčová slova**

mapování, GIS, mapové zobrazení, Proj.4, mapa, mobilní zařízení, smartphone, tablet, SpatiaLite, Shapefile

## **Keywords**

mapping, GIS, map projection, Proj.4, map, mobile device, smartphone, tablet, SpatiaLite, Shapefile

## **Citace**

Vojtěch Kalčík: Implementace GIS nástroje pro mobilní počítačová zařízení, diplomová práce, Brno, FIT VUT v Brně, 2013

# Implementace GIS nástroje pro mobilní počítačová zařízení

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Martina Hrubého, Ph.D.

.....  
Vojtěch Kalčík  
21. května 2013

## Poděkování

Rád bych poděkoval panu Ing. Martinovi Hrubému, Ph.D. za vedení a cenné odborné rady.

© Vojtěch Kalčík, 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 Úvod</b>	<b>3</b>
1.1 Motivace . . . . .	3
1.2 Cíl práce . . . . .	3
1.3 Stručný obsah kapitol . . . . .	3
<b>2 Úvod do mapování</b>	<b>5</b>
2.1 Historie a účel mapování . . . . .	5
2.2 Princip získávání dat . . . . .	5
2.3 Mapování jako modelování . . . . .	6
2.4 Postup mapování . . . . .	7
2.4.1 I. fáze - příprava . . . . .	7
2.4.2 II. fáze - realizace mapování . . . . .	8
2.4.3 III. fáze - následné zpracování . . . . .	8
2.5 Existující řešení . . . . .	8
<b>3 Technické aspekty GIS</b>	<b>10</b>
3.1 Geografický informační systém . . . . .	10
3.1.1 Mapové vrstvy . . . . .	10
3.2 Náhradní elipsoid . . . . .	11
3.3 Mapové zobrazení . . . . .	12
3.3.1 Mercatorovo zobrazení . . . . .	12
3.4 Geografický souřadný systém . . . . .	12
3.4.1 Systém šířka-délka (WGS-84) . . . . .	13
3.5 EPSG Geodetic Parameter Dataset . . . . .	13
3.6 Zjišťování geografické polohy . . . . .	14
3.6.1 Historické určování zeměpisné šířky a délky . . . . .	14
3.6.2 Trigonometrické určování délek . . . . .	14
3.6.3 Měření délek elektronickými dálkoměry . . . . .	15
3.6.4 Globální navigační satelitní systémy . . . . .	15
3.6.5 NAVSTAR GPS . . . . .	15
3.6.6 Diferenciální korekce . . . . .	16
3.7 Open Geospatial Consortium . . . . .	16
3.8 Open source knihovny používané pro GIS . . . . .	17
3.8.1 Proj.4 . . . . .	17
3.8.2 Java Map Projection Library . . . . .	17
3.8.3 JTS Topology Suite . . . . .	17
3.8.4 Geos . . . . .	18
3.8.5 SpatiaLite . . . . .	18

3.9	Formát Shapefile . . . . .	20
3.9.1	Soubor SHP . . . . .	21
3.9.2	Soubor SHX . . . . .	21
3.9.3	Soubor DBF . . . . .	24
3.10	Webové mapové služby . . . . .	24
3.11	Internetové dlaždicové mapy . . . . .	24
3.11.1	Formát dat . . . . .	26
3.11.2	Spherical Mercator Projection . . . . .	26
<b>4</b>	<b>Návrh grafického uživatelského rozhraní</b>	<b>28</b>
4.1	Principy mobilního uživatelského rozhraní . . . . .	28
4.2	Prvky GUI implementovaného programu . . . . .	29
4.2.1	Plátno pro vykreslení mapových vrstev . . . . .	30
4.2.2	Postranní panel . . . . .	31
4.2.3	Seznam mapových vrstev . . . . .	31
4.2.4	Vytváření prázdné mapové vrstvy . . . . .	32
4.2.5	Atributová tabulka . . . . .	32
4.2.6	Horní nástrojová lišta . . . . .	32
4.2.7	Editační tlačítka v plátně . . . . .	33
<b>5</b>	<b>Implementace programu</b>	<b>35</b>
5.1	Volba platformy . . . . .	35
5.2	Popis platformy Android . . . . .	35
5.2.1	Vývoj na platformě Android . . . . .	36
5.3	Dekompozice implementovaného programu . . . . .	37
5.4	Knihovny třetích stran . . . . .	38
5.5	Plátno pro zobrazení mapových vrstev . . . . .	39
5.6	Služby pro získání polohy . . . . .	39
5.7	Sklad OpenStreetMap dlaždic . . . . .	40
5.8	Import a export Shapefile . . . . .	40
5.9	SpatiaLite databáze . . . . .	41
5.10	Správce mapových vrstev . . . . .	42
5.11	Vektorové mapové vrstvy . . . . .	43
5.12	Dlaždicová mapová vrstva . . . . .	45
5.12.1	Knihovna Osmdroid . . . . .	46
5.12.2	Google Maps Android API . . . . .	46
<b>6</b>	<b>Protokol o mapování v terénu</b>	<b>47</b>
6.1	Postup mapování . . . . .	47
6.2	Výsledek mapování . . . . .	49
<b>7</b>	<b>Závěr</b>	<b>51</b>

# Kapitola 1

## Úvod

### 1.1 Motivace

V GIS (geografický informační systém) (kapitola 3.1) je nejnáročnějším problémem vybudování geodatabáze. Podle dokumentu [45, str. 49] více než 70 % nákladů na vytvoření aplikace padne na obstarání dat. Jednou z možností pořízení geografických dat je přímé mapování v terénu.

Mapování má dlouhou historii. Než byly zpřístupněny družicové navigační systémy pro civilní použití, bylo určování přesné polohy výsadou profesionálních geodetů s drahým vybavením [45, str. 76]. Pomocí družicového navigačního systému GPS (Global Positioning System), který je nejpoužívanější, dokážeme získat polohu dostatečně přesnou na to, aby mapovat v terénu mohl téměř každý.

Pro mapování v terénu využíváme mobilní počítačová zařízení. V odvětví mobilních počítačových zařízení probíhá prudký vývoj. Aplikace pro mapování v terénu, které jsme ještě před několika málo lety považovali za standardní, jsou už dnes naprosto nevyhovující hlavně způsobem ovládání (kapitola 4.1). Moderní smartphony a tablety, které se ovládají převážně prsty, vyžadují aplikace, které jsou tomuto konceptu přizpůsobené.

### 1.2 Cíl práce

Cílem diplomové práce je navrhnout a implementovat aplikaci pro získávání geografických dat v terénu s moderním mobilním zařízením. Je nutné, aby aplikace dodržela principy ovládání obvyklé u dnešních mobilních platforem. Program musí být datově kompatibilní se standardními GIS nástroji.

Přínos implementovaného programu je v pohodlném ovládání v terénu a pokročilých možnostech tvorby a editace geografických objektů.

### 1.3 Stručný obsah kapitol

Druhá kapitola se zabývá úvodem do mapování. Shrnuje stručně historii a účel tvorby map. Popisuje principy získávání dat, praktický postup mapování a existující programy, které sběr dat v terénu umožňují. Třetí kapitola rozebírá geografické informační systémy a technické aspekty, které s nimi souvisí. Popisuje souvislosti s modelováním geoprostoru, jako je náhradní elipsoid, geografický souřadný systém a mapové zobrazení. Kromě toho popisuje některé existující geografické knihovny a formáty. Čtvrtá kapitola se zabývá návrhem

grafického uživatelského prostředí. Pátá kapitola se zabývá implementačními aspekty programu. Popisuje použité knihovny a detaily, které souvisí s výslednou implementací. V šesté kapitole se nachází protokol popisující experimentální mapování v oblasti Údolí Bílého potoka u obce Veverská Bítýška. Sedmou kapitolou je závěr.

## Kapitola 2

# Úvod do mapování

Diplomová práce se zabývá implementací programu pro mapování v terénu. Proto je důležité nejdříve popsat, jaký je princip mapování. Na začátku kapitoly je popsána stručně historie a účel mapování. Dále se zabývá možnostmi získávání geografických dat. Na konci kapitoly jsou uvedeny existující mobilní aplikace vhodné pro mapování.

### 2.1 Historie a účel mapování

Žijeme ve světě, kde jsme obklopeni trojrozměrnými objekty. Lidé už od dávné historie zakreslují zmenšený obraz reálné krajiny na dvojrozměrnou plochu s úmyslem zachytit prostorovou informaci světa kolem sebe. Jeden z nejstarších nálezů, které se považují za plán krajiny, je rytina na mamutím klu nalezená na Jižní Moravě nedaleko Pavlova [47, kapitola 1.2.1]. Její stáří se odhaduje na 25000 let. S prohlubujícím poznáním světa lidstvo vylepšovalo a zpřesňovalo i mapy. Postupem času se vytvořil vědní obor kartografie, který se zabývá sestavováním map všeho druhu [47, kapitola 1.1.1].

Největší zájem na kvalitních mapových dílech mají armády [45, str. 13]. Armáda měla vždy nejkvalitnější mapy a na jejich tvorbu vynakládala hodně prostředků. Druhým největším uživatelem mapových děl je státní správa a samospráva [45, str. 13]. Mapy se využívají v mnoha odvětvích, jako je stavebnictví, doprava, památková ochrana, ochrana životního prostředí, katastr nemovitostí, zemědělství, inženýrské sítě a mnoho dalších.

### 2.2 Princip získávání dat

Rozlišujeme dva zdroje geo-údajů: primární a sekundární [45, str. 49]. Primární zdroje jsou veškeré formy měření v terénu, které provádíme za účelem budování naší geodatabáze. Sekundární zdroje jsou různé formy získávání dat z již hotových geodatabází. Obecně platí, že použití primárních zdrojů je pracnější, dražší a časově náročnější, avšak někdy není jiná možnost.

Prostorové i neprostorové údaje lze zadávat ručně nebo automaticky z nějakého senzoru. U neprostorových údajů je obvyklé, že uživatel zadá hodnotu na základě své znalosti. Automaticky může být zadán například čas zaznamenání nebo teplota. U prostorových je netypické, že by uživatel zadával přímo geografické souřadnice. Spíše se využívá výběr pozice z okna, ve kterém jsou zobrazena již existující prostorová data. Ta pomáhají ve výběru správné souřadnice. Automatický záznam prostorových informací se dělá obvykle pomocí družicového navigačního systému. Nejvíce používaným polohovacím systémem je americký

GPS. Jinou možností získání souřadnic je využití znalosti pozice vysílačů bezdrátových sítí a síly signálu jimi vysílaným. Mohou to být například Wi-Fi nebo GSM vysílače.

Pro získávání prostorových informací je ideální, když je možné zkombinovat družicový navigační systém a zároveň existující prostorová data.

## 2.3 Mapování jako modelování

Mapování je forma modelování geografického prostoru. Při modelování procházíme následujícími etapami [45, str. 10]:

1. *zkoumání reality a poznávání všech jejích aspektů,*
2. *formulování své úlohy; výsledkem této etapy by měl být jistý konceptuální model, kde zvýrazníme ty aspekty reality, které jsou relevantní pro naši úlohu,*
3. *realizace úlohy,*
4. *analýza údajů - snaha o dosažení vytyčeného cíle.*

Prostor lze chápat dvěma různými způsoby [45, str. 11]. Absolutní prostor jako soubor poloh s definovanými vlastnostmi, kterému odpovídá rastrová mapa, a relativní prostor jako soubor objektů s prostorovými vlastnostmi, jemuž odpovídá mapa vektorová. V diplomové práci se budu zabývat převážně relativním chápáním prostoru.

Geoobjekty jsou fyzické nebo abstraktní objekty, které obsahují geografickou informaci. *Geografická informace je ucelený údaj o hmotném nebo nehmotném objektu, kde její nutnou součástí je údaj o geografické poloze objektu* [45, str. 6]. Každý geoobjekt je určen následujícími údaji [45, str. 12]:

1. *identifikací - pojmenování objektu libovolným způsobem (včetně reference),*
2. *prostorovou polohou - udáním geografické souřadnice (případně tvaru ve formě posloupnosti souřadnic),*
3. *atributy - neprostorové údaje,*
4. *časem - pouze v některých aplikacích.*

Modelované údaje o objektu závisí na účelu výsledného modelu. Prostorové údaje geoobjektů, které mají v reálném světě tři dimenze, obvykle modelujeme s menším počtem dimenzí [45, str. 25]. Vezměme si například město. V některých případech nám stačí znát pouze jednu souřadnici, která udává polohu města na povrchu Země. V jiném případě bychom mohli potřebovat znát plochu a tvar města, takže bychom ho modelovali jako polygon. Geoobjekty modelujeme podle následujícího počtu dimenzí [45, str. 25]:

- *0D - bod s definovanou polohou,*
- *1D - úseky čar s konečnou délkou a nulovou plochou,*
- *2D - polygony.*

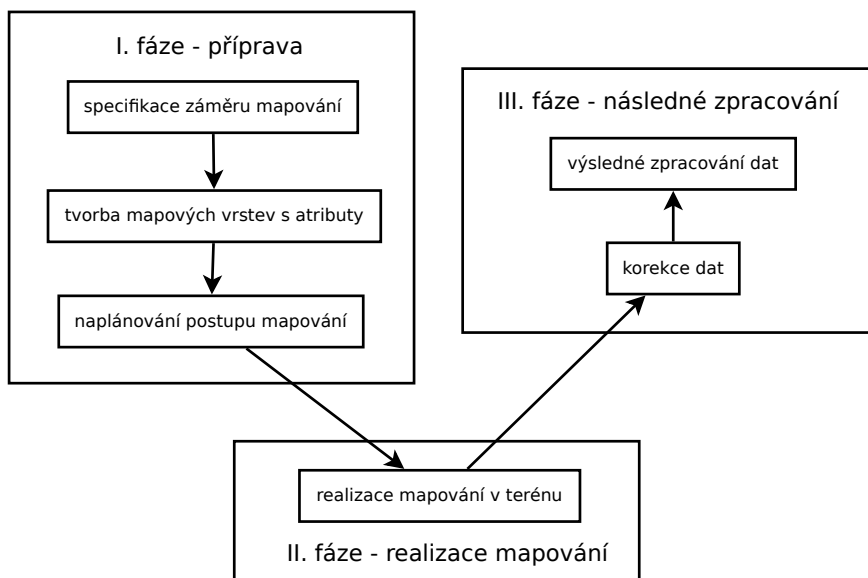
Podobně má smysl uvažovat pouze ty neprostorové údaje, které jsou pro nás zajímavé. Například pro město nás může zajímat počet obyvatel, ale počet středních škol už může být nadbytečná informace. Zaznamenat časový údaj může být zajímavé například pro teplotu na určité souřadnici.

Kromě samotných geoobjektů popisujeme i jejich vzájemné vztahy. Rozlišujeme geometrický a topologický popis vztahu geoobjektů [45, str. 12]. Geometrickým popisem může být například vzdálenost geoobjektů. Naproti tomu topologie popisuje logické vazby mezi geoobjekty. Topologickou informací může být, že řeka Vltava se vlévá do řeky Labe.

Při mapování oddělujeme geoobjekty se stejným tématem a typem do samostatné vrstvy. Například potoky uložíme do jiné liniové vrstvy než cesty, i když je budeme modelovat také jako liniové objekty. Zmíněný přístup má několik důvodů. Každá vrstva má definované atributy pro geoobjekty, které do ní patří a které jsou obvykle specifické pro dané téma. Rozdělení do vrstev vede k většímu pořádku v datech. Vycházíme z poznání, že tématické vrstvy je snadné sloučit, ale komplikované rozdělit [45, str. 26]. Oddělení témat do vrstev umožňuje jednodušší sdílení a přenášení pouze určitých témat.

## 2.4 Postup mapování

Diplomová práce se zabývá primárně mapováním v terénu. Smyslem terénního mapování je to, že se uživatel při mapování fyzicky nachází na místě modelovaného geoobjektu. Mapování lze rozdělit do třech fází, které jsou vidět na obrázku 2.1.



Obrázek 2.1: Diagram postupu mapování.

### 2.4.1 I. fáze - příprava

Příprava na mapování obvykle probíhá na desktopovém počítači kvůli většímu komfortu, ale nic nám nebrání ji provést na mobilním počítačovém zařízení. Nejdříve je nutné si určit, jaký je záměr mapování a co má být výsledkem. Na tom záleží použité mapové vrstvy a jejich atributy, které je vhodné si připravit dopředu.

Pro zvýšení efektivity mapování je výhodné naplánovat postup v terénu. Pro plánování obvykle používáme existující mapu oblasti, ve které se chystáme mapovat. Podle prostředí a mapovaných objektů zvolíme, jestli pro mapování využijeme nějaký dopravní prostředek nebo ho budeme realizovat pěšky. Je dobré si určit alespoň přibližnou trasu, aby se zamezilo chaotickému postupu mapovaným územím.

### 2.4.2 II. fáze - realizace mapování

Při mapování se využívá elektronický geodetický zápisník [45, str. 50]. Jedná se o program, který umožňuje zaznamenání získaných informací. V kapitole 2.5 jsou popsány některé existující implementace.

Jak bylo popsáno v kapitole 2.3, budeme rozlišovat tři typy geoobjektů. Prostorová informace geoobjektu bude reprezentována pouze posloupností souřadnic. Pro 0D geoobjekt to bude pouze jedna souřadnice, pro 1D libovolně dlouhá posloupnost souřadnic. Pro 2D má posloupnost spojenou první a poslední souřadnici a definuje obvod polygonu.

Samotné mapování je činnost, při které se zaznamenávají dvoudimenzionální souřadnice modelovaných geoobjektů. Souřadnice získáváme z družicového polohovacího systému. Je vhodné, aby měl uživatel možnost získané souřadnice upravovat nebo vytvářet i manuálně během mapování hlavně z důvodu nepřesností polohovacího systému. Další dobrou vlastností geodetického zápisníku je možnost zaznamenat i topologii objektů jejich přesným propojením. Kromě prostorových údajů se uloženým objektům přiřazují atributy jako neprostorové údaje.

Způsob zaznamenání souřadnic z družicového navigačního systému může být manuální, kdy si uživatel vybere časový okamžik, kdy zaznamenat souřadnici, nebo automatický, při kterém se souřadnice zaznamenávají automaticky po určité vzdálenosti od poslední zaznamenané souřadnice.

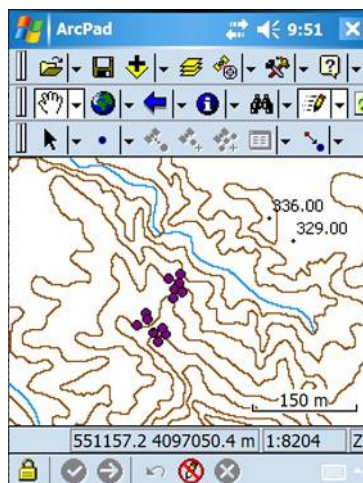
### 2.4.3 III. fáze - následné zpracování

Po zaznamenání geografických dat v terénu obvykle proběhne jejich zpracování na desktopovém počítači, jehož výstupem jsou data, která jsou cílem celého mapování. Data vzniklá při mapování v terénu nemusí být dokonalá. Na mobilních počítačových zařízeních nebývá příliš pohodlné psaní textů, proto mohou být atributy vyplněny například bez diakritiky, mohou obsahovat překlepy nebo se dokonce vyplnění některých atributů může nechat až do této fáze. Proto je vhodné provést korekci vytvořených dat. Někdy se provádí i korekce prostorových dat buď porovnáním s jinými geografickými daty, například ortofotomapou, nebo diferenciální korekcí (kapitola 3.6.6).

## 2.5 Existující řešení

Standardním elektronickým geodetickým zápisníkem je program ArcPad (obrázek 2.2) [20]. ArcPad vytvořila firma Esri. Běží na platformě Windows Mobile. Pro ovládání je nutný stylus a celý program vypadá jako zmenšený desktopový GIS (kapitola 3.1). Ovládání programu ArcPad je jeho slabinou. Uživatelské rozhraní plné malých ikon a nabídek, na které lze kliknout jedině pomocí stylusu, je velice nepohodlné.

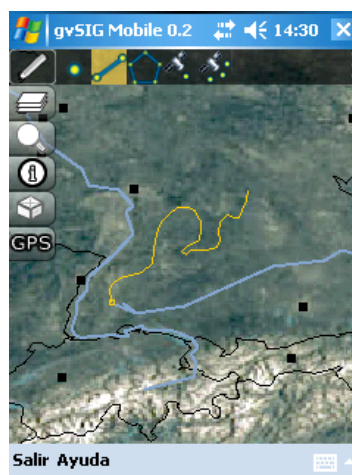
Pro moderní mobilní zařízení se systémy Android, iOS a Windows Phone vydala Esri jiný program, který se jmenuje ArcGIS for Mobile [19]. ArcGIS for Mobile vyžaduje připojení k internetu, bez kterého není možné dělat jakoukoli činnost. Pracuje pouze s online



Obrázek 2.2: Screenshot aplikace ArcPad.

daty z ArcGIS Online nebo ArcGIS for Server. Osobně považuji čistě on-line řešení pro terénní mapování nevhodné. Datové služby jsou u nás na mnoha místech nedostatečné.

Z open source nástrojů můžeme zmínit program gvSIG Mobile, který je zahrnutý v projektu gvSIG, jehož součástí je i desktopový GIS nástroj. GvSIG Mobile obsahuje pokročilé funkce. Umí pracovat s několika vektorovými i rastrovými formáty, do kterých patří i Shapefile [38]. Vektorové vrstvy dokáže vytvářet i editovat. GvSIG Mobile běží pouze na platformě Windows Mobile a jeho ovládání má podobné problémy jako program ArcPad. Screenshot aplikace je na obrázku 2.3.



Obrázek 2.3: Screenshot aplikace gvSIG.

Pro platformu Android existují programy SHP Viewer [26] a Shapefile over Map [25]. Oba umožňují načtení Shapefile souborů (kapitola 3.9), ale nepodporují editaci prostorových údajů. První slouží pouze k prohlížení a druhý dokáže navíc editovat atributy.

Program, který by byl přizpůsobený ovládání prsty pomocí dotykového kapacitního displeje, umožňoval off-line zaznamenávání geografické informace a podporoval export a import do formátu Shapefile jsem, neobjevil.

## Kapitola 3

# Technické aspekty GIS

V následujících sekcích budou popsány GIS technologie. V GIS existuje několik velkých open source projektů. Díky nim jsou k dispozici kvalitní knihovny, které umí počítat s různými náhradními elipsoidy (kapitola 3.2) a geografickými souřadnými systémy (kapitola 3.4), které jsou nedílnou součástí GIS.

### 3.1 Geografický informační systém

Program implementovaný v rámci diplomové práce má přímou vazbu na GIS. Jednak jeho hlavním smyslem je tvorba dat pro desktopové GIS, ale i on sám je jednoduchým GIS nástrojem.

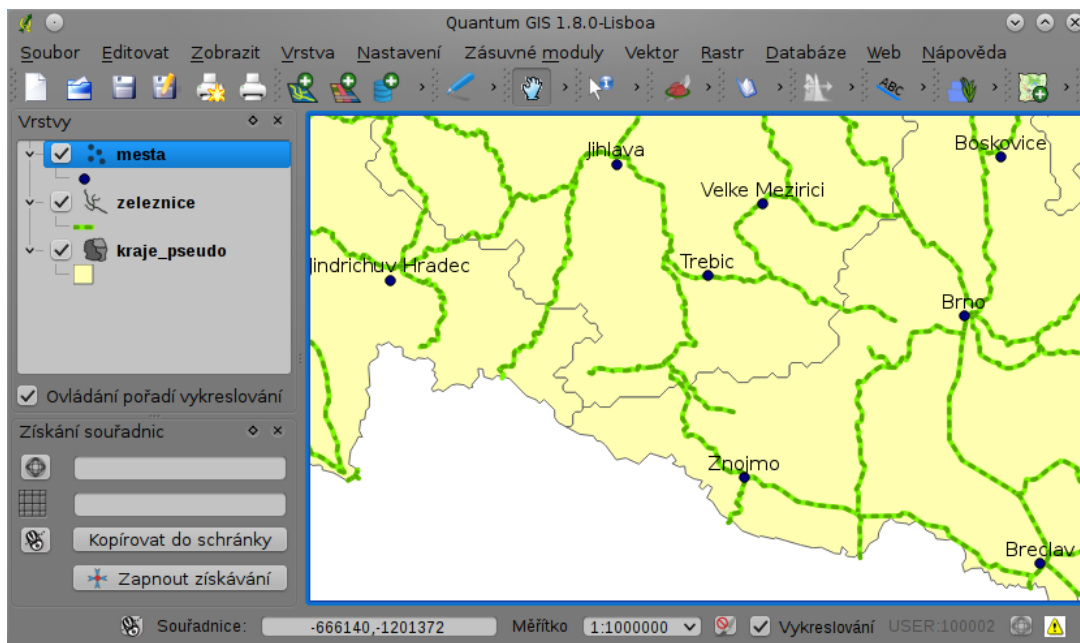
*GIS je na počítačích založený informační systém pro získávání, obhospodařování, analýzu, modelování a vizualizaci geoinformací. Geo data, která využívá, popisují geometrii, topologii, tematiku (atributy) a dynamiku (změny v čase) geoobjektů [45, str. 8].*

Na obrázku 3.1 je vidět screenshot open source nástroje Quantum GIS. Rozložení uživatelského rozhraní je velice podobné jako v jiných GIS programech. Nejvíce místa obvykle zabírá plátno, na kterém se vykreslují mapové vrstvy (kapitola 3.1.1). Vedle plátna bývá postranní panel, který obsahuje seznam mapových vrstev. Pořadí vrstev v seznamu je možné měnit, protože určuje překrytí vrstev. U každé vrstvy je zobrazeno, o jaký typ se jedná a jakým stylem se vykresluje do plátna. Jednotlivé vrstvy je možné skrývat. Ve vrchní části okna bývá klasická lišta nástrojů.

#### 3.1.1 Mapové vrstvy

Základní podobou geodatabázového souboru v GIS je pojem mapové vrstvy [45, str. 26]. Každá mapová vrstva by měla obsahovat pouze jedno téma, například řeky. Mapové vrstvy dělíme na vektorové a rastrové. Každá mapová vrstva musí být definována, v jakém geografickém souřadném systému (kapitola 3.4) je uložena prostorová informace.

Vektorová mapová vrstva je tvořena množinou geoobjektů [45, str. 28]. Rozlišujeme tři typy geoobjektů: bodové, liniové a polygonové. Každá vektorová mapová vrstva obsahuje pouze objekty jednoho typu. Kromě prostorové informace obsahuje vektorová vrstva i atributovou tabulku, ve které jsou uloženy údaje neprostorové. Sloupce tabulky definují, jaké informace mohou být o objektech uloženy. Každý sloupec má definován i datový typ informace. Typicky se využívá textový nebo číselný typ. Každý řádek tabulky obsahuje informace o jednom objektu.



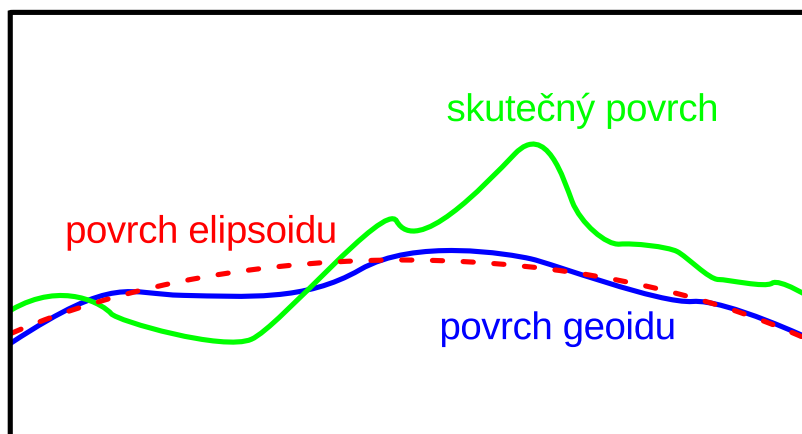
Obrázek 3.1: Screenshot aplikace Quantum GIS.

Rastrová vrstva reprezentuje obdélníkový výřez prostoru [45, str. 28]. Je vhodná pro údaje platné pro celý výřez. Spojitou informaci diskretizujeme obvykle do čtvercových plošek (buněk) tvořících mozaiku. Každá buňka rastru nese (typicky číselnou) atributovou hodnotu sledovaného jevu v daném prostoru. Rozlišujeme dva typy atributových hodnot. Hodnota popisovaného prostorového procesu, kterým může být například nadmořská výška. Druhým typem je kód, který nazýváme kategorií. Jde o výčtový typ. Například půdní typ: 1 = jíl, 2 = písek, 3 = vápenitá, ... K takové rastrové vrstvě je třeba mít překladovou tabulku dávající sémantiku číselným kategoriím.

## 3.2 Náhradní elipsoid

Povrch zeměkoule je velice členitý a nepravidelný. Pro potřebu modelování povrchu zeměkoule do dvojrozměrného prostoru je nutné tvar zjednodušit. Elementární zjednodušení zeměkoule je na těleso, které se nazývá geoid. Vychází z idealizované plochy zemského povrchu, která byla definována jako plocha, kde v každém jejím bodě má zemská tíže stejnou hodnotu [45, str. 15]. Tuto plochu pokládáme na úroveň střední hladiny moří. Na obrázku 3.2 je vidět, jaký je rozdíl mezi skutečným povrchem, povrchem geoidu a povrchem referenčního elipsoidu.

Geoid se dá velice těžko matematicky popsat, a proto se zavádí další zjednodušení. Zeměkoule se zjednodušuje na elipsoid, který se nazývá náhradní. Jde o rotační elipsoid, který rotuje kolem kratší z poloos. Bývá definován tak, aby jeho střed ležel ve středu Země a aby se co nejlépe přimykalo geoidu [45, str. 16]. Mezi používané elipsoidy patří WGS-84, Basselův elipsoid nebo Krasovského elipsoid.



Obrázek 3.2: Povrch geoidu.

### 3.3 Mapové zobrazení

Při definici geografického souřadného systému (kapitola 3.4) je důležité použít zobrazení.

Pro vytvoření mapy určitého území je nutné převést celý nebo část povrchu referenčního elipsoidu (kapitola 3.2) do dvojrozměrné plochy [47, kapitola 1.4]. Převod s sebou nese vždy zkreslení. Podle toho, na co se bude výsledná mapa používat, je potřeba vybrat takové zkreslení, které bude vadit co nejméně.

Existují mapová zobrazení, která jsou globální pro celou zemkouli. Například Mercatorovo zobrazení (kapitola 3.3.1). Existují i mapová zobrazení, která jsou přesná pro určité území, třeba stát, ale jinde dochází k velkému zkreslení. Mezi takové patří například Křovákovo zobrazení, využívané souřadným systémem S-JTSK, které je velice přesné pro území bývalého Československa, ale v globálním měřítku je nepoužitelné [45, str. 19].

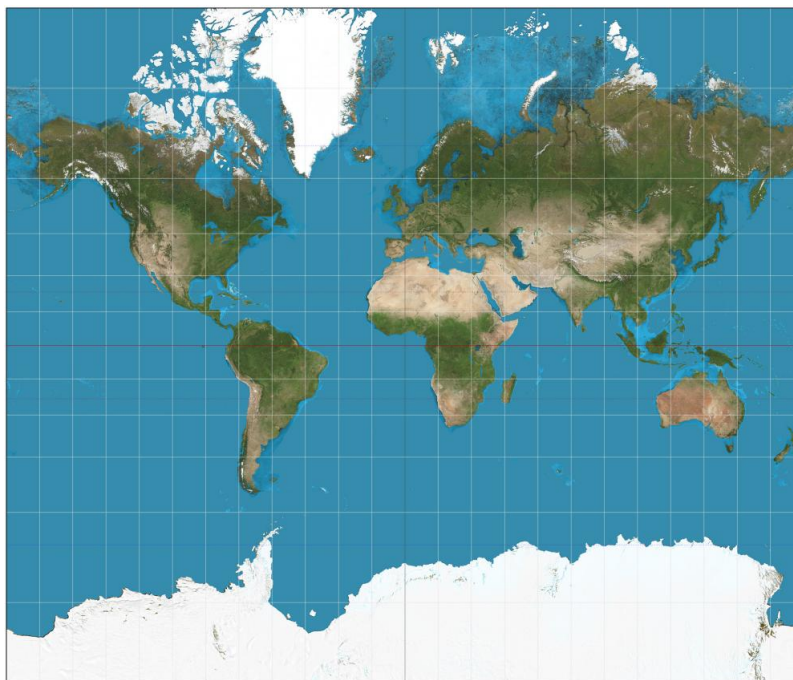
#### 3.3.1 Mercatorovo zobrazení

Mercatorovo zobrazení je pro diplomovou práci důležité, protože z něho vychází zobrazení Spherical Mercator Projection (kapitola 3.11.2), jehož vlastností je využito v podkladové mapě.

Mercatorovo zobrazení (obrázek 3.3) je válcové úhlojevné zobrazení (kapitola 3.3) v normální poloze [4]. To znamená, že elipsoid se promítá na válec, který je v takové poloze, že osa protínající oba póly prochází středem jeho podstav. Úhlojevné znamená, že naměřený azimut na mapě odpovídá azimutu ve skutečnosti. Proto se využívá zejména na námořních a leteckých navigačních mapách. Síť poledníků a rovnoběžek je pravoúhlá. V oblasti rovníku je zkreslení délek a ploch zanedbatelné, ale směrem k pólům se zvětšuje. Grónsko se jeví zhruba stejně velké jako Afrika, přestože ve skutečnosti je skoro 14 krát menší. Vzdálenosti mezi rovnoběžkami se směrem k pólům zvětšují až do nekonečna. Póly se zobrazí na plášť válce v nekonečnu. Oblasti se zeměpisnou šířkou větší než  $70^\circ$  jsou již značně zkreslené a oblasti se zeměpisnou šířkou větší než  $85^\circ$  se už obvykle nezobrazují.

### 3.4 Geografický souřadný systém

Geografický souřadný systém je nástroj pro vyjádření polohy na povrchu Země [45, str. 19]. Pro nástroje typu GIS je důležité, aby měly definováno, se kterým souřadným systémem



Obrázek 3.3: Mercatorovo zobrazení. (převzato z [4])

pracují, a dokázaly transformovat souřadnice mezi systémy. Geografické souřadné systémy rozlišujeme na globální a lokální. Snahou globálních je postihnout celý geografický prostor Země. Prostor globálního souřadného systému je povrch elipsoidu (kapitola 3.2). Lokální geografické systémy jsou vytvořené pro popis menšího území, například státu. V lokálních souřadných systémech je dosaženo větší přesnosti a umožňují využít Eukleidovskou metriku pro výpočet vzdálenosti mezi dvěma body.

#### 3.4.1 Systém šířka-délka (WGS-84)

Souřadnice každého bodu na povrchu elipsoidu se skládá ze zeměpisné šířky a délky.

Zeměpisná šířka je dána úhlem mezi normálou v bodě a rovinou rovníku [5]. Z toho vyplývá, že bod ležící na rovníku má zeměpisnou šířku  $0^\circ$  a póly  $90^\circ$ . Zeměpisná šířka se rozděluje podle polokoule na severní a jižní. Pro počítačové vyjádření je vhodnější odlišit severní a jižní šířku znaménkem. Severní šířce dáváme kladné znaménko a jižní záporné [45, str. 19].

Zeměpisná délka se určuje obdobným způsobem jako zeměpisná šířka. Je dána úhlem mezi rovinou nultého poledníku a rovinou poledníku procházejícího bodem. Nultý poledník je stanoven dohodou na úroveň hvězdárny v Greenwich nedaleko Londýna [45, str. 21]. Zeměpisná délka nabývá hodnoty  $0^\circ - 180^\circ$ . Dělí se na východní a západní. Východní délce dáváme znaménko kladné a západní záporné.

### 3.5 EPSG Geodetic Parameter Dataset

EPSG Geodetic Parameter Dataset je databáze, která obsahuje definice používaných geografických souřadných systémů (kapitola 3.4) a náhradních elipsoidů (kapitola 3.2). Od

roku 2005 je spravována asociací The International Association of Oil & Gas producers (OGP) [31].

Protože je databáze celosvětová, tak se hojně využívá v GIS nástrojích pro jednoznačnou definici geografických souřadných systémů. Každý souřadný systém obsahuje jedinečný kód, který ho identifikuje.

## 3.6 Zjišťování geografické polohy

Pro mapování je základem zjištění přesné geografické polohy. S vývojem technologií se vyvíjely i možnosti a přesnost určování polohy. V dřívějších dobách se jednalo hlavně o určení zeměpisné šířky a délky, což mělo asi největší význam pro navigaci při plavbách po moři.

V moderní geodézii jsou všechny metody v zásadě stejné a liší se jenom technickým provedením. Ať použijeme jakoukoli technologii zjišťování polohy, absolutně vždy budeme vyhodnocovat svou vzdálenost od třech bodů se známou referenční polohou [45, str. 76]. Zmíněný způsob se nazývá trilaterace.

Dříve se využívalo odvozování polohy z triangulační sítě měřením úhlů [50, str. 34]. V šedesátých a sedmdesátých letech minulého století se začalo využívat měření vzdáleností pomocí světelných elektronických dálkoměrů [50, str. 34]. S možností využití družicových navigačních systémů i pro nevojenské účely se otevřely nové možnosti určování polohy.

### 3.6.1 Historické určování zeměpisné šířky a délky

Nejstarším způsobem je určování polohy v souřadném systému šířka-délka (kapitola 3.4.1). Určení zeměpisné šířky je jednodušší [45, str. 19]. Zeměpisná šířka lze určit v noci zjištěním úhlu mezi horizontem a polárkou nebo ve dne maximální výškou slunce nad obzorem. Pro zjištění zeměpisné šířky se využívaly v průběhu historie následující přístroje: astroláb, Jakubova hůl, kamal a sextant [2]. Sextant je ze zmíněných přístrojů nejpřesnější. První přímý předchůdce sextantu byl navržen v roce 1731.

Zjišťováním zeměpisné délky byl dlouho velký problém. Dokument [45, str. 20] popisuje její historii a způsob zjištění následovně. Zeměpisná délka nemá přirozenou vztahnou rovinu, kterou je pro zeměpisnou šířku rovník. Zeměpisnou délku je možné určit ze znalosti rozdílu mezi slunečním časem na aktuální pozici a slunečním časem ve vztahném bodě. Problémem bylo, že neexistovaly dostatečně přesné hodiny, aby bylo možné mít při plavbě stále přesný sluneční čas vztahného místa. V roce 1707 ztroskotalo pět bitevních anglických lodí u pobřeží Scilly Isles. Při katastrofě zahynulo více než 2000 anglických námořníků. Kvůli této události vypsala tehdejší anglická královna výzkumný grant na vyřešení problému určování zeměpisné délky. Grant vyhrál John Harrison v roce 1713 sestrojením přesných hodin. Během historie existovalo více vztahných poledníků až se nakonec určil jeden společný Greenwichský.

### 3.6.2 Trigonometrické určování délek

Dokument [50, str. 38-39] popisuje způsob nepřímého (trigonometrického) určování délek. Trigonometrické určování délek je založeno na odvození délky z geometrických obrazců, obvykle trojúhelníků. Až do 60. let minulého století měla geodézie k dispozici pouze přístroj teodolit, který dokázal měřit jen úhly. Proto bylo zjištění pozice založeno na přepočtu délek podle úhlů mezi vztahnými body se známou pozicí.

### 3.6.3 Měření délek elektronickými dálkoměry

Dokument [50, str. 34] popisuje elektronické dálkoměry. S objevem dostatečně přesných elektronických dálkoměrů přešla geodézie převážně na měření délek pomocí nich. Díky tomu byly nahrazeny triangulační (úhlové) sítě sítěmi délkovými o dlouhých stranách.

Elektronické dálkoměry jsou založeny na dvou různých principech. Vyráběly se jak rádiové, tak světelné dálkoměry. Rádiové dálkoměry měly dosah i více jak 100 km. Světelné dálkoměry měly dosah do 70 nebo 80 km, ale byly přesnější. Pro běžné geodetické práce stačí dálkoměry s menším dosahem. Díky tomu se dnes vyrábí převážně světelné dálkoměry s dosahem do několika kilometrů. Velký dosah má význam hlavně při budování geodetických polohových základů, ale ty se dnes budují hlavně pomocí družicových metod.

Geodeti dnes využívají hybridní přístroje, které se nazývají totální stanice. Totální stanice v sobě obsahuje elektronický teodolit, dálkoměr, počítač a různé další pomůcky a zařízení umožňující zrychlit a zkvalitnit měření všech veličin a jeho zpracování [50, str. 17].

### 3.6.4 Globální navigační satelitní systémy

Globální navigační satelitní systémy (anglicky Global Navigation Satellite System - GNSS) slouží k získání geografické polohy na celém světě. GNSS jsou budovány jako pasivní dálkoměrné systémy. Pasivní znamená, že přijímač pouze přijímá signál z družic a nic nevysílá. To má důvody jednak vojenské (vysíláním by mohla být prozrazena přítomnost), ale také technické (konstrukce antén a spotřeba energie) [44, str. 12]. Přijímač určuje svoji vzdálenost od několika družic a na základě protínání těchto vzdáleností určí svoji polohu. Tato metoda se nazývá trilaterace [45, str. 77].

Určování vzdálenosti přijímače od družice lze provádět na základě kódových, fázových a doplerovských měření [44, str. 12-13].

- Kódová měření: Družice vysílá dálkoměrné kódy, které obsahují časové značky, ze kterých se dá zjistit, v jakém čase byl signál odeslán. Přijímač na základě časového rozdílu mezi odesláním a přijetím signálu spočítá vzdálenost. Dané měření je nejméně spolehlivé, ale využívá se ve většině navigačních a turistických přijímačů.
- Fázová měření: Jsou založena na znalosti vlnové délky signálu. Stačí spočítat celé vlny a délku poslední nedokončené vlny. Fázové měření je přesnější než předešlé a používá se například pro geodetické účely.
- Dopplerovská měření: Jsou založena na fyzikálním principu Dopplerova jevu. Dopplerův jev popisuje změnu frekvence a vlnové délky přijímaného signálu oproti vysílanému signálu, pokud se přijímač nebo vysílač pohybuje. V praxi se toto měření používá především při zjišťování rychlosti pohybu přijímače.

Z provozovaných globálních navigačních systémů zmiňme NAVSTAR GPS provozovaný americkou armádou a GLONASS provozovaný ruskou armádou. Evropská unie pracuje na GNSS Galileo, který ještě není dokončen. Galileo bude oproti předchozím dvěma civilní GNSS.

### 3.6.5 NAVSTAR GPS

NAVSTAR GPS je nejrozšířenější a nejpoužívanější GNSS. GPS je popsána v knize [44, str. 10-14].

GPS se skládá ze tří segmentů:

- uživatelský segment - je tvořen anténami a procesory přijímačů a poskytuje uživateli informace o poloze, rychlosti a přesném čase,
- kosmický segment - je tvořen soustavou umělých družic Země obíhajících po známých oběžných drahách,
- řídicí segment - je tvořen sadou pozemních stanic a slouží k řízení, monitorování a údržbě družic.

Družice vysílají signály na dvou nosných frekvencích, které se označují jako L1 a L2. Na frekvenci L1 (1 575,42 MHz) jsou vysílány dva dálkoměrné kódy. Jedná se o P-kód, který může být šifrován a používá se pro vojenské účely, a C/A kód, který není šifrován a používá se ve většině běžných přijímačů k určování polohy. Frekvence L2 (1 227,60 MHz) obsahuje jen P-kód, takže je využívána pouze jen speciálně vybavenými přijímači. Obě nosné frekvence přenáší ještě binární kód s navigační zprávou.

Aby mohl přijímač správně spočítat svoji pozici, potřebuje znát přesnou pozici družic, od kterých měří vzdálenost. Potřebné informace obsahuje navigační zpráva, pomocí níž přijímač spočítá přesnou pozici a čas odeslání dálkoměrného kódu.

Navigační zpráva obsahuje různé údaje. Například informaci o stavu družice nebo almanach, který obsahuje méně přesné parametry oběžných drah všech družic v kosmickém segmentu pro rychlejší vyhledání pozic družic. Přesné pozice družic se nazývají efemeridy, pomocí nichž přijímač dokáže určit polohu družice. Data obsažená v navigační zprávě jsou obvykle platná 4 hodiny, odeslání kompletní navigační zprávy trvá 12,5 minut.

Tím, že je GPS vojenský GNSS, může ministerstvo obrany USA kdykoli znovu spustit selektivní dostupnost, což znamená, že do kódu se zanesou umělé chyby, která výrazně zhorší přesnost. Popřípadě mohou vypnout nešifrovaný signál a GPS bude v civilní sféře nepoužitelná.

### 3.6.6 Diferenciální korekce

Systém GPS se dopouští chyby. Chyba je dána nepřesností v určení klíčových parametrů satelitů a v nepřesnosti měření doby šíření signálu od satelitu směrem k přijímači [45, str. 78]. Pro potlačení zmíněné chyby se používá diferenciální korekce. Pokud GPS využívá diferenciální korekci, tak se označuje jako DGPS.

Aby bylo možné využít diferenciální korekce, je nutné použít dva přijímače. Přijímač, který potřebuje korekci získaných souřadnic, je pohyblivý a referenční přijímač, u kterého je známa jeho geografická poloha, je stacionární [45, str. 79]. Referenční přijímač dokáže zjistit aktuální chybu porovnáním své skutečné a určené polohy. Pokud se chyba aplikuje na pohyblivý přijímač, je možné dosáhnout výrazně lepší přesnosti.

Diferenciální korekci je možné provádět dvěma způsoby. Jako real-time, kdy je časovaný záznam GPS korekce vysílán pohyblivému přijímači, nebo je možné korekci aplikovat v rámci post-processingu.

## 3.7 Open Geospatial Consortium

Dokument [8] definuje Open Geospatial Consortium (OGC) jako mezinárodní standardizační organizaci založenou na dobrovolné shodě. OGC vznikla v roce 1994 a zahrnuje více než 400 komerčních, vládních a nevýdělečných organizací za účelem spolupráce na procesu otevřené shody podporující vývoj a implementaci standardů pro geoprostorová data

a služby, GIS, zpracování dat a jejich výměnu. OGS specifikovala více jak 30 standardů, z nichž některé souvisí s diplomovou prací.

## 3.8 Open source knihovny používané pro GIS

Pro implementaci GIS nástrojů je velkou výhodou, že existují kvalitní open source knihovny, které zjednodušují práci s geografickými daty. Problematika GIS zahrnuje velké množství formátů a matematika spojená se zpracováním geografických informací není triviální, takže bez dobrých knihoven by byl vývoj v této oblasti mnohem náročnější.

### 3.8.1 Proj.4

Proj.4 je původně vytvořená Geraldem Evendenem v United States Geological Survey, což je americká nezávislá vědeckovýzkumná vládní agentura [24]. Dnes je Proj.4 open source knihovna, která je jedním z projektů nadace Open Source Geospatial Foundation. Je distribuována pod licencí MIT [24].

Proj.4 je psaná v jazyce C. Je využívána většinou open source GIS nástrojů pro přepočítání mezi geografickými souřadnicemi šířka-délka a kartézskými souřadnicemi mapy vzniklé některým mapovým zobrazením (kapitola 3.3)[24]. Kromě toho umí Proj.4 i konverzi souřadnic mezi náhradními elipsoidy (kapitola 3.2) a souřadnými systémy (kapitola 3.4). Proj.4 dokáže také vypočítat vzdálenost a azimut mezi dvěma body, které jsou zadány souřadnicemi šířka-délka. Knihovna Proj.4 je součástí SpatiaLite (kapitola 3.8.5), které je použito v implementovaném programu popisovaném v diplomové práci pro uložení dat.

### 3.8.2 Java Map Projection Library

Java Map Projection Library je open source knihovna psaná v jazyce Java [22]. Je distribuována pod licencí Apache License. Implementuje podmnožinu mapových zobrazení (kapitola 3.3) z knihovny Proj.4 (kapitola 3.8.1). Oproti knihovně Proj.4 neumí konverzi a měření vzdálenosti mezi body a neobsahuje všechna zobrazení. Knihovna byla původně použita v implementovaném programu v rámci diplomové práce, ale po použití SpatiaLite (kapitola 3.8.5) pro uložení dat se ukázala jako zbytečná.

### 3.8.3 JTS Topology Suite

JTS Topology Suite je open source knihovna psaná v jazyce Java. Je distribuována pod licencí LGPL. Knihovna poskytuje objektový model pro dvojrozměrnou eukleidovskou geometrii společně s některými důležitými geometrickými funkcemi [7]. Knihovna je primárně zaměřená na to, aby byla využita jako základ pro vektorové vyjádření objektů v GIS nástrojích.

JTS implementuje geometrický model a API definované Open Geospatial Consortium (kapitola 3.7) ve standardu Simple Features pro SQL [7]. Obsahuje třídy pro modelování bodů, linií a polygonů, které jsou tvořeny vrcholy. Z geometrických operací, které podporuje, můžeme jmenovat například kontrolu topologické správnosti, vzdálenostní a plošné funkce, průnik, rozdíl, sjednocení, doplněk, konvexní obálka, minimální ohraničující obdélník a spoustu dalších [7]. Knihovna JTS Topology Suite byla použita při implementaci programu popisovaného v diplomové práci.

### 3.8.4 Geos

Knihovna Geos je distribuovaná pod licenci LGPL. V dokumentu [32] je popsána její historie. V roce 2003 během vývoje rozšíření PostGIS pro PostgreSQL, které přidalo do relační databáze funkcionalitu pro uložení prostorových dat, se ukázalo, že funkce implementované v JTS Topology Suite (kapitola 3.8.3) by byly pro PostGIS velice přínosné. Problémem bylo, že PostGIS vyžadovalo implementaci v jazyce C/C++ a JTS Topology Suite bylo implementováno v jazyce Java. Z toho důvodu vznikla knihovna Geos, která je C++ portem knihovny JTS Topology Suite. Na portování knihovny se podílel i autor JTS Topology Suite. Knihovna Geos je součástí SpatiaLite (kapitola 3.8.5), která je použita v programu popisovaném v diplomové práci.

### 3.8.5 SpatiaLite

Knihovna SpatiaLite je použita v implementovaném programu v rámci diplomové práce pro uložení a načítání dat. Je rozšířením knihovny SQLite. V dokumentu [30] jsou popsány vlastnosti SQLite. SQLite je open source knihovna psaná v jazyce C, která implementuje jednoduchý relační databázový systém. Nemá žádné další závislosti a je mutliplatformní. SQL databázové systémy jsou obvykle založeny na principu klient-server. SQLite je výjimka, protože se jedná pouze o malou knihovnu, která se přilinkuje k aplikaci a je možné ji používat přes jednoduché API ihned bez další konfigurace. SQLite implementuje většinu z SQL92. Databáze je uložena pouze v jediném souboru. SQLite i SpatiaLite jsou distribuovány pod třemi licencemi MPL, GPL a LGPL, ze kterých si uživatel může vybrat.

Knihovna SpatiaLite je popsána v dokumentu [28]. SpatiaLite umožňuje uložení, načítání a zpracování geografických dat nad databází SQLite. SpatiaLite vyžaduje tři další knihovny: knihovnu Geos (kapitola 3.8.4) pro prostorovou analýzu, knihovnu Proj.4 (kapitola 3.8.1) pro přepočty mezi souřadnými systémy a knihovnu libiconv pro konverzi mezi znakovými sadami.

Manuál k používání SpatiaLite se nachází v dokumentu [29]. Veškerá manipulace s daty se provádí výhradně přes SQL, které se spouští stejným způsobem jak v čistém SQLite. Pro využití databázového souboru pro uložení prostorových dat je nutné ho inicializovat následujícím příkazem:

```
SELECT InitSpatialMetaData()
```

Při inicializaci se vytvoří dvě tabulky s metadaty. První tabulkou je `spatial_ref_sys`, ve které je uložena celá EPSG databáze (kapitola 3.5), která obsahuje 3722 záznamů. Každý záznam obsahuje EPSG kód, název souřadného systému, parametry pro knihovnu Proj.4 a definici souřadného systému ve formátu Well-known text, který je definovaný organizací OGC (kapitola 3.7).

Druhou tabulkou je `geometry_columns`. Každý její záznam definuje sloupec některé tabulky, který obsahuje geometrii uložených geoobjektů. Záznamy obsahují informaci o názvu tabulky, názvu sloupce, typu a dimenzích geoobjektů, identifikátor souřadného systému a indikátor, jestli je sloupec indexovaný.

Každá vektorová vrstva je uložena formou tabulky. Sloupec tabulky může být v některém z klasických SQLite datových typů, kterými jsou [27]:

- **INTEGER** - celé číslo,
- **REAL** - reálné číslo uložené na 8 bytech,

- TEXT - text ve znakové sadě UTF-8, UTF-16BE nebo UTF-16LE,
- BLOB - binární data.

Klasické sloupce obsahují neprostorové údaje. Aby tabulka mohla obsahovat i prostorová data, musí být jeden nebo více sloupců definováno v tabulce `geometry_columns`. Geoobjekty mohou být jedním z následujících typů:

- POINT - bodový geoobjekt,
- LINESTRING - liniový geoobjekt,
- POLYGON - polygonový geoobjekt,
- MULTIPOINT - geoobjekt složený z více bodů,
- MULTILINESTRING - geoobjekt složený z více linií,
- MULTIPOLYGON - geoobjekt složený z více polygonů,
- GEOMETRYCOLLECTION - geoobjekt složený z prvních třech typů (v GIS se prakticky nepoužívá).

SpatiaLite zná ještě další typy, které jsou odvozené od zmíněných typů. Ty navíc obsahují i třetí souřadnici Z nebo měřítko. Tyto rozšířené typy se moc nepoužívají.

Vytvoření sloupce pro uložení geografické informace se dělá následujícím SQL příkazem:

```
SELECT AddGeometryColumn(table String, column String, srid Integer,
    geom_type String[, dimension String[, not_null Integer]]) : Integer
```

Jednotlivé argumenty mají následující význam: název tabulky, název sloupce, identifikátor souřadného systému z tabulky `spatial_ref_sys`, typ geoobjektů, dimenze (obvykle 'XY'), indikátor, jestli je zakázáno uložit místo geometrie NULL. Funkce vrátí 1, pokud se podařilo sloupec vytvořit, jinak 0.

Popisovaná funkce vytvoří jednak sloupec v tabulce vrstvy, ale navíc přidá potřebný záznam do tabulky `geometry_columns` a přidá potřebné triggery. Vytvořený sloupec je možné smazat příkazem:

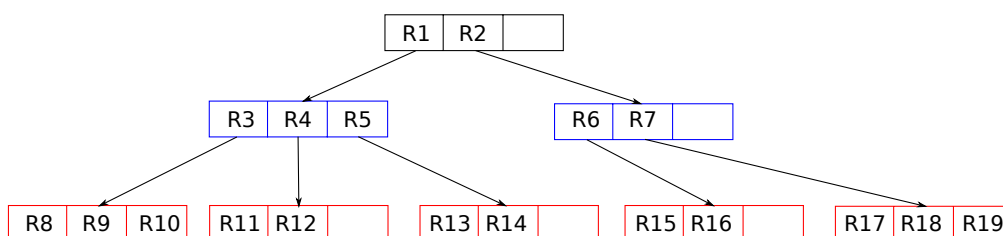
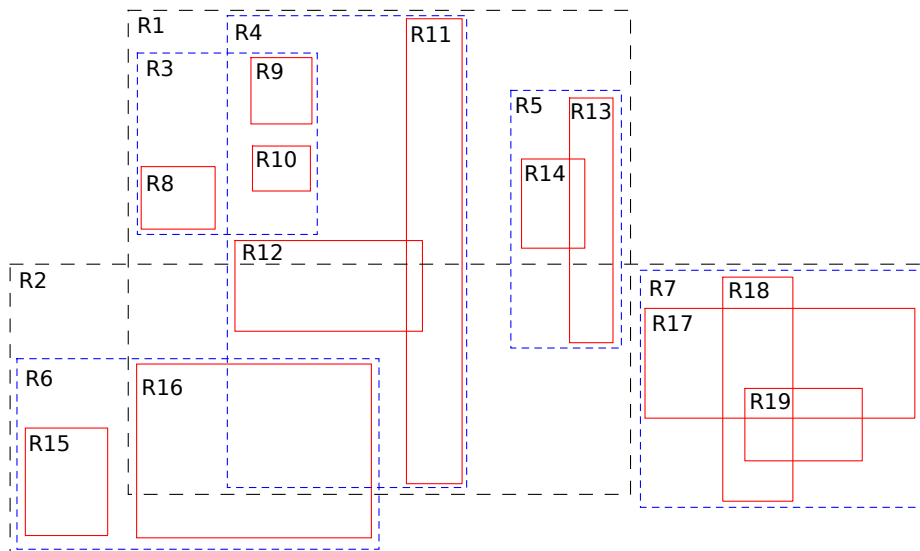
```
SELECT DiscardGeometryColumn(table String, column String) : Integer
```

Zmíněná funkce zruší všechna metadata a triggery, ale v tabulce vrstvy sloupec zůstane.

Data uložená ve SpatiaLite je možné indexovat pomocí R-tree. Jedná se o stromovou datovou strukturu, která se používá pro prostorové přístupové metody. Na obrázku 3.4 je vidět schéma jednoduchého R-stromu. Metoda je založena na tak zvaných MBR (minimum bounding rectangles - minimální ohraničující obdélníky) geoobjektů. Každá úroveň stromu obsahuje MBR svých potomků, takže se zrychlí nalezení objektů nacházejících se v určité oblasti. SpatiaLite využívá variantu R\*Tree. Index je uložený v tabulce, jejíž jméno je složeno z názvu tabulky a názvu sloupce - `idx_tabulka_sloupec`.

Pro vytvoření a zrušení indexu slouží následující SQL příkazy:

```
SELECT CreateSpatialIndex(table String, column String) : Integer
SELECT DisableSpatialIndex(table String, column String) : Integer
```



Obrázek 3.4: Schéma R-stromu na jednoduchém příkladu. (převzato z [6])

Pro získání geoobjektů z tabulky lavičky a sloupce `geometry` v oblasti  $49,17^\circ - 49,18^\circ$  severní šířky a  $16,6^\circ - 16,62^\circ$  východní délky pomocí indexu by vypadal příkaz následovně:

```
SELECT AsText(geometry) FROM "lavičky"
WHERE ROWID IN (SELECT pkid FROM "idx_lavičky_geometry"
WHERE pkid MATCH RTreeIntersects(16.6, 49.17, 16.62, 49.18))
```

Další důležitou funkcí, kterou SpatiaLite obsahuje, je funkce `Transform`, která slouží k převodu mezi geografickými souřadnými systémy. Funkce používá knihovnu Proj.4 (kapitola 3.8.1). Převod souřadnice  $49,2261589^\circ$  severní šířky,  $16,5966067^\circ$  východní délky v geografickém souřadném systému šířka-délka (kapitola 3.4.1) a použitým náhradním elipsoidem WGS-84 (EPSG kód 4326) do souřadného systému WGS-84 / UTM zóna 33N (EPSG kód 32633) lze provést následujícím příkazem:

```
SELECT AsText(Transform(MakePoint(16.5966067, 49.2261589, 4326), 32633))
```

### 3.9 Formát Shapefile

Datový formát Shapefile vytvořila firma Esri. Slouží k ukládání vektorových prostorových dat. Shapefile se stal velice oblíbeným formátem pro přenos dat mezi aplikacemi. Program

implementovaný v rámci diplomové práce provádí import a export do formátu Shapefile. Specifikace formátu je popsána v dokumentu [1].

Shapefile neumí uložit topologickou informaci. Výhodou je jednoduchost a rychlost načítání a zápisu dat. Formát Shapefile je složený z několika souborů, které musí mít stejný název a různou příponu. Musí povinně obsahovat tři soubory. Hlavní s příponou shp, indexový s příponou shx a databázový s příponou dbf. Název by měl splňovat konvence pojmenování souborů v systému MS DOS, avšak dnešní programy zvládají i názvy bez zmíněného omezení.

Kromě třech povinných souborů může obsahovat Shapefile ještě další soubory. Asi nejdůležitější je definice geografického souřadného systému. Pro definici souřadného systému mohou být použity dva soubory. Oba soubory obsahují definici ve formátu Well Known Text, který definovala organizace OGC (kapitola 3.7), avšak v definici se liší. Esri má vlastní definici, která se nachází v souboru s příponou prj. Definice souřadného systému podle OGC se nachází v souboru s příponou qpj. Dalším praktickým souborem je soubor s příponou cpg, který obsahuje znakovou sadu atributů. Shapefile může obsahovat další soubory, kterými jsou většinou různé indexy.

### 3.9.1 Soubor SHP

Hlavní soubor s příponou shp je binární a obsahuje popis geometrie modelovaných geoobjektů. Umožňuje uložení bodové, liniové a polygonové geometrie. Soubor začíná hlavičkou, která má fixní délku 100 bytů. Hlavička má strukturu popsanou v tabulce 3.1 [1].

V tabulce 3.2 se nachází typy geoobjektů. Typy, které končí písmenem M, obsahují pro každý vrchol navíc hodnotu míra (anglicky measure), u které si uživatel může zvolit, k čemu ji využije. Typy, které končí písmenem Z, jsou třídímní a mohou obsahovat i míru.

Za hlavičkou souboru následují jednotlivé záznamy. Každý záznam obsahuje vlastní hlavičku, která má fixní délku 8 bytů. Tabulka 3.3 popisuje strukturu hlavičky.

Za hlavičkou záznamu následuje jeho obsah. První 4 byty definují typ záznamu podle tabulky 3.2 a za ním následují byty, které jsou různé podle typu záznamu. Tabulka 3.4 zobrazuje obsah záznamu bodového geoobjektu. Tabulka 3.5 zobrazuje obsah záznamu multibodového geoobjektu. Typem point jsou myšleny souřadnice X a Y, které jsou typu  $2 \times \text{double}$ . Tabulka 3.6 zobrazuje obsah záznamu liniového nebo polygonového geoobjektu. Zmíněné dva typy mohou být rozděleny do více částí, proto obsahují navíc pole indexů začátků jednotlivých částí. Jednotlivé části polygonu musí mít spojený první a poslední bod. Tedy musí mít totožnou souřadnici. Ostatní typy geoobjektů vychází z popsané struktury, akorát pokud obsahují Z nebo M, je přidána minimální hodnota, maximální hodnota a pole všech hodnot, které má délku podle počtu bodů.

### 3.9.2 Soubor SHX

Soubor s příponou shx obsahuje index geoobjektů uložených v hlavním souboru s příponou shp. Každý záznam obsahuje offset a délku každého záznamu v hlavním souboru pro rychlé hledání vpřed i vzad. Struktura hlavičky indexového souboru je totožná s hlavičkou hlavního souboru (tabulka 3.1). Struktura záznamu indexového souboru má fixní velikost 8 bytů a je znázorněna v tabulce 3.7.

Byty	Typ	Pořadí bytů	Popis
0-3	integer	big-endian	kód souboru - hodnota 9994
4-23	5× integer	big-endian	nepoužité
24-27	integer	big-endian	délka souboru včetně hlavičky v 16-bitových slovech
28-31	integer	little-endian	verze - hodnota 1000
23-35	integer	little-endian	typ geobjektů viz. tabulka 3.2
36-43	double	little-endian	minimální hodnota X
44-51	double	little-endian	minimální hodnota Y
52-59	double	little-endian	maximální hodnota X
60-67	double	little-endian	maximální hodnota Y
68-75	double	little-endian	minimální hodnota Z (pokud má Z, jinak 0.0)
76-83	double	little-endian	maximální hodnota Z (pokud má Z, jinak 0.0)
84-91	double	little-endian	minimální hodnota M (pokud má M, jinak 0.0)
92-99	double	little-endian	maximální hodnota M (pokud má M, jinak 0.0)

Tabulka 3.1: Popis hlavičky hlavního souboru shp.

Hodnota	Typ
0	Null Shape
1	Point - bod
3	PolyLine - linie
5	Polygon - polygon
8	MultiPoint - objekt tvořen více body
11	PointZ
13	PolyLineZ
15	PolygonZ
18	MultiPointZ
21	PointM
23	PolyLineM
25	PolygonM
28	MultiPointM
31	MultiPatch - povrch složený z částí (příliš se nepoužívá)

Tabulka 3.2: Typy geobjektů.

Byty	Typ	Pořadí bytů	Popis
0-3	integer	big-endian	číslo záznamu
4-7	integer	big-endian	velikost záznamu v 16-bit. slovech bez hlavičky

Tabulka 3.3: Popis hlavičky jednotlivých záznamů.

Byty	Typ	Pořadí bytů	Popis
0-3	integer	little-endian	typ záznamu - hodnota 1
4-11	double	little-endian	souřadnice X
12-19	double	little-endian	souřadnice Y

Tabulka 3.4: Obsah záznamu pro bodový objekt.

Byty	Typ	Pořadí bytů	Popis
0-3	integer	little-endian	typ záznamu - hodnota 8
4-35	4× double	little-endian	MBR (min. X, min. Y, max. X, max. Y)
36-39	integer	little-endian	počet bodů
40-N	počet bodů × point	little-endian	pole bodů

Tabulka 3.5: Obsah záznamu pro multibodový objekt.

Byty	Typ	Pořadí bytů	Popis
0-3	integer	little-endian	typ záznamu - hodnota 3 nebo 5
4-35	4× double	little-endian	MBR (min. X, min. Y, max. X, max. Y)
36-39	integer	little-endian	počet částí
40-43	integer	little-endian	počet bodů
44-N	počet částí × integer	little-endian	pole indexů částí
N-NN	počet bodů × point	little-endian	pole bodů

Tabulka 3.6: Obsah záznamu pro liniový a polygonový objekt.

Byty	Typ	Pořadí bytů	Popis
0-3	integer	big-endian	ofset záznamu v 16-bit. slovech
4-7	integer	big-endian	velikost záznamu v 16-bit. slovech bez hlavičky

Tabulka 3.7: Popis záznamu indexového souboru.

### 3.9.3 Soubor DBF

Soubor s příponou dbf obsahuje atributy geoobjektů. Jedná se o formát dBase. Databáze dBase byla jedním z prvních více rozšířených systémů řízení báze dat. Publikována byla americkou firmou Ashton-Tate v roce 1980 [9].

Pro atributovou tabulku uloženou v souboru dbf musí platit, že jeden její záznam koresponduje s jednou geometrií uloženou v hlavním souboru. Pořadí záznamů musí být stejné jako pořadí geometrií.

Formát souboru dbf je popsán v dokumentu [43]. Soubor dbf je binární. Na začátku souboru se nachází hlavička. Prvních 32 bytů je fixních (tabulka 3.8) a za nimi se nachází definice jednotlivých sloupců (tabulka 3.9), kde definice každého sloupce je na 32 bytů. Posledním bytem hlavičky je ukončovač, který má hodnotu 0Dh. Z toho vyplývá, že hlavička je dlouhá  $32 + 32 \times \text{počet sloupců} + 1$  bytů.

Každý záznam má na začátku 1 byte jako příznak smazání, který je nastaven buď na hodnotu 20h, což znamená, že záznam je validní, nebo 2Ah, která znamená, že záznam je smazaný. Následují jednotlivé hodnoty, které jsou dlouhé podle definice sloupce v hlavičce. Byty, které nejsou nastaveny na platný znak, obsahuje ASCII hodnotu mezery 20h.

## 3.10 Webové mapové služby

Pro diplomovou práci jsou webové mapové služby alternativou, kterou by bylo možné využít jako podkladovou mapu. Nakonec jsem využil místo WMS (Web Map Service) OpenStreetMap ve formě dlaždic 3.11.

Pro sdílení geografických dat v distribuovaném prostředí internetu vytvořila organizace OGC (kapitola 3.7) otevřený formát WMS, který se stal jedním z jejich nejpopulárnějších standardů. WMS je popsáno v dokumentu [46]. WMS poskytují pouze rastrová data. Funguje tak, že klient pomocí protokolu HTTP metodou GET nebo POST zažádá server o určitou vrstvu a oblast v určitém souřadném systému, a server vrátí rastr, který pokrývá žádanou oblast.

Ve standardu WMS jsou definovány dva povinné dotazy na server. Na dotaz `GetCapabilities` server vrátí informace ve formátu XML o službách, které poskytuje, například typ rastrového obrázku, vrstvy, které zpřístupňuje, podporované souřadné systémy (kapitola 3.4) nebo oblast, kterou pokrývá. Druhým dotazem je `GetMap`, při kterém server pošle žádanou mapu. V rámci dotazu se specifikuje požadovaný rastrový formát, souřadnicový box, souřadný systém a další.

Dalším dotazem, který je ve specifikaci, ale je volitelný, je `GetFeatureInfo`. Na dotaz server pošle XML soubor s atributy určitého prvku na souřadnicích předaných s dotazem.

Pro OpenStreetMap data (kapitola 3.11) existují jak servery poskytující dlaždice, tak WMS servery. Nevýhodou WMS oproti dlaždicím je výrazně větší zátěž serveru. WMS je sofistikovanější, ale pro každý dotaz `GetMap` musí server vytvořit rastr, který má požadované vlastnosti. Oproti tomu dlaždice jsou na serveru fyzicky uloženy.

## 3.11 Internetové dlaždicové mapy

Data z internetové dlaždicové mapy OpenStreetMap byla v diplomové práci použita jako podkladová mapová vrstva.

Na internetu existují mapy, které poskytují geografická data pro různé služby. Z nejznámějších to jsou OpenStreetMap, Google Maps nebo Bing Maps. Google Maps je služba

Byty	Popis
0	verze
1	rok - 1900
2	měsíc
3	den
4-7	počet záznamů (32 bitů)
8-9	velikost hlavičky (16 bitů)
10-11	velikost záznamu (16 bitů)
12-13	nepoužito
14	detekce nedokončené transakce
15	detekce šifrování
16-19	rezervované pro LAN
20-27	rezervované pro víceuživatelské dBase
28	flag MDX indexového souboru
29	znaková sada (DOS / Windows Code Page)
30-31	nepoužité

Tabulka 3.8: Fixní část hlavičky dbf souboru.

Byty	Popis
0-10	jméno sloupce v ASCII ukončené hodnotou 00h
11	datový typ sloupce (viz. tabulka 3.10)
12-15	adresa v paměti (pro Shapefile hodnota 0)
16	maximální velikost hodnoty sloupce
17-31	pro Shapefile se nastavuje hodnota 0

Tabulka 3.9: Definice sloupce tabulky.

Kód	Typ	Popis
C	Character	text dlouhý maximální 254 znaků
N	Number	číslo uložené jako text dlouhý maximálně 18 znaků
L	Logical	boolean hodnota, velikost 1 znak (true - Y,y,T,t; false - N,n,F,f)
D	Date	datum uložené na 8 znaků ve formátu RRRRMMDD

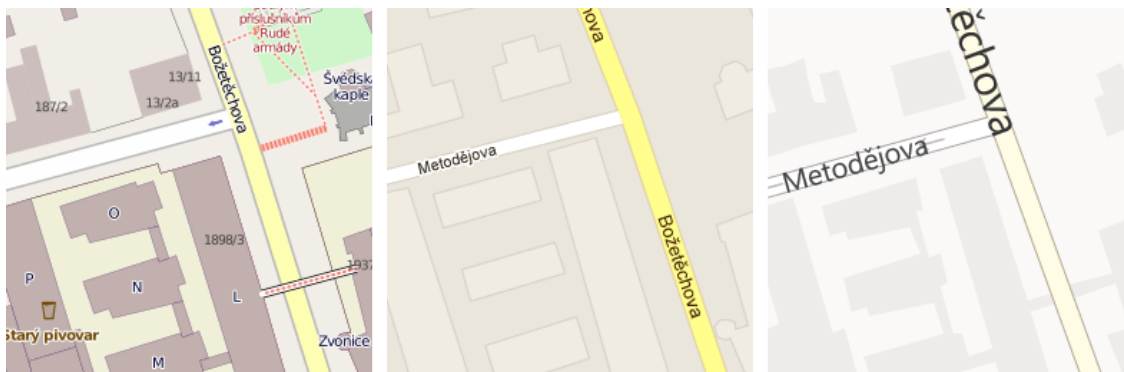
Tabulka 3.10: Některé datové typy sloupce.

poskytovaná firmou Google a Bing Maps firmou Microsoft. OpenStreetMap vznikají po vzoru Wikipedie jako svobodná databáze geografických dat, kterou vytváří sami uživatelé [11]. Internetové mapy jsou často používány jako podkladová mapa pro nástroje, které pracují s geografickými daty.

### 3.11.1 Formát dat

Všechny tři výše zmíněné internetové mapy (kapitola 3.11) používají stejný formát uložení dat. Geografická data jsou zobrazena mapovou projekcí Spherical Mercator Projection (kapitola 3.11.2)[15]. Díky tomu je výsledná mapa čtvercová.

Geografická data jsou renderovaná pro několik úrovní přiblížení. Každá úroveň se skládá z dlaždic. Dlaždice mají rozměr 256x256 px [10]. Úrovně se číslují celými čísly. Úroveň 0 je tvořena jedinou dlaždicí, která obsahuje celé zobrazované území. Strana čtverce mapy zobrazeného v každé úrovni je dvakrát větší než v úrovni o jednu menší. Z toho vyplývá, že každá úroveň obsahuje  $2^{2z}$  dlaždic, kde  $z$  je úroveň přiblížení [12]. Každá dlaždice má souřadnici  $x, y$ . Dlaždice 0, 0 se nachází v levém horním rohu vrstvy. Na obrázku 3.5 je vidět dlaždice 143 156, 89 773 pro úroveň přiblížení 18 ve službách OpenStreetMap, Google Maps a Bing Maps.



Obrázek 3.5: Porovnání stejné dlaždice pro OpenStreetMap, Google a Bing Maps.

Uložení dat jako dlaždice má hlavní výhodu v rychlosti. Server má uloženo velké množství rastrů a pouze je odesílá. OpenStreetMap zobrazuje celý svět v úrovních přiblížení 0-18. Úroveň 18 obsahuje 68 719 476 736 dlaždic. Většinu těchto dlaždic si nikdo nikdy nezobrazí, a proto část dlaždic je uložena a část se renderuje pouze, když jsou potřeba [13]. 6. ledna 2012 zabíraly dlaždice 1272 GB.

### 3.11.2 Spherical Mercator Projection

Spherical Mercator Projection je pojem, který se vžil pro zobrazení, které používají internetové mapy jako OpenStreetMap, Google Maps nebo Bing Maps (kapitola 3.11). Pro toto zobrazení dříve neexistoval oficiální EPSG kód (kapitola 3.5). Z počátku se organizace OGP, která spravuje EPSG databázi, vyjádřila, že ho považuje za technicky chybné a nebude pro něj vytvářet specifikaci [41]. Až po několika letech pro něj vytvořila specifikaci pod kódem 3785. Stalo se tak v polovině roku 2008. Půl roku na to byl kód 3785 nahrazen kódem 3857, který se využívá i dnes.

Pro Mercatorovo zobrazení se používá náhradní koule, která má poloměr 6 378 137 m [41]. Poloměr koule je tedy stejný jako délka hlavní poloosy elipsoidu WGS-84[16]. Důležité

je, že každý bod na náhradní kouli má stejnou zeměpisnou šířku a délku jako ekvivalentní bod na náhradním elipsoidu WGS-84. Zobrazení je definované pro území mezi  $180^\circ$  západní délky až  $180^\circ$  východní délky a  $85,0511^\circ$  jižní šířky až  $85,0511^\circ$  severní šířky. Zobrazené území má čtvercový tvar.

## Kapitola 4

# Návrh grafického uživatelského rozhraní

Grafické uživatelské rozhraní je pro implementovaný program velice důležité. U mobilních aplikací je na něj kladen větší důraz než u desktopových, protože mobilní aplikace musí být intuitivní a přizpůsobené ovládní prsty. Pokud tomu tak není, obvykle ji uživatelé, nepřijmou i když poskytuje sebelepší funkčnost.

### 4.1 Principy mobilního uživatelského rozhraní

Mezi způsobem ovládní mobilního zařízení a stolního počítače jsou výrazné rozdíly. Uživatelské rozhraní stolního počítače bývá koncipováno tak, aby se každý program otevřel v jednom nebo více oknech. Protože monitor stolního počítače má obvykle velké rozlišení, může program zobrazovat kromě hlavní plochy také různé panely nebo nástrojové lišty. Program se obvykle ovládá převážně myší a pro zrychlení častých operací bývají definovány klávesové zkratky.

Naproti tomu u mobilního zařízení jsme často omezeni velikostí displeje. Program běží přes celou obrazovku a hlavní plochu se snažíme co nejméně zmenšovat jakýmkoli dalšími prvky, které nejsou nezbytně nutné. Ovládní se u mobilních zařízení postupem času vyvíjí. Před příchodem kapacitních displejů býval k mobilnímu zařízení stylus a zařízení se ovládalo převážně jím [48]. Často bylo ovládní koncipováno velice podobně jako pro stolní počítač, akorát všechny ovládací prvky byly menší. Z toho důvodu se zařízení špatně ovládalo prsty a převážně se využíval stylus s tenkým hrotem.

V roce 2007 firma Apple vydala iPhone s konceptem ovládní mobilních zařízení, které postupem času přejala většina ostatních výrobců. Díky kapacitnímu displeji se zařízení ovládá prsty. To znamená, že všechna tlačítka musí být větší, protože prstem nelze kliknout tak přesně jako stylusem.

Kapacitní displej umožňuje multi-touch a různá gesta [48]. Díky tomu vznikl nový koncept ovládní. Už nejde jen o kliknutí na tlačítko, ale mobilní zařízení rozpoznává například dlouhý stisk, dvojklik nebo posun po displeji jedním nebo více prsty [18]. Speciální gesto používané hlavně pro zoom se nazývá pinch. Jde o gesto prováděné naráz dvěma prsty, které se navzájem přibližují nebo oddalují.

V posledních letech dochází k redukci hardwarových tlačítek. Dříve bylo více mobilních zařízení s hardwarovou klávesnicí. Dnes se většina mobilních zařízení ovládá skoro výhradně přes displej, který vyplňuje co největší plochu mobilního zařízení. Na obrázku 4.1 je vidět



Obrázek 4.1: Sony Ericsson Xperia X1 a HTC Sensation XE. (převzato z [6])

rozdíl mezi Sony Ericsson Xperia X1 z roku 2008 a HTC Sensation XE z roku 2011. Xperia má kromě hardwarové klávesnice pod displejem devět hardwarových tlačítek. Oproti tomu smartphone od HTC má displej, který zabírá většinu jeho přední plochy, a pod ním jsou čtyři malá dotyková tlačítka.

Mobilní uživatelské rozhraní by mělo být jednoduché a intuitivní. Často je lepší dát přednost vhodné ikoně než textu [17]. Některé věci, na které jsme zvyklí na stolním počítači, se u mobilních aplikací dělají jinak. Například místo tlačítka save se obvykle ukládá aktuální stav aplikace bez vědomí uživatele. Se souborovým systémem by měl uživatel pracovat, jen pokud je to nutné. Program by neměl ani na chvíli přestat reagovat při výpočtu, jak jsme na to zvyklí z desktopových aplikací. Je vhodné pro ovládání využívat gesta a animace, které vytváří dojem, že objekty se chovají stejně jako objekty v reálném světě. Takové chování zvyšuje intuitivnost ovládání.

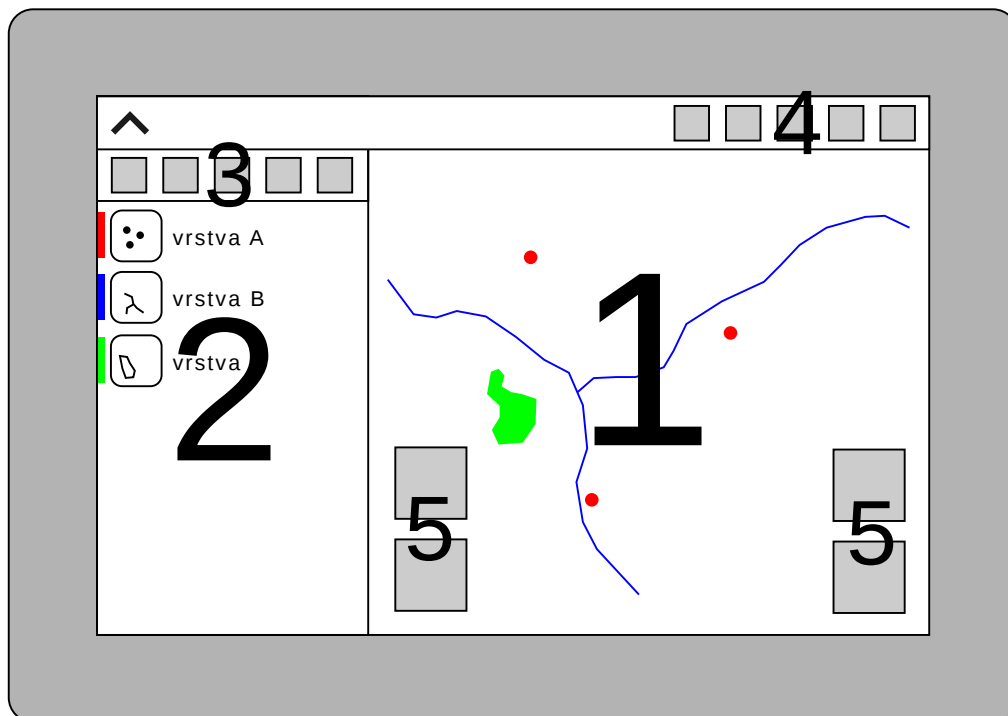
V roce 2010 vydala firma Apple tablet iPad, který se stal velice úspěšným. Podle vzoru iPadu vznikly další tablety s jinými operačními systémy. Tablet je mobilní zařízení, které se ovládá velice podobným způsobem jako dnešní smartphony. Obvykle také obsahuje stejný operační systém. Hlavní rozdíl je v tom, že obsahuje výrazně větší displej, takže používání je pohodlnější.

## 4.2 Prvky GUI implementovaného programu

Při návrhu GUI programu bylo potřeba brát v potaz několik různých aspektů. Primárně by způsob používání měl vycházet ze zvyklostí v programech typu GIS. Na druhou stranu bylo nutné ovládání navrhnout tak, aby program bylo možné ovládat prsty. Další věc je, že

implementovanou aplikaci by mělo být možné používat jak na tabletu, tak na smartphone s výrazně menším displejem. Ovládací prvky by měly co nejméně zabírat prostor pro okno s mapovými vrstvami.

Na obrázku 4.2 je vidět navržené schéma grafického uživatelského prostředí. V následujících kapitolách budou popsány jednotlivé části GUI. Pro lepší orientaci jsem do schématu přidal čísla, na která se budu odkazovat.



Obrázek 4.2: Schéma grafického uživatelského prostředí.

### 4.2.1 Plátno pro vykreslení mapových vrstev

Pod číslem 1 na obrázku 4.2 se nachází plátno pro vykreslení mapových vrstev. Posun a zoom v plátně se dělá typicky pro mobilní aplikace. Položením prstu na displej a jeho posunem se provádí posun plátna. Změna měřítka mapových vrstev se dělá pomocí gesta pinch (kapitola 4.1).

Během návrhu se ukázalo, že by bylo výhodné oddělit editaci od prohlížení geografických dat. Z toho důvodu má aplikace dva módy, ve kterých se může nacházet mód prohlížení a mód editace. Módy se přepínají pomocí tlačítek v horní nástrojové liště pod číslem 4 ve schématu na obrázku 4.2 (kapitola 4.2.6).

V prohlížečím módu je možné kliknutím zvolit objekt vybrané vrstvy (kapitola 4.2.3). Zvolený objekt se zvýrazní a zároveň se vybere korespondující řádek v atributové tabulce (kapitola 4.2.5).

V editačním módu se kliknutím na objekt vybrané vrstvy umožní jeho editace. Editovaný objekt se vykreslí v jiném stylu než ostatní. Zobrazí se vrcholy, výplně se ukáží poloprůhledně a linie přerušovaně. Pokud je editovaný objekt linie nebo polygon, tak je možné kliknutím vybrat jeden vrchol. Bodový objekt nebo vybraný vrchol lze posunout

táhnutím prstu na jiné místo. Editace objektů se provádí převážně editačními tlačítky (kapitola 4.2.7).

Při známé poloze z lokalizačních služeb je vykreslen na dané pozici černý křížek a souřadnice zobrazena v pravém spodním rohu plátna. Podobně pro manuální výběr souřadnice se v plátně vykreslí červený křížek a červené souřadnice.

Manuální výběr souřadnice je možné zapnout pouze při editačním módu. Pokud je zapnutý, tak kliknutím do plátna si program zapamatuje souřadnici a vykreslí na ni červený křížek. Do vybrané pozice lze posunout existující vrchol nebo vytvořit nový pomocí editačních tlačítek (kapitola 4.2.7). V programu je ještě druhý typ manuálního výběru souřadnice, ve kterém je možné vybírat pouze souřadnice existujících vrcholů všech vrstev. Tento typ je výhodné použít pro zachování topologie geoobjektů. Umožní napojit geoobjekt přímo do jiného, takže vrcholy v místě napojení mají totožnou souřadnici.

## 4.2.2 Postranní panel

Pod číslem 2 na obrázku 4.2 se nachází postranní panel, ve kterém se zobrazí buď seznam mapových vrstev (kapitola 4.2.3), nebo atributová tabulka (kapitola 4.2.5) vybrané vrstvy. Panel je možné podle potřeby kdykoli skrýt nebo zobrazit. Pokud běží aplikace na displeji, který má malou šířkou, tak ho při zobrazení zakryje celý.

Panel je možné zobrazit kliknutím při levém okraji plátna. Skrýt panel je možné kliknutím na ikonu, která se nachází úplně vlevo v panelu nástrojů pod číslem 3 na obrázku 4.2. Jedná se o ikonu 1 na obrázku 4.3.

Seznam vrstev a atributovou tabulku by bylo možné vložit na samostatnou obrazovku, ale takové řešení je pro tablety nepraktické.

## 4.2.3 Seznam mapových vrstev

Seznam mapových vrstev obsahuje všechny načtené mapové vrstvy. Do seznamu bylo třeba vložit vizuální informace o vrstvách. Každá vrstva má u sebe barevný obdélník, který identifikuje barvu, kterou je vykreslena v plátně. Vedle barvy se nachází čtverec, ve kterém je zobrazen typ vrstvy. Kromě toho slouží ke změně pořadí vrstev. Stačí na něm podržet prst a přetáhnout ho na požadovanou pozici. Pořadí vrstev v seznamu určuje stejně jak v klasických GIS nástrojích překrývání mapových vrstev.

Pro většinu operací s vrstvami je nutné mít některou vybranou. Výběr vrstvy se provede kliknutím v místě názvu vrstvy. Některé operace nemají ikonu v oblasti 3 na schématu. Menu s dalšími operacemi se zobrazí po dlouhém stisku na oblast názvu mapové vrstvy. Přes menu je možné mapovou vrstvu smazat nebo exportovat do Shapefile (kapitola 3.9).

Na obrázku 4.3 jsou vidět ikony, které jsou zobrazené v nástrojové liště pod číslem 3 ve schématu na obrázku 4.2 pro seznam mapových vrstev. Jejich význam je následující:

1. skryje postranní panel,
2. přepne panel na atributovou tabulku vybrané vrstvy,
3. skryje nebo zobrazí vybranou mapovou vrstvu,
4. nastaví měřítko a pozici plátna tak, aby byla vidět celá vybraná vrstva,
5. otevře dialog pro přidání nové mapové vrstvy.



Obrázek 4.3: Ikony pro seznam mapových vrstev.

#### 4.2.4 Vytváření prázdné mapové vrstvy

Po kliknutí na tlačítko 5 z obrázku 4.3 se otevře dialog, ve kterém je možné vybrat buď import Shapefile, nebo vytvoření prázdné vrstvy. V dialogu pro vytvoření prázdné vrstvy se zadává název a typ nové vrstvy. Dále je možné přidat libovolný počet atributů, které mají název a typ. Dialog obsahuje vždy jeden přednastavený atribut `datetime`. Pokud vrstva obsahuje atribut `datetime` a vytvoří se nový objekt, tak hodnota atributu je předvyplněna na aktuální datum a čas.

#### 4.2.5 Atributová tabulka

Atributová tabulka se zobrazí v levém panelu pod číslem 2 ve schématu na obrázku 4.2. Jedná se o klasickou tabulku, která má na začátku hlavičku, ve které jsou názvy atributů vybrané vrstvy. Každý řádek tabulky koresponduje s jedním objektem. V tabulce je možné kliknutím vybrat jeden řádek. Pokud se plátno nachází v prohlížečím módu (kapitola 4.2.6), tak se korespondující objekt zvýrazní v plátně. Pokud se objekt nachází mimo zobrazené území, tak se pohled vycentruje na něj.

Na obrázku 4.4 jsou vidět ikony zobrazené pod číslem 3 ve schématu na obrázku 4.2, když je vidět atributová tabulka. Jejich význam je následující:

1. skryje postranní panel,
2. přepne panel na seznam mapových vrstev,
3. nastaví měřítko a pozici plátna tak, aby byl vidět celý vybraný objekt,
4. otevře dialog pro editaci atributů vybraného objektu,
5. smaže vybraný objekt.



Obrázek 4.4: Ikony pro atributovou tabulku.

#### 4.2.6 Horní nástrojová lišta

Pod číslem 4 ve schématu na obrázku 4.2 se nachází horní nástrojová lišta. Lišta se automaticky skrývá po uplynutí určitého času od jejího zobrazení. Lišta je poloprůhledná a zobrazuje se nad zbytkem okna. V levé části se nachází ikona šipky pro možnost ukrytí lišty dříve, než by se skryla sama.

Na obrázku 4.5 jsou vidět ikony zobrazené v horní nástrojové liště. Trvale jsou vidět pouze ikony 3, 4 a 6. Ostatní se zobrazují podle stavu programu. Ikony 1, 2, 3 a 4 slouží k zapínání a vypínání některé služby nebo módu. Pokud je daná služba vypnutá, tak ikona je černá. Pokud je zapnutá, tak je ikona bílá. Význam jednotlivých ikon je následující:

1. je viditelná pouze v editačním modu (kapitola 4.2.1); zapíná nebo vypíná manuální výběr souřadnice (kapitola 4.2.1),
2. je také viditelná pouze v editačním módu; zapíná nebo vypíná manuální výběr souřadnice pro tvorbu topologie geoobjektů (kapitola 4.2.1),
3. zapíná nebo vypíná editační mód,
4. zapíná nebo vypíná GPS,
5. je viditelná pouze pokud je zapnutá GPS; kliknutím na ni je pohled plátna vycentrován na aktuální souřadnice získané z GPS,
6. otevře nabídku s nastavením programu.



Obrázek 4.5: Ikony pro seznam mapových vrstev.

#### 4.2.7 Editační tlačítka v plátně





Pod číslem 5 ve schématu na obrázku 4.2 se nachází oblast, ve které se zobrazují editační tlačítka. Tlačítka jsou zobrazena pouze v editačním módu (kapitola 4.2.1). Zobrazují se pouze ta, která je možné v daný okamžik použít, což zjednodušuje ovládání. Jsou částečně průhledná, aby příliš vizuálně nezmenšila mapové plátno. Je důležité, aby byla dostatečně velká, aby se pohodlně používala. Na obrázku 4.6 jsou zobrazeny ikony editačních tlačítek. Tlačítka 1, 2 a 3 se objevují na levé spodní straně plátna a tlačítka 4, 5 a 7 se objevují na straně pravé. Ikona 6 se v některých případech objevuje místo ikony 5 a podobně ikona 8 je alternativou k ikoně 7.

Zaznamenat souřadnice z GPS je možné buď manuálně, nebo je nechat zaznamenávat automaticky. Automatický záznam funguje tak, že po určité minimální vzdálenosti od posledního zaznamenaného vrcholu objektu se vytvoří vrchol nový. Takové řešení je lepší než ukládat novou souřadnici jen na základě minimálního času od posledního zaznamenání pozice, protože pokud by se člověk zastavil, tak by se vytvářely nesmyslné vrcholy. V nastavení programu je možné upravit minimální vzdálenost mezi vrcholy, minimální čas od posledního záznamu, který má význam hlavně v šetření baterie, a minimální přesnost souřadnice, aby se zamezilo zaznamenání příliš chybných souřadnic.

Význam tlačítek z obrázku 4.6 je následující:

1. uloží editovaný objekt,
2. vrátí se na uložený stav objektu před začátkem editace,

3. pokud je vrchol vybráný, tak ho smaže, jinak smaže celý editovaný objekt,
4. přesune vybraný bod nebo vrchol na pozici danou buď GPS nebo manuálně vybranou,
5. přidá bod nebo vrchol se souřadnicemi danými buď GPS, nebo manuálně vybranými; pokud je vybráný vrchol, tak se vkládá do posloupnosti souřadnic před něj, jinak na konec,
6. ikona se zobrazí, pokud se přidává vrchol na souřadnice, již existujícího vrcholu pro zachování topologie mezi geoobjekty,
7. umožňuje spustit automatický záznam vrcholů; zobrazuje se pouze pro liniové a polygonové vrstvy,
8. pokud je spuštěný automatický záznam, tak se zobrazí místo ikony číslo 7; umožňuje přerušeni automatického záznamu.

	levé	pravé	pravé
1	✓	4	
2	←	5	
3	✗	7	
		8	

Obrázek 4.6: Ikony pro seznam mapových vrstev.

Při ukládání nového objektu se objeví dialog, který umožňuje vyplnit hodnoty všech atributů. Pole pro vyplňování atributů dovolí vyplnit hodnotu pouze tak, aby odpovídala typu atributu. Například pro celočíselný typ je možné zadat pouze číslice.

## Kapitola 5

# Implementace programu

Následující kapitola se zabývá detaily, které jsou nutné k implementaci aplikace. Popisuje zvolenou platformu, knihovny a způsob implementace některých zásadních částí výsledného programu.

### 5.1 Volba platformy

Z mobilních operačních systémů, které připadaly v úvahu, jsem zvolil Android. Android je nejrozšířenější mobilní platforma [3]. Oproti operačním systémům iOS nebo Windows Phone má některé nevýhody. Existuje mnoho zařízení s různým hardwarem, na kterých běží, takže je náročnější odladit aplikaci tak, aby fungovala alespoň na většině z nich. Například velké rozdíly jsou v rozlišení a rozměrech displejů.

Na druhou stranu má Android i výhody. Protože je nejrozšířenější mobilní platformou, tak se kolem něj vytvořila velká komunita programátorů, což usnadňuje vývoj. Vyvíjet je možné na operačních systémech Linux, Windows i Mac OS. Spouštět vytvářenou aplikaci na vlastním zařízení je možné i bez registrace a placení poplatků.

### 5.2 Popis platformy Android

Android je open source platforma vlastněná firmou Google [3]. Pro Android se vyvíjí převážně v programovacím jazyce Java. V aplikacích je možné využít kód psaný v jazyce C/C++, ale této možnosti se využívá většinou pouze v případech, kdy je třeba využít nějakou knihovnu, která není v jazyce Java dostupná.

Platforma Android je založena na linuxovém jádře. Součástí platformy jsou knihovny psané v jazyce C/C++, kterými jsou například [3]:

- Media Libraries - pro přehrávání různých audio a video formátů,
- LibWebCore - pro zobrazení html stránky,
- libc - standardní knihovna jazyku C vyladěná pro embedded zařízení,
- SQLite - odlehčená relační databázová knihovna,
- OpenSSL - knihovna pro kryptografické funkce,
- FreeType - knihovna pro renderování fontů,

- OpenGL - knihovna pro vykreslování 3D grafiky.

Aplikace běží ve virtuálním stroji. Pro Android byl vyvinut firmou Google aplikační virtuální stroj Dalvik [3]. Nový virtuální stroj vznikl kvůli licenčním právům. Jazyk Java a jeho knihovny jsou volně šiřitelné, ale Java Virtual Machine není. Dalším důvodem byla optimalizace virtuálního stroje pro mobilní zařízení, kde byl kladen velký důraz na úsporu energie.

### 5.2.1 Vývoj na platformě Android

Pro vývojáře je k dispozici Android software development kit (SDK), který obsahuje debugger, knihovny, emulátor, dokumentaci, příklady kódu a tutoriály [3]. Oficiálně podporované vývojové prostředí je Eclipse s ADT pluginem. Pro vývoj v jazyku C/C++ je k dispozici Native Development Kit (NDK).

Základním prvkem každé aplikace na platformě Android je komponenta, která se nazývá aktivita (anglicky activity) [33]. Aktivita reprezentuje jednu obrazovku s uživatelským rozhraním. Aplikace může být složena z více aktivit, které jsou na sobě nezávislé. Každá aplikace musí obsahovat nejméně jednu třídu, která dědí z třídy `Activity`.

Aktivita se může nacházet ve čtyřech různých stavech [39]:

1. je aktivní,
2. ztratila fokus, ale je stále viditelná - je pozastavena,
3. překryje ji jiná aktivita - je zastavena,
4. systém aktivitu vymazal z paměti a při otevření musí být kompletně obnovena.

Aktivita má definováno, jaké metody a v jakém pořadí jsou volány při přecházení mezi jednotlivými stavy. Některé ze zmíněných metod jsou například `onCreate()`, `onResume()`, `onPause()`, `onDestroy()` a další. Pokud aktivita není aktivní, tak ji systém může zrušit například kvůli nedostatku paměti [39].

Programátor si musí zajistit implementaci metod, které jsou k tomu určené, aby si uložil a obnovil hodnoty proměnných, jenž jsou nutné k uložení aktuálního stavu. Metoda určená pro uložení stavu se nazývá `onSaveInstanceState(Bundle outState)` a metoda pro obnovu `onRestoreInstanceState(Bundle savedInstanceState)`. Atribut typu `Bundle` slouží k serializaci a deserializaci proměnných. Aktivita obsahuje spoustu dalších metod, které začínají slabikou `on` a obsluhují různé události. Například metoda `onOptionsItemSelected()` slouží k vytvoření menu aplikace a metoda `onOptionsItemSelected(int featureId, MenuItem item)` je volána, pokud uživatel vybral některou položku v menu.

Další komponentou, kterou je možné využít, je služba (service) [33]. Služba se používá v případě, že chceme spustit nějakou dlouho trvající činnost nezávisle na grafickém uživatelském rozhraní. Například pro přehrávání hudby na pozadí.

Aktivity a služby jsou spouštěny pomocí asynchronní zprávy, která se nazývá `intent` [33]. Ten slouží ke komunikaci mezi aktivitami nebo aplikacemi. Dá se využít například pro otevření souboru. Aplikace nemusí implementovat vlastní souborový dialog. Stačí, když zašle `intent`, že chce najít cestu k souboru, a systém uživateli otevře program, který ho umí obsloužit. Uživatel vybere soubor a nalezená cesta se pomocí `intentu` zašle zpět aplikaci, která ho vyvolala.

Android původně běžel pouze na smartphonech, které mají malý displej. Na malém displeji obvykle není potřeba dělit obrazovku na panely. Později se začal používat na tabletech a jiných zařízeních s větším displejem. Proto byla kvůli nim přidána nová komponenta, která se nazývá fragment [34]. Fragment má podobný životní cyklus jako aktivita, ale jedná se o oddělený blok grafického uživatelského prostředí, který je možné zobrazit buď samostatně na malém displeji, nebo společně s jiným fragmentem na displeji větším. Hlavní komponentou zůstává aktivita. Fragments se pouze zobrazují v rámci aktivity.

Jako typický příklad pro použití fragmentu lze uvést e-mailového klienta. Pokud klient poběží na smartphone, tak uživatel přepíná mezi obrazovkou seznamu e-mailů a detailem jednoho e-mailu. Na tabletu bude mít klient v postranním panelu zobrazený seznam e-mailů a zároveň vedle okno s vybraným emailem.

Kromě souborů se zdrojovým kódem v jazyce Java obsahuje aplikace pro platformu Android ještě xml soubory, ve kterých jsou definované některé její součásti. Každá aplikace musí povinně obsahovat soubor `AndroidManifest.xml`, ve kterém se nachází název programu, verze, použité SDK, práva a aktivity, které jsou jeho součástí. Ve formátu xml je možné definovat i grafické uživatelské rozhraní a zdroje. Zdroje, které mohou být definovány ve formátu XML jsou řetězce, styly, barvy, menu, nastavení a další. Definice řetězců v xml má velkou výhodu v jednoduché lokalizaci.

Vývojáři platformy Android počítali s tím, že aplikace poběží na velkém množství zařízení s různým hardwarem. Jedna z nejproblematictějších věcí je různorodost v rozlišení displeje. Při vývoji je nutné myslet na to, že pokud použijeme ikonu o velikosti například 32x32 pixelů, tak pro zařízení s velkým rozlišením může být velice malá a může být na ni těžké kliknout. Naopak při malém rozlišení může zabírat zbytečně moc místa. Z toho důvodu se bitmapy pro android generují v několika rozměrech a systém si vybere tu bitmapu, která nejlépe koresponduje s rozlišením zařízení, na kterém běží. Další věci, která pomáhá s řešením zmíněného problému, jsou jednotky nezávislé na rozlišení, ve kterých je možné definovat rozměry a pozice prvků grafického uživatelského rozhraní.

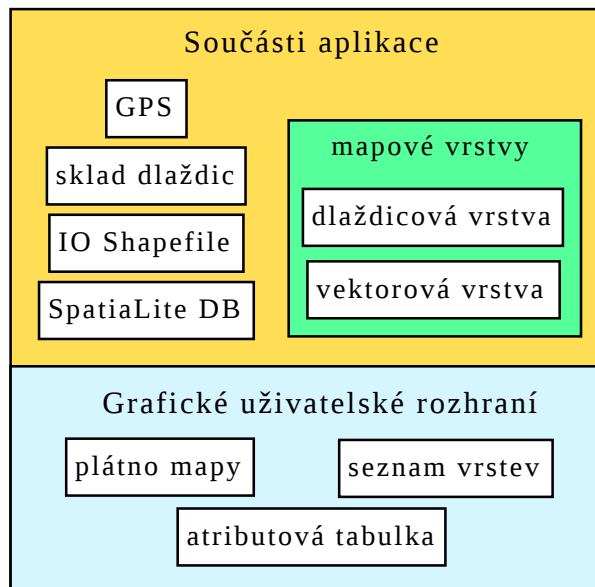
### 5.3 Dekompozice implementovaného programu

Na obrázku 5.1 je vidět dekompozice aplikace na logické části. Základem aplikace je grafické uživatelské rozhraní, které referencuje ostatní součásti aplikace a řídí chod celého programu.

Grafické uživatelské rozhraní se dá rozdělit na tři hlavní části: plátno pro zobrazení mapových vrstev (kapitola 5.5), seznam mapových vrstev a atributová tabulka aktuální vrstvy. Poslední dvě zmíněné části se zobrazují v levém panelu (pod číslem 2 na obrázku 4.2).

Další součásti aplikace zobrazené na schématu 5.1 jsou následující:

- GPS - stará se o komunikaci s GPS modulem a ostatními službami, které se starají o získání aktuální polohy (kapitola 5.6),
- sklad dlaždic - stará se o stahování dlaždic z OpenStreetMap a jejich uložení (kapitola 5.7),
- IO shapefile - část zodpovědná za import a export ve formátu Shapefile (kapitola 5.8),
- SpatiaLite DB - rozhraní pro přístup k vrstvám a geoobjektům uloženým v SpatiaLite databázi (kapitola 5.9),
- mapové vrstvy - část aplikace, která se stará o mapové vrstvy; v aplikaci jsou aktuálně podporované dva typy vrstev (kapitola 5.10):



Obrázek 5.1: Schéma dekompozice aplikace.

- vektorová vrstva - editovatelná vrstva, která může být jednoho ze tří typů: bodová, liniová nebo polygonová (kapitola 5.11),
- dlaždicová vrstva - funguje jako podkladová mapa (kapitola 5.12).

## 5.4 Knihovny třetích stran

V implementovaném programu jsem využil čtyři knihovny třetích stran:

**SpatiaLite-Android** je využita k uložení vektorových mapových vrstev. Knihovna bude více popsána v kapitole 5.9.

**JTS Topology Suite** (kapitola 3.8.3) je knihovna, která je využita pro objektovou reprezentaci vektorových geoobjektů. Použití JTS Topology Suite je zvláště výhodné v kombinaci s knihovnou SpatiaLite, protože SpatiaLite umí vrátit objekty ve formátu Well-known Text nebo Well-known binary. Knihovna JTS Topology Suite dokáže oba formáty parsovat. Je dělaná přesně pro to, na co byla použita v implementovaném programu, takže nebylo nutné implementovat věci, bez kterých by se jinak program neobešel.

**Java shapeFile Reader** je knihovna od Thomase Diewalda [49], která umí číst Shapefile. Použití knihovny bude popsáno více v kapitole 5.8.

**drag-sort-listview** je knihovna, která je distribuována pod licencí Apache License [42]. Knihovna implementuje grafickou komponentu pro seznam, ve kterém je možné měnit pořadí prvků tažením prstu. Každá položka seznamu obsahuje oblast, za kterou je možné chytnout a přesunout položku na jinou pozici v seznamu. Přetahování položky v jiném místě způsobí obvyklé skrolování seznamu. Na mobilních zařízeních je takové chování u seznamů, kde potřebujeme měnit pořadí, obvyklé. Bohužel Android neobsahuje oficiální komponentu, která by zmíněné chování umožňovala. Z toho důvodu

jsem musel pro seznam mapových vrstev, u kterých je třeba měnit pořadí vrstev, využít knihovnu drag-sort-listview.

## 5.5 Plátno pro zobrazení mapových vrstev

Pro implementaci plátna jsem využil standardní komponentu `SurfaceView`, která poskytuje plátno, do kterého je možné vykreslovat geometrické tvary i bitmapy.

Pro plátno bylo nutné implementovat obsluhu dotyků. Jedná se o krátký dotyk, gesto pro posun a gesto pinch (kapitola 4.1) pro zoom. Obsluha gest se dělá překrytím metody `onTouchEvent`, která je volaná opakovaně během toho, co se uživatel dotýká displeje. Metoda má jeden parametr typu `MotionEvent`, ze kterého je možné zjistit souřadnice dotyku a typ události. Je možné rozlišit situaci, kdy se uživatel poprvé dotkne displeje, když posunuje prstem po displeji, když se dotýká dvěma prsty nebo když ukončí dotyk. Díky tomu není velký problém implementovat gesta pro posun a pinch. Pro krátký dotyk obsahuje Android třídu `GestureDetector`, která ho rozpozná.

Při posunu a zoomu se ukázalo vhodné optimalizovat překreslování plátna. Dokud se uživatel dotýká displeje, tak se do plátna překresluje neustále to samé, akorát zobrazená bitmapa se transformuje, tak aby byla posunutá a zvětšená podle posunu prstů. Až v momentě, kdy se uživatel přestane dotýkat displeje, se aktualizuje pozice a rozměry zobrazené oblasti a načtou se nová vektorová data a dlaždice.

Význam krátkého dotyku se mění podle mnoha aspektů. Pokud je jeho souřadnice blízko levého nebo horního okraje plátna, tak se zobrazí levý panel nebo horní nástrojová lišta. Ostatní případy záleží na módu programu a jsou popsány v kapitole 4.2.1.

Souřadnice plátna narůstají v jiném směru než souřadnice geografických souřadných systémů (kapitola 3.4). Plátno má nejmenší hodnoty obou souřadnic v levém horním rohu a naopak geografické souřadné systémy je mají v levém spodním rohu. Podle toho je třeba přepočítat výsledné souřadnice, na kterých se zobrazí daný objekt.

## 5.6 Služby pro získání polohy

Pro získání aktuální polohy má android speciální třídu `LocationManager`. Přihlášení k získávání informace o pozici se dělá metodou `requestLocationUpdates`. Té se předá jako argumenty provider, minimální čas, minimální vzdálenost a instance třídy `LocationListener` [35].

Provider může být buď `GPS_PROVIDER`, který určuje, že se aktuální poloha získá z GPS, nebo `NETWORK_PROVIDER`, který zjišťuje polohu pomocí Wi-Fi nebo GSM sítě [35].

Druhý argument určuje minimální čas mezi updaty polohy. Minimální čas ovlivňuje výdrž baterie. Oproti tomu zvětšení minimální vzdálenosti mezi polohami energii nešetří, protože systém stejně musí kontrolovat, jaká je vzdálenost [35]. Avšak lze pomocí ní ošetřit, aby se neměnila poloha, když se GPS přijímač nehýbe, což je způsobeno nepřesností polohy získané z družicového navigačního systému.

Čtvrtým argumentem je objekt, v jehož metodě `onLocationChanged` se obslouží update pozice.

V programu popisovaném v diplomové práci je příjem polohy implementován dvakrát. Jeden se stará o to, aby měl uživatel v plátně na souřadnici, kde se nachází, vykreslenou značku. Tato souřadnice se použije pro manuální záznam polohy (kapitola 4.2.7). Druhý se spouští pouze v případě automatického záznamu linie nebo polygonu, protože automatický

záznam musí běžet jako služba, aby se souřadnice nepřestaly zaznamenávat při vypnutí displeje nebo přepnutí do jiné aplikace.

Pro automatický záznam souřadnic je nutné rozlišit dva případy. V prvním případě, pokud je zobrazené plátno s mapovými vrstvami, tak je třeba, aby se každá nová souřadnice rovnou vykreslila do plátna. V druhém případě, pokud je aktivní jiná aplikace, nebo dokonce zhasnutý displej, není možné, aby služba posílala nové souřadnice aplikaci. Je to dáno životním cyklem aktivity (kapitola 5.2.1), protože v momentě, kdy aktivita přestává být aktivní, tak uloží svůj stav a ten si obnoví, až když se stane znovu aktivní. Co se mezitím změnilo v paměti aplikace je přepsáno při obnově stavu. Proto si služba postupně ukládá získané souřadnice do seznamu. Až se teprve aplikace obnoví, tak jí pošle všechny nové souřadnice.

## 5.7 Sklad OpenStreetMap dlaždic

Služba OpenStreetMap vrací dlaždice jako png obrázky. Pro uložení se nabízelo přímé uložení png obrázků do adresáře na kartě. Jako vhodnější jsem zvolil ukládání dlaždic do SQLite databáze. Takové řešení má velkou výhodu v jednodušší správě dlaždic. SQL příkaz umožní jednoduše najít dlaždice, které je třeba zobrazit nebo které jsou zastaralé, takže je třeba je obnovit.

Část programu, která je zodpovědná za správu dlaždic, je rozdělena na dvě třídy: `TileCache` a `TileDownloader`. Třída `TileCache`, která je vytvořena podle návrhového vzoru singleton, takže má pouze jednu instanci, se stará o komunikaci s databází. Pokud dlaždicová vrstva potřebuje určité dlaždice, tak o ně požádá instanci třídy `TileCache`. Ta se podívá do databáze po dlaždicích, které má k dispozici a vrátí je. Pokud je zařízení připojeno k internetu, tak dlaždice, které jsou zastaralé nebo nebyly vůbec nalezeny, začne stahovat třída `TileDownloader` ve speciálním vlákne, aby neblokovala zbytek programu. Postupně je vkládá do databáze přes instanci třídy `TileCache`. Většina metod třídy `TileCache` je synchronizovaných, protože se k databázi přistupuje z více vláken. Při stahování dlaždic je třeba po určitém čase donutit plátno znovu se překreslit, aby se nově stažené dlaždice vykreslily.

Navrácení dlaždic z databáze a dekodování bitmapy z formátu png je časově relativně náročná operace, takže není vhodné, aby se při posunu mapy neustále prováděla. Většina dotazů na dlaždice je stejná jako předchozí dotaz. Proto se jednotlivé bitmapy dlaždic uloží do paměti a znovu se načítají z databáze, jen pokud dlaždicová vrstva žádá o jiné dlaždice než v předchozí žádosti.

Cesta k SQLite souboru s uloženými dlaždicemi je:  
<externí uložení>/TerrainGIS/tile\_cache.sqlite.

## 5.8 Import a export Shapefile

Formát Shapefile je popsán v kapitole 3.9. Shapefile je pravděpodobně nejpoužívanějším formátem pro přenos geografických vektorových vrstev. Proto bylo potřeba, aby ho implementovaný program podporoval. Knihovny, které implementovaly čtení i zápis Shapefile v Javě, byly součástí velkých knihoven a obsahovaly závislosti, díky kterým nebylo možné je využít na platformě Android. Objevil jsem akorát jednu malou knihovnu `Java shapeFile Reader` distribuovanou pod licencí GPL, která uměla Shapefile pouze číst [49].

Pro import jsem využil knihovnu Java shapeFile Reader, ale export jsem musel celý implementovat sám. Využil jsem kód knihovny Java shapeFile Reader, který jsem rozšířil i pro možnosti ukládání souborů Shapefile. Knihovna nebrala ohled na znakovou sadu při načítání atributů, takže i tento problém jsem v ní opravil.

Díky tomu, že jsem použil kód šířený pod licencí GPL, tak mě to zavazuje zachovat licenci, ale to jsem měl v plánu i tak, takže to není omezení.

Pro výběr souboru k importu jsem využil intent (kapitola 5.2.1), na základě kterého systém otevře souborový manažer, jenž dovolí uživateli vybrat shp soubor. Export Shapefile se provede do adresáře <externí úložiště>/TerrainGIS/output/ pod názvem, který zadá uživatel.

## 5.9 SpatiaLite databáze

Využití relační databáze na uložení geografických dat má velké výhody. Pomocí SQL lze vybírat z databáze jen data, která jsou potřeba, o hodně věcí se postará databáze sama. Ukládání a aktualizace dat je díky SQL také zjednodušena. Klasické relační databáze typu klient-server mají ale nevýhodu v dalším běžícím procesu a složitém systému, který je nezbytný pro fungování databáze.

SpatiaLite (kapitola 3.8.5) je dobrým kompromisem, protože data jsou uložena v jediném souboru, jedná se o docela malou knihovnu a umožňuje pohodlný přístup pomocí SQL. Navíc SpatiaLite přidává mnoho funkcí pro zpracování geografických dat, takže ve výsledku se jedná o mocný nástroj.

SpatiaLite je psaná v jazyce C, takže na platformě Android ji není možné využít přímo. SpatiaLite-Android je projekt, který obsahuje knihovnu upravenou tak, že ji lze přeložit pomocí NDK (kapitola 5.2.1) a spustit na platformě Android [36]. SpatiaLite-Android je distribuováno pod třemi licencemi MPL, GPL a LGPL, mezi kterými je možné si vybrat. Pro volání knihoven psaných v C/C++ z virtuálního stroje Javy se používá rozhraní Java Native Interface (JNI) [40]. Pro rozhraní mezi Javou a SpatiaLite je použita knihovna Java SQLite[37].

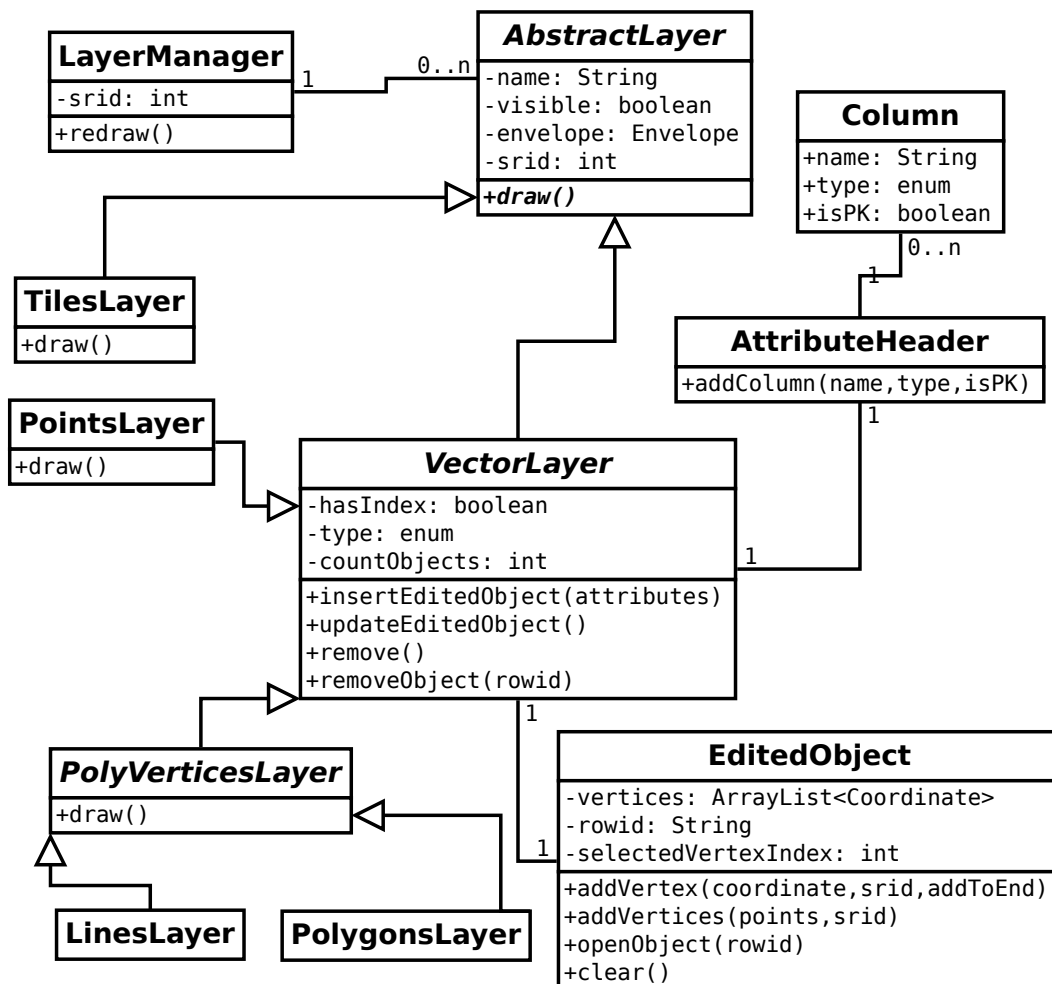
SpatiaLite databáze s vrstevmi je uložena v souboru <externí úložiště>/TerrainGIS/db.sqlite. V implementovaném programu se o komunikaci se SpatiaLite stará jediná třída s názvem `SpatiaLiteIO`. Obsahuje spoustu metod pro různé činnosti z databází. Význam nejdůležitějších z nich je následující:

- `ArrayList<Layer> getLayers()` - vrátí seznam vrstev, které se nachází v databázi; vrstva je popsána názvem, typem a geografickým souřadným systémem,
- `Envelope getEnvelopeLayer(String name, String column, boolean useRTree)` - vrátí minimální ohraničující obdélník vrstvy s názvem `name` a sloupcem `column`, který obsahuje geometrii objektů; atribut `useRTree` specifikuje, jestli se má pro zjištění obdélníku použít prostorový index,
- `Coordinate transformSRS(Coordinate point, int fromSrid, int toSrid)` - přepočítá geografický souřadný systém,
- `SpatialiteGeomIterator getObjectsInEnvelope(Envelope envelope, String name, String column, int layerSrid, int outputSrid, boolean useRTree)` - vrátí objekty, které se nachází v oblasti definované v `envelope`; metoda vrací iterátor,

- void insertObject(Geometry geom, String name, String column, int inputSrid, int tableSrid, AttributeHeader header, AttributeRecord attributesValues, boolean usePK) - vloží objekt i s atributy do databáze,
- void createEmptyLayer(String name, String type, String columns, int srid) - vytvoří novou vrstvu,
- void removeLayer(String name, String geometryColumn, boolean hasIndex) - smaže existující vrstvu.

## 5.10 Správce mapových vrstev

O vrstvy se v implementovaném programu stará správce mapových vrstev, který je tvořen třídou `LayerManager`. Instance třídy `LayerManager` je v celém programu pouze jedna, takže je implementována podle návrhového vzoru singleton. Na obrázku 5.2 je vidět zjednodušený diagram tříd, které reprezentují mapové vrstvy. Kvůli přehlednosti jsou zobrazeny pouze některé významnější atributy a metody.



Obrázek 5.2: Zjednodušený diagram tříd reprezentujících mapové vrstvy.

Správce mapových vrstev obsahuje v atributu `srid` definici společného geografického souřadného systému (kapitola 3.4) pro všechny vrstvy formou EPSG kódu (kapitola 3.5). Do společného souřadného systému se transformují všechny mapové vrstvy. Souřadný systém se odvíjí od podkladové vrstvy. Je to z toho důvodu, že vektorové vrstvy je možné jednoduše transformovat z jednoho geografického souřadného systému do jiného transformací všech vrcholů. Avšak u bitmap je transformace náročná. Tedy společný souřadný systém je založený na zobrazení Spherical Mercator (kapitola 3.11.2) s EPSG kódem 3857.

Třída `LayerManager` obsahuje metodu `redraw`, jenž slouží k překreslení všech vrstev do plátna. Správce vrstev volá v rámci metody `redraw` metody `draw` všech viditelných vrstev tak, aby bylo zachováno jejich pořadí.

Mapové vrstvy dědí z abstraktní třídy `AbstractLayer`, která obsahuje atributy společné pro všechny vrstvy:

- `name` - název vrstvy,
- `visible` - určuje, jestli je vrstva viditelná,
- `envelope` - minimální ohraničující obdélník,
- `srid` - EPSG kód souřadného systému vrstvy.

Vektorová vrstva reprezentovaná třídou `VectorLayer` bude popsána v kapitole 5.11 a dlaždicová vrstva `TilesLayer` v kapitole 5.12.

## 5.11 Vektorové mapové vrstvy

Třídy reprezentující vektorovou mapovou jsou znázorněny ve zjednodušeném diagramu tříd na obrázku (5.2). Třída `VectorLayer` implementuje většinu funkcionalitu, která souvisí s vektorovými vrstvami. Ze třídy `VectorLayer` dědí `PointsLayer` a abstraktní třída `PolyVerticesLayer`. Z `PolyVerticesLayer` dědí `LinesLayer` a `PolygonsLayer`.

Ve zděděných třídách se nachází hlavně metoda `draw`, která slouží k vykreslení vrstvy do plátna (kapitola 5.5), protože způsob vykreslení je závislý na typu vrstvy. Liniová a polygonová vrstva má společného předka třídu `PolyVerticesLayer`, protože vykreslení polygonů a linií si je blízké, takže je výhodou, aby sdílelo některou funkcionalitu.

Třída `VectorLayer` obsahuje následující atributy:

- `hasIndex` - detekuje, jestli vrstva využívá v databázi prostorový index,
- `type` - typ vrstvy,
- `countObjects` - počet objektů, které vrstva obsahuje.

Dále popíšu některé z jejích významnějších metod:

- `insertEditedObject(AttributeRecord attributes)` - vloží do SpatiaLite databáze právě editovaný nový objekt; `attributes` obsahuje záznam atributů pro nový objekt,
- `updateEditedObject()` - aktualizuje editovaný objekt ve SpatiaLite databázi,
- `remove()` - ze SpatiaLite databáze smaže vrstvu, kterou reprezentuje,

- `removeObject(String rowid)` - ze SpatiaLite databáze smaže objekt s identifikátorem `rowid`.

Třída `VectorLayer` obsahuje vnořenou třídu `EditedObject`, která reprezentuje právě editovaný objekt (kapitola 4.2.7). Díky tomu má každá vrstva vlastní editovaný objekt a je možné přepínáním mezi vrstvami editovat různé objekty. Třída `EditedObject` obsahuje následující atributy:

- `vertices` - seznam vrcholů editovaného objektu,
- `rowid` - identifikátor editovaného objektu v databázi SpatiaLite,
- `selectedVertexIndex` - index vybraného vrcholu.

Třída `EditedObject` dále obsahuje veřejné metody, z nichž bych zmínil význam následujících:

- `addVertex(Coordinate coordinate, int srid, boolean addToEnd)` - do seznamu vrcholů přidá další vrchol se souřadnicí `coordinate` a v geografickém souřadném systému s EPSG kódem `srid`; jestli se přidá vrchol na konec seznamu nebo před právě vybraný vrchol, závisí na na proměnné `addToEnd`,
- `addVertices(ArrayList<Coordinate> coordinates, int srid)` - podobně jak předchozí funkce, ale umožňuje přidat víc vrcholů naráz; vrcholy se přidávají pouze na konec,
- `openObject(String rowid)` - načte objekt pro editaci s identifikátorem `rowid` z databáze SpatiaLite,
- `clear()` - zruší editovaný objekt, takže se vrátí do uloženého stavu.

Další třídou referencovanou v třídě `VectorLayer` je třída `AttributeHeader`, která definuje atributovou tabulku dané vrstvy. Každý sloupec je reprezentován třídou `Column`, která obsahuje následující atributy:

- `name` - název atributu,
- `type` - datový typ atributu; souvisí s datovými typy databáze SpatiaLite (kapitola 3.8.5),
- `isPK` - detekuje, jestli je daný sloupec ve SpatiaLite databázi označen jako `primary key`.

Každá z vektorových vrstev je zobrazena jinou barvou. Barvy se generují automaticky tak, aby byly navzájem různé. Pro generování barev jsem využil barevný model HSV, který se skládá ze tří položek:

- `Hue` - barevný tón,
- `Saturation` - sytost barvy,
- `Value` - hodnota jasu.

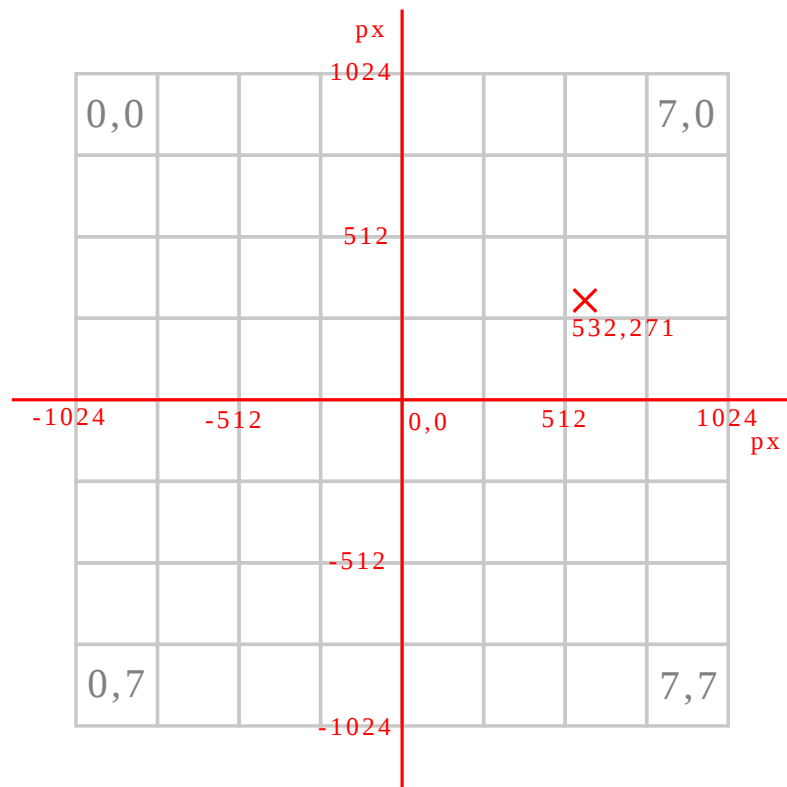
Barevný model HSV má tu výhodu, že pokud se nastaví určitá hodnota sytosti a jasu, tak změnou barevného tónu po dostatečně velkém kroku systematicky získáváme rozdílné barvy.

## 5.12 Dlaždicová mapová vrstva

Jako podkladovou vrstvu jsem se rozhodl využít internetovou dlaždicovou mapu (kapitola 3.11). V porovnání s WMS (kapitola 3.10) je přístup k dlaždicím rychlejší a je jednodušší uložit dlaždice pro offline použití.

Dlaždice jsem využil ze služby OpenStreetMap. OpenStreetMap data jsou pod licencí Open Data Commons Open Database License [14]. Oproti Google Maps a Bing maps je jejich licence svobodná. OpenStreetMap umožňuje přímý přístup k dlaždicím (kapitola 3.11.1). U Google maps je také možné přistupovat k dlaždicím přímo, ale oficiálně podporovaný přístup je pouze přes Google Maps Api (kapitola 5.12.2). Bing maps nepodporuje přímý přístup k dlaždicím a ani nenabízí API pro Android zdarma.

Pro dlaždicovou mapovou vrstvu je důležitý přepočítání souřadnice na dlaždici, ve které se nachází. Princip přepočtu bude ukázán na příkladě, který je znázorněn na obrázku 5.3. Souřadnice vykreslené červeně jsou v souřadném systému Spherical Mercator (kapitola 3.11.2) převedené podle úrovně přiblížení na pixely. Svislá osa koresponduje s nultým poledníkem a vodorovná osa s rovníkem, takže znaménkem je daná polokoule. Šedou barvou jsou znázorněny dlaždice. Celý svět je složen z  $8 \times 8$  dlaždic. To znamená, že úroveň přiblížení je 3 (kapitola 3.11.1). Každá dlaždice má rozměry  $256 \times 256$  pixelů, takže svět je zobrazen na  $2048 \times 2048$  pixelech.



Obrázek 5.3: Přepočítání souřadnice v pixelech na dlaždici, ve které se nachází.

Ukázkový bod se souřadnicemi 532, 271 se nejdříve převede na souřadnice, ve kterých platí, že počátek je v levém horním rohu. Polovina velikosti světa je 1024 pixelů. Pokud tato hodnota bude uložena v proměnné `world_half`, tak nové souřadnice dostaneme vztahem  $(x + world\_half, -y + world\_half)$ . Nové souřadnice tedy budou (1556, 753). Konkrétní

dlaždice se už přepočte jen za pomoci celočíselného dělení a rozměru jedné dlaždice ( $x \text{ div } 256, y \text{ div } 256) = (6,2)$ ).

### 5.12.1 Knihovna Osmroid

Osmroid je open source knihovna, která implementuje API pro použití dat z internetových dlaždicových map (kapitola 3.11), a to hlavně těch, které jsou založené na OpenStreetMap datech [23]. Avšak umí stahovat mapy například i z Google Maps. Osmroid je koncipován jako náhrada za Google Maps Android API (kapitola 5.12.2). Stejně jako on nabízí komponentu grafického uživatelského rozhraní MapView. MapView se přidá do okna aplikace mezi ostatní komponenty a o vše se už stará samo. Je možné do něj přidat různé komponenty například vlastní body zájmu, přehledovou mapku nebo linie. Knihovna je distribuována pod licencí Apache License.

Z počátku jsem v diplomové práci využíval knihovnu Osmroid pro stahování a ukládání dlaždic z OpenStreetMap, ale nakonec se ukázalo jako vhodnější implementovat sklad vlastním způsobem (kapitola 5.7) kvůli jednoduššímu zajištění požadované funkčnosti a lepší udržovatelnosti kódu.

### 5.12.2 Google Maps Android API

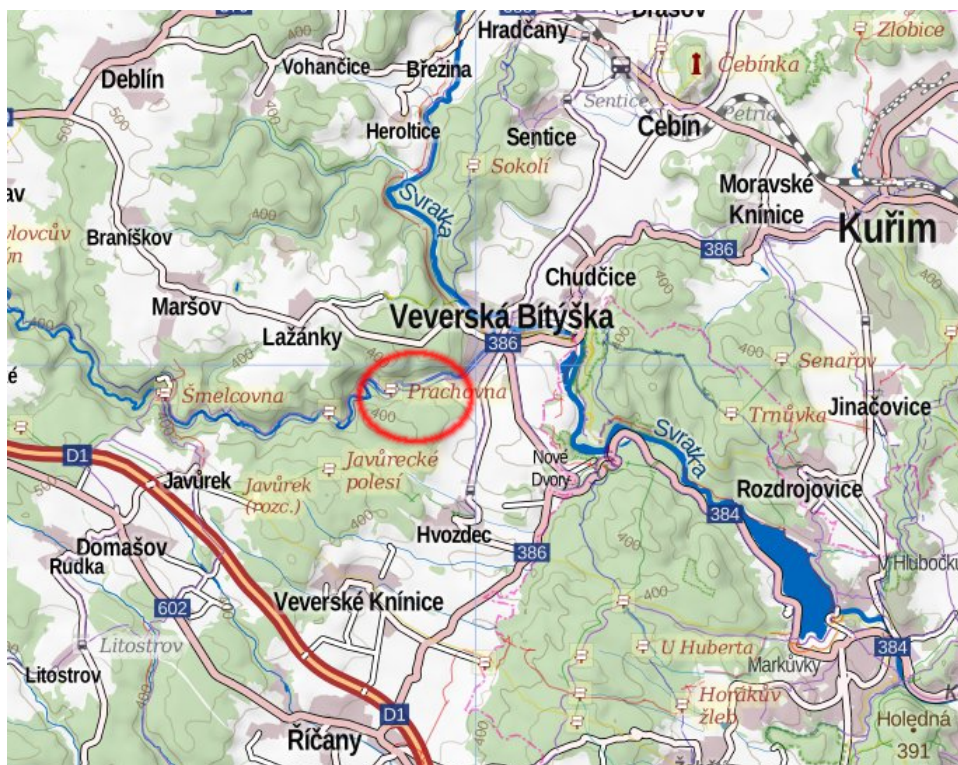
Google Maps Android API umožňuje vývojáři použít data z Google Maps ve své aplikaci [21]. Umožňuje v mapě zobrazit body zájmu formou ikon, linie, polygony nebo libovolnou bitmapu. Knihovnu vyvíjí firma Google.

Google Maps Android API se ukázalo jako nevhodné pro využití v diplomové práci. Neobsahuje požadované vlastnosti pro stahování a ukládání dlaždic.

## Kapitola 6

# Protokol o mapování v terénu

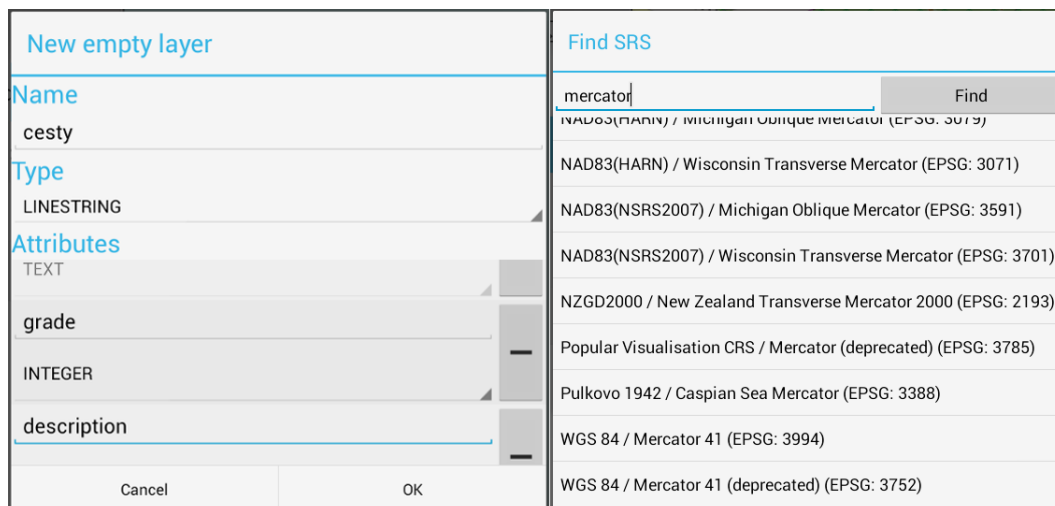
Cílem mapování bylo vytvořit několik vektorových mapových vrstev ve zvolené lokalitě. Pro mapování jsem zvolil část Údolí Bílého potoka u obce Veverská Bítýška. Zaměřil jsem se na cesty, vodstvo a možnosti překročení Bílého potoka. Oblast mapování je vyznačena červenou elipsou na obrázku 6.1.



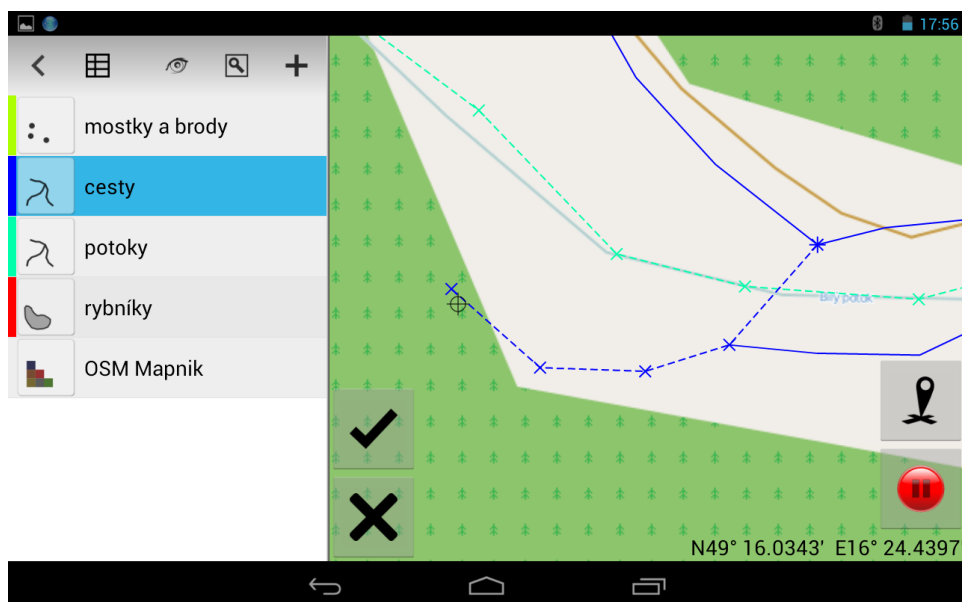
Obrázek 6.1: Přehledová mapa oblasti mapování.

### 6.1 Postup mapování

Pro mapování jsem využil tablet Google Nexus 7. Geografickou informaci jsem získával jednak pomocí integrovaného GPS modulu, ale hlavně externím bluetooth GPS modulem od firmy NaviLock. Mapování jsem prováděl celé pěšky.



Obrázek 6.2: Screenshoty dialogů.

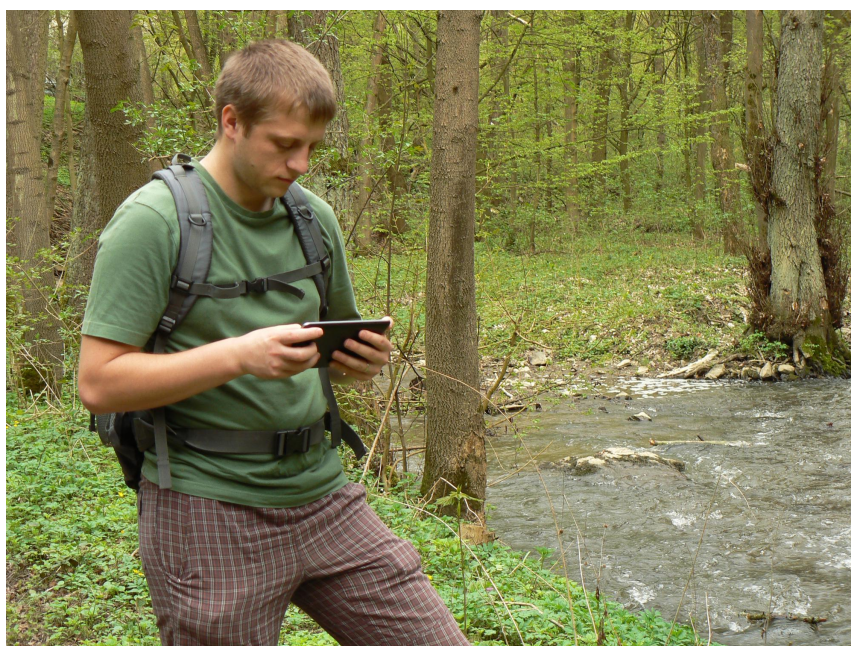


Obrázek 6.3: Screenshot záznamu cesty.

V rámci přípravy na mapování jsem vytvořil následující čtyři vrstvy a jejich atributy:

- mostky a brody - bodová vrstva
  - datetime - text
  - description - text
- cesty - liniová vrstva
  - datetime - text
  - grade - celočíselná hodnota v rozsahu 1-5 určující kvalitu cesty (1 znamená nejlepší)

- description - text
- potoky - liniiová vrstva
  - datetime - text
  - name - text
  - description - text
- rybníky - polygonová vrstva
  - datetime - text
  - name - text
  - description - text



Obrázek 6.4: Fotografie z mapování.

Atribut datetime slouží k uložení data a času zaznamenání objektu. Atribut description slouží k popisu objektu. Atribut name je název objektu. Atribut grade je stupeň kvality cesty. Je z rozsahu 1-5 a nižší číslo znamená vyšší kvalitu.

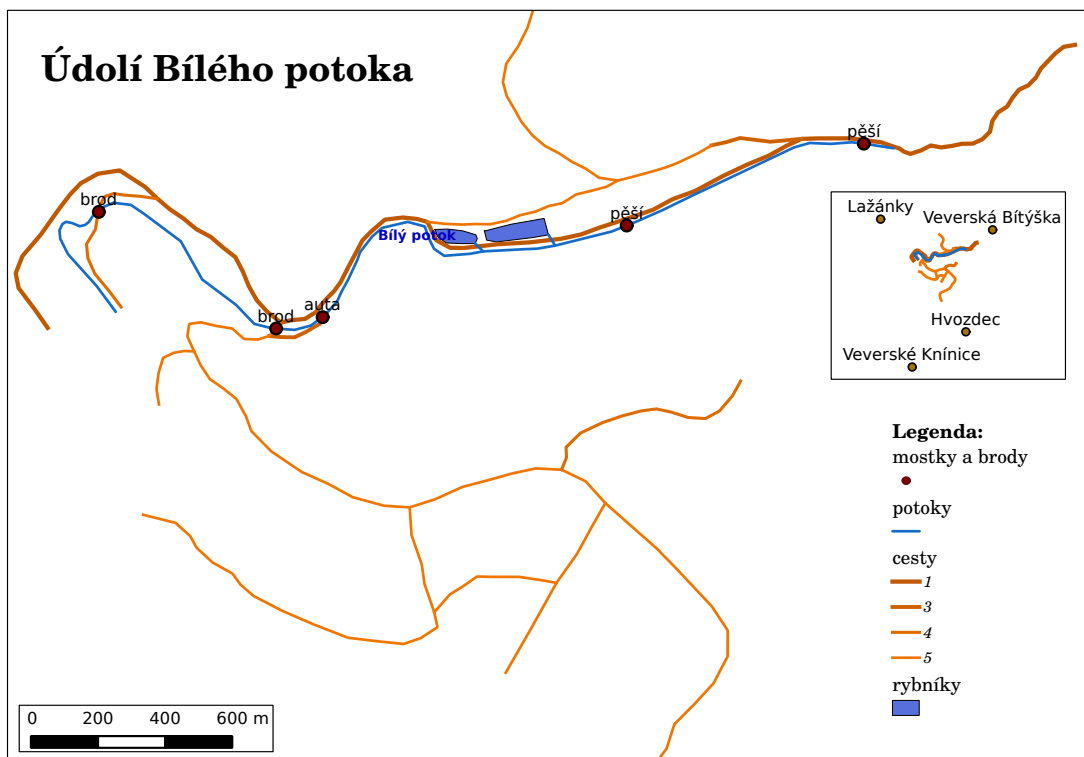
Za pomoci existující mapy jsem si prohlídl oblast, do které jsem se chystal jet mapovat. Rozhodl jsem se, kde zaparkuji auto a jakým způsobem projdu oblast.

Obrázky dokumentují mapování. Na obrázku 6.2 jsou dva dialogy. První ukazuje dialog pro vytvoření prázdné vrstvy a druhý vyhledávání souřadného systému podle názvu. Obrázek 6.3 zobrazuje screenshot programu a obrázek 6.4 byl vyfocen při mapování.

## 6.2 Výsledek mapování

Výsledkem mapování jsou modely geoobjektů a jejich atributy. Na obrázku 6.5 je vidět mapa, kterou jsem vytvořil v programu Quantum GIS z vrstev vytvořených během mapování.

Při vizualizaci cest byl použit atribut grade. Čím tmavší a širší čára cesty, tím je cesta kvalitnější.



Obrázek 6.5: Mapa vytvořená ze zaznamenaných vrstev.

# Kapitola 7

## Závěr

Lze prohlásit, že neexistuje aplikace vhodná pro mapování v terénu s moderními mobilními počítačovými zařízeními. Programů, které by alespoň umožňovaly prohlížet geografická data, několik je, ale jsou zpoplatněné. Starší geodetické elektronické zápisníky jako ArcPad nebo gvSIG pro zastaralou platformu Windows Mobile jsou nevyhovující. Proto implementace speciální aplikace pro mapování v terénu na moderních mobilních zařízeních má smysl.

Kvalitní open source knihovny pro zpracování geografické informace jsou převážně dostupné pro aplikace vyvíjené pro desktop. Na platformě Android je vývojář omezený různými aspekty, takže některé knihovny nelze použít. Avšak nakonec ty nejdůležitější knihovny pro správné fungování implementovaného programu k dispozici byly.

Pomocí implementované aplikace bylo provedeno experimentální mapování, jehož výstupem byly geoobjekty ve čtyřech různých vektorových vrstvách. Mapování ukázalo, že program ke sběru geografických dat lze využít.

Program bude uvolněn pod open source licenci GPL a umístěn na market s aplikacemi. Možnosti dalšího rozvoje jsou velké. Mnoho funkcionality, kterou známe z desktopových nástrojů GIS, by bylo výhodné doimplementovat i do výsledné aplikace. Mohla by například podporovat některé rastrové formáty, umožnit uživateli měnit styl vykreslení jednotlivých vrstev nebo měřit vzdálenosti a plochy. Pro mapování by bylo výhodné, kdyby bylo možné použít speciální bodovou vrstvu, kde by ke každému bodu byla přiřazena fotografie pořízená v daném místě. Program by mohl umožnit export a import v dalších formátech.

# Literatura

- [1] ESRI Shapefile Technical Description [online].  
<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>, 1998-07.
- [2] Díl 6. - Staré navigační pomůcky. *časopis Zeměměřič*, ročník 6, č. 1+2, 1999, ISSN 1211-488X.
- [3] WWW stránky: Android (operační systém) [online].  
[http://cs.wikipedia.org/wiki/Android\\_\(opera%C4%8Dn%C3%AD\\_syst%C3%A9m\)](http://cs.wikipedia.org/wiki/Android_(opera%C4%8Dn%C3%AD_syst%C3%A9m)), 2001- [cit. 2012-01-03].
- [4] WWW stránky: Mercator projection [online].  
[http://en.wikipedia.org/wiki/Mercator\\_projection](http://en.wikipedia.org/wiki/Mercator_projection), 2001- [cit. 2012-12-29].
- [5] WWW stránky: Latitude [online]. <http://en.wikipedia.org/wiki/Latitude>, 2001- [cit. 2012-12-31].
- [6] WWW stránky: Wikimedia Commons [online]. <http://commons.wikimedia.org/>, 2001- [cit. 2013-01-02].
- [7] WWW stránky: JTS Topology Suite [online].  
[http://en.wikipedia.org/wiki/JTS\\_Topology\\_Suite](http://en.wikipedia.org/wiki/JTS_Topology_Suite), 2001- [cit. 2013-05-08].
- [8] WWW stránky: Open Geospatial Consortium [online].  
[http://en.wikipedia.org/wiki/Open\\_Geospatial\\_Consortium](http://en.wikipedia.org/wiki/Open_Geospatial_Consortium), 2001- [cit. 2013-05-08].
- [9] WWW stránky: dBase [online]. <http://en.wikipedia.org/wiki/DBase>, 2001- [cit. 2013-05-09].
- [10] WWW stránky: Slippy map tilenames [online].  
[http://wiki.openstreetmap.org/wiki/Slippy\\_map\\_tilenames](http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames), 2004- [cit. 2013-01-01].
- [11] WWW stránky: Vítejte na OpenStreetMap [online].  
[http://wiki.openstreetmap.org/wiki/CS:Main\\_Page?uselang=cs](http://wiki.openstreetmap.org/wiki/CS:Main_Page?uselang=cs), 2004- [cit. 2013-01-01].
- [12] WWW stránky: Zoom levels [online].  
[http://wiki.openstreetmap.org/wiki/Zoom\\_levels](http://wiki.openstreetmap.org/wiki/Zoom_levels), 2004- [cit. 2013-01-01].
- [13] WWW stránky: Tile Disk Usage [online].  
[http://wiki.openstreetmap.org/wiki/Tile\\_Disk\\_Usage](http://wiki.openstreetmap.org/wiki/Tile_Disk_Usage), 2004- [cit. 2013-05-10].
- [14] WWW stránky: Autorská práva a licence [online].  
<http://www.openstreetmap.org/copyright>, 2004- [cit. 2013-05-17].

- [15] OpenLayers Library Documentation: Spherical Mercator [online].  
[http://docs.openlayers.org/library/spherical\\_mercator.html](http://docs.openlayers.org/library/spherical_mercator.html), [cit. 2013-01-01].
- [16] Spatial Reference: projection 4326 [online].  
<http://spatialreference.org/ref/epsg/4326/html/>, [cit. 2013-01-01].
- [17] Dokumentace Android: Design Principles [online].  
<http://developer.android.com/design/get-started/principles.html>, [cit. 2013-01-02].
- [18] Dokumentace Android: Gestures [online].  
<http://developer.android.com/design/patterns/gestures.html>, [cit. 2013-01-02].
- [19] Domovská stránka: ArcGIS for Mobile [online].  
<http://www.esri.com/software/arcgis/about/mobile-gis-for-you>, [cit. 2013-01-02].
- [20] Domovská stránka: ArcPad [online]. <http://www.esri.com/software/arcgis/arcpad>, [cit. 2013-01-02].
- [21] Dokumentace: Introduction to the Google Maps Android API v2 [online].  
<https://developers.google.com/maps/documentation/android/intro>, [cit. 2013-01-03].
- [22] Domovská stránka: Java Map Projection Library [online].  
<http://www.jhllabs.com/java/maps/proj/>, [cit. 2013-01-03].
- [23] Domovská stránka: osmdroid [online]. <http://code.google.com/p/osmdroid/>, [cit. 2013-01-03].
- [24] Domovská stránka: PROJ.4 - Cartographic Projections Library [online].  
<http://trac.osgeo.org/proj/>, [cit. 2013-01-03].
- [25] Domovská stránka: Shapefile over Map [online].  
<https://play.google.com/store/apps/details?id=com.dabebro.som>, [cit. 2013-04-28].
- [26] Domovská stránka: SHP Viewer [online].  
<https://play.google.com/store/apps/details?id=com.afanche.android.ATViewSHP>, [cit. 2013-04-28].
- [27] Dokumentace: Datatypes In SQLite Version 3 [online].  
<http://www.sqlite.org/datatype3.html>, [cit. 2013-05-08].
- [28] Dokumentace: SpatiaLite - spatial extensions for SQLite [online].  
<http://www.gaia-gis.it/gaia-sins/spatialite-manual-2.3.1.html>, [cit. 2013-05-08].
- [29] Dokumentace: SpatiaLite Cookbook [online].  
<http://www.gaia-gis.it/gaia-sins/spatialite-cookbook/index.html>, [cit. 2013-05-08].
- [30] Domovská stránka: Features Of SQLite [online]. <http://www.sqlite.org/features.html>, [cit. 2013-05-08].
- [31] Domovská stránka: OGP Geomatics Committee [online]. <http://www.epsg.org/>, [cit. 2013-05-08].
- [32] WWW stránky: History of the JTS Topology Suite (and GEOS) [online].  
<http://tsusiatsoftware.net/jts/jts-history.html>, [cit. 2013-05-08].

- [33] Dokumentace Android: Application Fundamentals [online].  
<http://developer.android.com/guide/components/fundamentals.html>,  
 [cit. 2013-05-11].
- [34] Dokumentace Android: Fragments [online].  
<http://developer.android.com/guide/components/fragments.html>, [cit. 2013-05-11].
- [35] Dokumentace Android: LocationManager [online].  
<http://developer.android.com/reference/android/location/LocationManager.html>,  
 [cit. 2013-05-11].
- [36] Domovská stránka: spatialite-android [online].  
<https://code.google.com/p/spatialite-android/>, [cit. 2013-05-12].
- [37] Domovská stránka: SQLite Java Wrapper/JDBC Driver [online].  
<http://www.ch-werner.de/javasqlite/>, [cit. 2013-05-12].
- [38] Domovská stránka: gvSIG - Features [online].  
<http://www.gvsig.org/web/projects/gvsig-mobile/features>, [cit. 2013-05-16].
- [39] Dokumentace Android: Activity [online].  
<http://developer.android.com/reference/android/app/Activity.html>, [cit. 2013-05-17].
- [40] Java Native Interface [online]. [http://en.wikipedia.org/wiki/Java\\_Native\\_Interface](http://en.wikipedia.org/wiki/Java_Native_Interface),  
 WWW stránky: 2001- [cit. 2013-05-12].
- [41] Aitchison, A.: WWW stránky: The Google Maps / Bing Maps Spherical Mercator Projection [online]. <http://alastaira.wordpress.com/2011/01/23/the-google-maps-bing-maps-spherical-mercator-projection/>, 2011-01-19  
 [cit. 2013-01-01].
- [42] Carl Bauer: Domovská stránka: DragSortListView [online].  
<https://github.com/bauerca/drag-sort-listview>, [cit. 2013-05-11].
- [43] Erik Bachmann: Dokumentace: Xbase Data file (\*.dbf) [online].  
<http://www.clicketyclick.dk/databases/xbase/format/dbf.html>, 2010-02-27  
 [cit. 2013-05-09].
- [44] Hojgr, R.; Stankovič, J.: *GPS*. Computer Press, 2007, ISBN 978-80-251-1734-7.
- [45] Hrubý, M.: Geografické Informační systémy (předmět GIS na VUT FIT): studijní opora. 2006-09.
- [46] Jan Jiránek and Jan Říha: WWW stránky: Web Map Service (WMS) [online].  
<http://geo3.fsv.cvut.cz/wms/index.php?menu=wmsinfo>, [cit. 2013-05-10].
- [47] Konečný, M.; Kaplan, V.; Keprtová, K.; aj.: Multimediální učebnice kartografie a geoinformatika [online]. <http://147.251.65.2/ucebnice/kartografie/>, 2005  
 [cit. 2012-12-29].
- [48] Petro, B.: WWW stránky: History of User Interfaces: iPad at 3 Years [online].  
<http://billpetro.com/ipad-5th-generation-user-interface>, 2013-04-05 [cit. 2013-05-17].

- [49] Thomas Diewald: Domovská stránka: Java shapeFile Reader [online].  
<http://thomasdiewald.com/blog/?p=1382>, [cit. 2013-05-11].
- [50] Vitásek, J.; Nevosád, Z.: GEODÉZIE I (předmět na VUT FAST): studijní opora.  
2004.