

Univerzita Pardubice

Fakulta Elektrotechniky a Informatiky

Návrh a realizace mobilní aplikace využívající možnosti geolokace

Bc. Jaroslav Šafář

Diplomová práce

2013

Zadání práce

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 23.8.2013

Bc. Jaroslav Šafář

Poděkování

Tímto bych chtěl poděkovat panu Ing. Jiřímu Kyselovi za celkové vedení mé práce a podnětné rady a připomínky. Rád bych poděkoval i svému otci Jiřímu Šafářovi, který mě vždy v mém snažení podporoval.

ANOTACE

Diplomová práce se zabývá možnostmi geolokace a využitím výsledků geolokace v mobilních aplikacích webových i nativních. V rámci diplomové práce bylo úkolem navrhnout a vytvořit nativní mobilní aplikaci, která bude využívat geolokační metody, pracovat s geografickými údaji získanými pomocí těchto metod a zobrazovat je uživateli ve srozumitelné formě.

KLÍČOVÁ SLOVA

možnosti geolokace, mobilní zařízení, geografická data, navigace, GPS,

TITLE

Design and implementation of mobile application using geolocation options

ANOTATION

The thesis explores possibilities of geolocation and using of the results of geolocation in web-based and native mobile applications. The task of the thesis was to design and develop native mobile application which will use several different geolocation methods. The application will process the data obtained by these methods and present them to the user in easily understandable format.

KEYWORD

geolocation options, mobile device, geographic datam navigation, GPS,

Obsah

Úvod.....	12
1 Možnosti geolokace	13
1.1 Geolokace síťových zařízení.....	13
1.1.1 Aktivní metody	13
1.1.2 Pasivní metody.....	15
1.2 Využití elektromagnetických signálů.....	17
1.2.1 Družicové systémy GNSS	17
1.2.2 Systém pozemních stanic.....	19
1.2.3 Lokální bezdrátové sítě.....	21
1.3 Přímé metody určení pozice.....	22
1.3.1 Inerciální navigační systém	22
1.3.2 Výpočet aktuální pozice.....	24
1.4 Alternativní metody geolokace	25
1.4.1 Využití QR – kodů.....	25
1.4.2 Využití fotografie.....	25
2 Využití geolokace	27
2.1 Lokálně kontextové služby	27
2.2 Využití geolokace ve webových aplikacích.....	28
2.2.1 Geolocation API	29
2.3 Využití geolokace nativních aplikacích	31
3 Mobilní zařízení.....	33
3.1 Mobilní platformy	33
3.2 Komponenty mobilních zařízení.....	34
3.2.1 Akcelerometr	34
3.2.2 Magnetometr.....	36
3.2.3 Modul GPS	36
4 Návrh aplikace	37
4.1 Funkční požadavky	38
4.2 Návrh datových struktur.....	38
4.2.1 Uchování geografických souřadnic	38
4.2.2 Struktura mapy.....	40
4.2.3 Uložení objektů mapy.....	41

4.3	Použité metody geolokace.....	41
4.3.1	Zaměření mobilního zařízení.....	42
4.3.2	Výpočet nové pozice.....	42
4.4	Vizualizace mapy.....	45
4.5	Navigace.....	46
4.5.1	Problém nejkratší cesty.....	46
5	Implementace mobilní aplikace.....	48
5.1	Implementace datových struktur.....	48
5.1.1	Třída Binární strom.....	48
5.1.2	Třída 3D strom.....	49
5.1.3	Třída Graf.....	49
5.2	Implementace geolokace.....	51
5.2.1	Třída GPS reader.....	51
5.2.2	Třída Step detector a třída Compass.....	52
5.2.3	Třída Point calculator.....	55
5.3	Implementace mapy.....	55
5.3.1	Uložení mapy.....	55
5.3.2	Struktura mapy.....	56
5.4	Aktivity aplikace.....	57
6	Výsledek.....	59
6.1	Uživatelské rozhraní.....	59
6.2	Využití a přesnost aplikace.....	64
	Závěr.....	66
	Použitá literatura.....	67
	Příloha A – CD s vytvořenou aplikací.....	69
	Příloha B – programátorská příručka.....	70

Seznam ilustrací a tabulek

Obrázek 1 Metody CBG, Octan, GeoWeight	14
Obrázek 2 Geolokace pomocí WiFi Google Maps	16
Obrázek 3 Družicový dálkoměrný systém GPS	18
Obrázek 4 Geolokace pomocí fotografie	26
Obrázek 5 Geolokace pomocí BTS	30
Obrázek 6 Geolokace pomocí Node B	30
Obrázek 7 Geolokace pomocí sítě WiFi	31
Obrázek 8 Základní princip akcelerometru	35
Obrázek 9 Kapacitní akcelerometr	35
Obrázek 10 UseCase diagram navrhované aplikace	37
Obrázek 11 Princip třídění 3D stromu	39
Obrázek 12 Změna délky trasy při nastavené inklinaci	44
Obrázek 13 Princip Dijkstrova algoritmu	47
Obrázek 14 Implementace třídy binárního stromu	48
Obrázek 15 Implementace třídy 3D stromu	49
Obrázek 16 Implementace třídy Graf	50
Obrázek 17 Třída GpsReader	51
Obrázek 18 Implementace krokoměru	53
Obrázek 19 Implementace kompasu	54
Obrázek 20 Třída PointCalculator	55
Obrázek 21 Implementace databáze db4o	56
Obrázek 22 Třída FortripMap	57
Obrázek 23 Úvodní obrazovka aplikace	59
Obrázek 24 Zobrazení mapy v aplikaci	60
Obrázek 25 Rozložení ovládání zobrazení mapy	61
Obrázek 26 Zobrazení vyhledané trasy	62
Obrázek 27 Editační obrazovka	63
Obrázek 28 Přidání nového bodu	64
Obrázek 29 Ukázka plánu vstupních objektů podzemního prostoru	65

Tabulka 1 Srovnání přesnosti aktivních metod.....	15
Tabulka 2 Srovnání síťových metod geolokace.....	17
Tabulka 3 Srovnání přesnosti metod geolokace u GSM sítí.....	21
Tabulka 4 Srovnání metod využívající el. mag. Signály	22
Tabulka 5 Srovnání přímých metod.....	25
Tabulka 6 Srovnání alternativních metod.....	26

Seznam zkratk a značek

LBS	Lokální kontextové služby
GPS	(Global Positioning System) systém pro určování polohy
GNSS	Global navigation satellite system
BTS	(Base Transceiver Station) základová stanice sítí GSM
Db4o	Database for objects

Úvod

V současné době je využívání mobilních zařízení, jako jsou chytré mobilní telefony a tablety, velmi rozšířené a oblíbené. To umožňuje snadný přístup k informacím kdekoli a kdykoli. Vzhledem k tomu je výhodné ne-li potřebné, aby mobilní zařízení dokázalo určit svou aktuální geografickou polohu pro snadnější a přesnější splnění uživatelských požadavků.

Tato diplomová práce má za úkol popsat možnosti geolokace v mobilních zařízeních a využití geolokace v mobilních aplikacích. Výsledkem je přehled dostupných metod a technologií, které jsou využitelné při geolokaci v mobilních zařízeních, jehož obsahem je srovnání přesnosti, výhod, nevýhod a parametrů. Dále obsahuje analýzu využitelnosti výsledků metod v mobilních aplikacích nativních i webových.

Dalším úkolem diplomové práce je návrh a implementace nativní mobilní aplikace využívající geolokaci. Navrhovaná aplikace umožňuje určit novou geografickou polohu v závislosti na vykonané trase v daném směru z pozice se známou polohou. U aplikace se předpokládá využití v místech, kde není možné využívat vnější zdroj informace k použití jiné metody geolokace. Zároveň předpokládá, že není dostupný žádný využitelný mapový podklad. Úkolem aplikace je usnadnění orientace v takových prostředích a proto má dvě základní funkcionality. V první části umožňuje sběr dat a tím vytvoření mapového podkladu vykonaných tras společně s jejich popisem. Druhá část umožňuje v tomto mapovém podkladu vyhledávat nejkratší trasu a umožnit přibližné navigování v prostoru.

V úvodní části se práce zabývá možnostmi geolokace a jejich využití v mobilních aplikacích. Dále se práce zaměřuje na popis současných mobilních zařízení, hlavně na senzory využitelné z hlediska geolokace a možnosti jejich použití. V dalších kapitolách je popsán konkrétní postup návrhu a implementace mobilní aplikace. V poslední kapitole se práce zhodnotí funkčnosti aplikace z hlediska přesnosti a možnosti využití.

1 Možnosti geolokace

Pojmem geolokace se rozumí aktivita pro získání tzv. geografických dat, určujících polohu uživatele na povrchu Země [10]. Odjakživa bylo velmi důležité co nejpřesněji zjistit svou aktuální pozici. Člověk přitom využíval různé prostředky od orientace podle oblohy, přes mechanické prostředky jako je kompas, po moderní technické prostředky. S růstem potřeby snadnějšího a rychlejšího získávání geografické informace byly vyvinuty různé metody a technologie.

V současnosti jsou požadavky na snadné a rychle určení, přesné nebo přibližné, geografické polohy velmi vysoké. To vše díky velké rozšířenosti mobilních zařízení, jako jsou chytré mobilní telefony a tablety využívající aplikace, které plní uživatelské požadavky v závislosti na aktuální geografické poloze. Existují různé případy a rozdílné požadavky na přesnost geografické informace, kde přesnost může být v jednotkách metrů, ale i stovkách metrů až kilometrů.

1.1 Geolokace síťových zařízení

Metody geolokace síťových zařízení lze aplikovat u mobilních zařízení, která jsou připojena k internetové síti. Využívá se informací o připojeném zařízení a to konkrétně informace o IP adrese zařízení. Jedná se o tzv. IP geolokaci a existují dva přístupy realizace – aktivní a pasivní. Společnou výhodou je snadný přístup k polohové informaci pouze díky připojení k internetové síti. Naopak nevýhodou může být fakt, že výsledkem nemusí být přesná, ale pouze přibližná geografická souřadnice mobilního zařízení [1, 19].

1.1.1 Aktivní metody

Aktivní metody geolokace síťových zařízení nejčastěji provádějí měření zpoždění datového přenosu v síti. Na základě tohoto měření odhadují polohu stanice vzhledem ke skupině známých referenčních bodů, u kterých je možné určit jejich geografickou polohu [1].

Metoda GeoPing

Nejstarší geolokační metoda, která využívá měření zpoždění v síti. Při použití vyžaduje velké množství pasivních referenčních bodů se známou geografickou polohou a několik aktivních zařízení provádějících měření. Principem metody je porovnávání vektorů zpoždění příslušících k jednotlivým referenčním bodům a lokalizované stanici. Výsledná poloha je

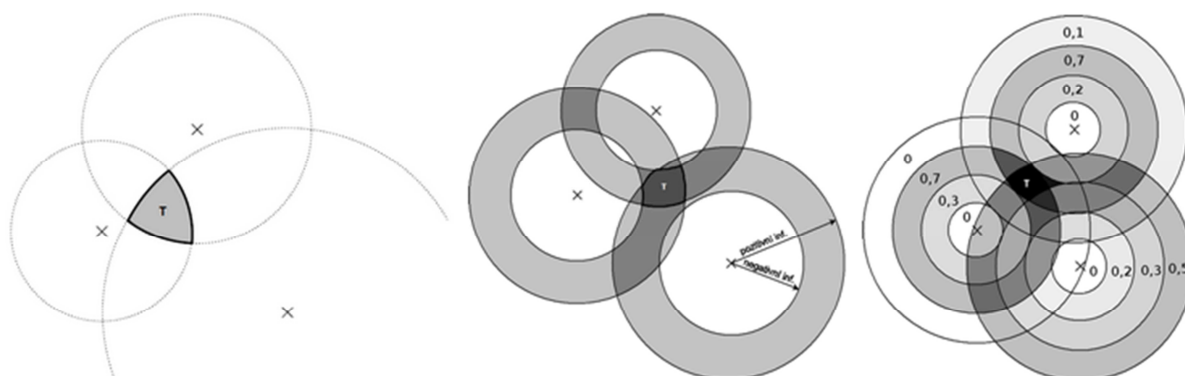
zaměření polohy jednoho z referenčních bodů. Metoda tedy nezaměří polohu mobilního zařízení [19].

Metoda ShortestPing

Jedná se o nejjednodušší metodu využívající měření zpoždění a síť referenčních bodů. Principem je zaměření polohy referenčního bodu, který je k připojenému zařízení nejbliž. Nejmenší vzdálenost se určí podle nejmenší hodnoty zpoždění. Nevýhodou je opět zaměření referenčního bodu a ne samotného zařízení [1].

Metoda CBG

Metoda CBG, tedy Constraint-Based Geolocation, je založena na měření zpoždění s co největším počtem referenčních bodů a principu trilaterace¹. Ze vztahu mezi změřeným zpožděním a geografickou vzdáleností k referenčnímu bodu, je vytvořena hranice nevzdálenějšího umístění stanice. V průsečíku vzniklých kruhových polí se pak pravděpodobně nachází lokalizované zařízení. Oproti předchozím metodám je výsledek přesnější zaměřením polohy samotného zařízení a ne referenčního bodu [1].



Obrázek 1 Metody CBG, Octan, GeoWeight

(zdroj: <http://access.feld.cvut.cz/view.php?cisloclanku=2012070001>)

Z principů metody CBG vychází několik dalších metod:

- *Speed of Internet (SOI)* – zjednodušený výpočet a snížení zatížení sítě za cenu menší přesnosti.
- *Metoda Octan* – kromě oblasti předpokládaného výskytu stanice určuje i oblast, kde se lokalizovaná stanice nacházet nemůže.

¹ Měření vzdálenosti k několika známým bodům.

- *Metoda GeoWeight* – vytvoření pásem vzdáleností a k nim přiřazené naměřené hodnoty zpoždění. Je vypočítaná hodnota pravděpodobnosti a lokalizovaná stanice se nachází v místě s největší pravděpodobností.

Metoda	průměrná chyba	medián chyby
GeoPing	270 km	203 km
ShortestPing	259 km	293 km
CBG	227 km	174 km
SOI	334 km	285 km
Octant	153 km	87 km
GeoWeight	–	44 km

Tabulka 1 Srovnání přesnosti aktivních metod

(zdroj: <http://access.feld.cvut.cz/view.php?cislocianku=2012070001>)

1.1.2 Pasivní metody

Jak název napovídá, pasivní metody oproti aktivním metodám neprovádějí žádné aktivní měření. Geografické údaje získávají zpracováním dostupných informací o síťových zařízeních, které jsou získávány ve formě záznamů z veřejných nebo komerčních databází. Metody se dělí podle typu využívané informace [1, 19].

Pomocí pasivních metod lze získat i poměrně přesné geografické údaje. Přesnost určení polohy je ale závislá na dostupnosti dobrých informací v externí databázi. Záznamy musí obsahovat kvalitní informace o infrastruktuře sítě, v některých případech i infrastrukturách okolních sítí poblíž mobilního zařízení.

Geolokace pomocí IP

Geografické údaje jsou získány porovnáním záznamů IP adres v databázích organizace IANA, která spravuje přidělování IP adres. Existuje několik druhů databází, veřejných a neveřejných, které se mohou v poskytovaných informacích značně lišit. Mezi nejznámější databáze patří:

- *WHOIS* – nejčastěji využívaná veřejná databáze obsahující záznamy o majitelích internetových domén a IP adres (kontaktní údaje, správce atp.),
- *GEOIP* – obsahuje záznamy IP adres, které obsahují různé množství informací, takže v některých případech zaznamenává přesný záznam polohy a geografické souřadnice, ale v častých případech obsahuje pouze zemi původu.

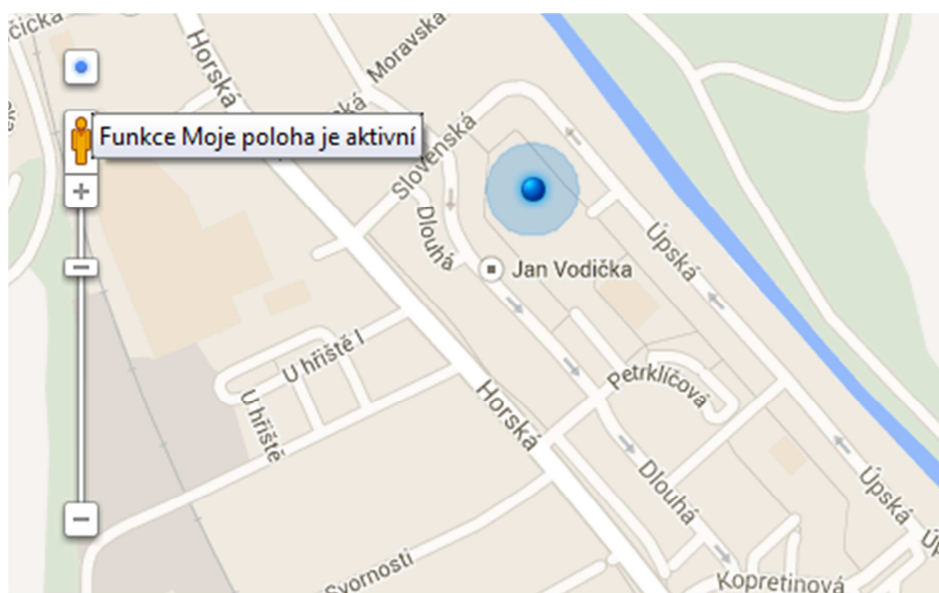
Protože IP adresa síťového zařízení nebo bran, přes které je zařízení připojené, je dostupná velmi snadno, patří tato metoda mezi nejjednodušší. Nevýhodou je závislost na rozdílných databázích, které mohou poskytovat jiný druh informací, mohou se lišit, v přesnosti poskytované informace, a množství poskytovaných informací [19].

Geolokace pomocí DNS

Princip je stejný jako využití IP adresy. Polohu zařízení lze učit reverzními záznamy DNS, tedy přeložením IP adresy na doménové jméno. Oproti využití IP adresy je záznam DNS uživatelsky srozumitelnější a zapamatovatelnější. Struktura DNS záznamů je tříděna hierarchicky, takže nejčastějším výsledkem je určení země původu [19].

Geolokace pomocí WiFi

V případě, kdy je zařízení připojeno prostřednictvím bezdrátové sítě WiFi, je možné získat geografické údaje na základě poloh okolních přístupových bodů. V průběhu lokalizace je načten seznam informací o dostupných přístupových bodech (SSID, MAC, síla signálu), v okolí stanice. Ten je porovnán s externí databází obsahující záznamy o bezdrátových sítích. Na základě nálezu přístupových bodů v databázi a síly dostupného signálu je určena předpokládaná pozice stanice [2, 19].



Obrázek 2 Geolokace pomocí WiFi Google Maps

(zdroj: Google)

Jednu z největších databází přístupových bodů bezdrátových sítí vlastní společnost Google, která záznamy o bezdrátových sítích získala při tvorbě svých map. Velká výhoda metody je

využití pouhého webového prohlížeče k získání poměrně přesných výsledků. Požadavkem metody je dobrá infrastruktura bezdrátových sítí, která zaručuje přesnější výsledky. Metoda nemusí být všude dostupná z důvodu chybějících záznamů o infrastruktuře [2].

Aktivní metody	
+ Nezávislost na manuálně spravovaných databázích. + Možnost využití ve webovém prohlížeči.	– Nedosahují přesnosti pasivních metod. – Vyšší zatížení sítě. – Závislost na dobré infrastruktuře sítě.
Pasivní metody	
+ Vyšší přesnost, zvláště pak WiFi geolokace. + Možnost využití ve webovém prohlížeči.	– Závislost na manuálně spravovaných databázích. – Závislost na dobré infrastruktuře sítě.

Tabulka 2 Srovnání síťových metod geolokace

(zdroj: autor)

1.2 Využití elektromagnetických signálů

Metody využívají vnějšího zdroje signálu, kterým jsou vysílače na povrchu země nebo v blízkém vesmíru. Metody pracují převážně na principu trilaterace, tedy dálkoměrných systémů. Přijímač signálu určuje svojí vzdálenost k několika záměrným bodům. Geografická poloha je výsledkem protnutí zaměřených vzdáleností mezi přijímačem a danými body (vysílači signálu). Přijímač může počítat z dostupných informací různé kombinace a výsledek je průměrem vypočtených kombinací. Výsledkem by měla být přesná souřadnice místa, kde se přijímač nachází [6, 14].

1.2.1 Družicové systémy GNSS

Systémy GNSS, tedy Global Navigation Satellite System, jsou radionavigační družicové systémy, které umožňují zaměření přesné informace o zeměpisné souřadnici přijímače, rychlosti a čase, kdekoliv a kdykoliv na zemském povrchu nebo v jeho blízkosti. Nejrozšířenějším systémem je americký systém Navstar GPS. Systém GPS, tedy Global Positioning System, byl vytvořen k vojenskému využití a je provozován ministerstvem obrany USA. Využívání GPS v civilním sektoru není nijak omezeno, ale systém je stále primárně vojenský. Příjem signálu GPS je v současné době možný ve většině využívaných mobilních zařízení [6, 14].

Na stejném principu jsou vyvíjeny další dva globální družicové systémy, ruský Glonass a evropský navigační systém Galileo. Systém Glonass je také vojenský systém, vyvíjený Ruskou federací. V současnosti není tak rozšířený jako systém GPS, ale je stejně využitelný i pro civilní uživatele. Systém je často využíván spolu s příjmem signálu GPS a umožňují tak vzájemné zpřesňování výsledků [5, 6, 14,].

Systém Galileo je první nevojenský autonomní systém, na jehož vzniku se podílejí členské státy Evropské unie. Cílem projektu je osamostatnění od vojenských systémů a získání přesnějších výsledků a to hlavně pro civilní využití. Systém Galileo by díky velkému počtu družic a jejich rozmístění v kosmickém segmentu, měl dosahovat daleko přesnějších výsledků. V praxi by mělo být dosaženo přesnosti až jeden metr. V současné době je systém ve fázi budování a využitelný by měl být od roku 2014. Plný provoz systému je pak v plánu na rok 2019 [6, 15]

Princip určování polohy

Družicové polohové systémy jsou budovány jako pasivní dálkoměrné systémy. Poloha přijímače je výsledkem protnutí zaměřených vzdáleností mezi přijímačem a družicemi viz obrázek 3. Pasivní systém v tomto případě znamená, že signál je přijímačem opravdu pouze přijímán a sám k družici žádný signál nevysílá [6, 14].



Obrázek 3 Družicový dálkoměrný systém GPS

(zdroj: Vojenský geografický a hydrometeorologický úřad)

Určení vzdálenosti družic od přijímače lze provádět třemi způsoby. Na základě kódových měření (tj. využívání dálkoměrných kódů), fázových měření (tj. využití nosné vlny signálu

a počítáním celých vln potřebných k dosažení cíle) nebo dopplerovských měření, které je založeno na fyzikálním principu Dopplerova jevu [6].

Každý signál vysílaný družicí GPS je kombinací nosné vlny, dálkoměrného kódu a navigační zprávy. Z navigačních zpráv je přijímač schopen identifikovat družici. Z její pevně stanované oběžné dráhy dokáže přijímač vypočítat její pozici v každém okamžiku, hlavně v okamžiku vysílání dálkoměrného kódu [6, 15].

Přesnost

Přesnost přijímače závisí na použitém typu přístroje, metodě měření, ale i místě, času a klimatických podmínkách v době lokalizace. Přesnost u nejlepších geodetických přijímačů je, v ideálních podmínkách, uváděna i v jednotkách milimetrů. U běžných přijímačů je pak dosahovaná přesnost přibližně 5 - 15m [6, 15].

Omezení

Základním omezením systému GPS je obecně dostupnost signálu družic. Vzhledem k velké vzdálenosti vysílačů není intenzita signálu příliš velká. Z toho vyplývá omezení v místech, kde není zachytitelný slabý signál (bez přímého výhledu na oblohu) např. budovy, členitý terén s hustou vegetací. Kromě dostupnosti hraje svou roli i postavení družic. Dostupnost lze matematicky ohodnotit.

Samotné přijímače nejsou schopné z pouhého přijetí signálu určit orientaci přístroje. K tomu, aby mohl být určen směr dalšího postupu, je potřeba, aby se pohyboval nebo využil magnetického kompasu [6].

1.2.2 Systém pozemních stanic

Vedle v současnosti nejčastěji používané metody GPS existuje také možnost využití pozemních vysílačů radiových signálů. Příkladem může být využití geolokace pomocí vysílačů mobilního telefonního signálu. Využívá se známé pozice základových stanic mobilních sítí GSM² a stanic mobilní sítě UTMS³. U sítí GSM jsou to stanice BTS, tedy Base Transceiver Station, a u sítí UTMS jsou to stanice Node-B [10, 16].

² Groupe Spécial Mobile – Globální systém pro mobilní komunikaci

³ Universal Mobile Telecommunication System

Princip určování polohy

Pozemní základové stanice tvoří síť a rozdělují pokrývané území na buňky. U každé stanice známe její geografickou polohu a každá buňka je identifikována svým *Cell ID*. Pro zaměření polohy tedy stačí identifikovat buňku, ve které se mobilní zařízení nachází. Pro zpřesnění lze využít ještě časového předstihu vysílání a rychlosti šíření radiového signálu (Timing advance – *TA*). Díky tomu lze určit vzdálenosti od základových stanic. Při příjmu signálu z více stanic je přesnost ještě větší [16].

Vyšší přesností dosahující následující metody, které využívají pro zaměření vždy více základových stanic a výsledek je průnik jednotlivých měření.

- *Metoda E-OTD* – využívá měření rozdílů příchodu signálů od tří a více stanic. Základním předpokladem je zajištění synchronizace základových stanic v síti, což se v praxi často nedodrжуje. Síť je vybavena zařízením na měření reálných časových rozdílů.
- *Metoda AOA* – základem metody je využití směrových antén, díky čemuž lze měřit úhly, pod kterým je signál přijímán. Výsledkem metody je pak protnutí přímek, které procházejí stanicí a mobilním zařízením. Čím více stanic je dostupných, tím je výsledek přesnější.
- *Metoda E-CGI* – je zlepšením metody Cell ID a TA o měření úrovní signálů. Využívá se znalosti vysílacích výkonů základových stanic a modelu šíření signálu.

Vysílaný signál

Při geolokaci se využívá radiového signálu, který není primárně určen pro zjišťování geografických informací. V tomto případě se využívá síť pro mobilní komunikaci a zařízení musí být vybaveno modulem pro příjem mobilního signálu, například GSM nebo UTM. S.

Přesnost

V závislosti na využívané metodě lze dosahovat různých úrovní přesnosti. Od získání oblastní polohové informace mohou jednotlivé metody dosahovat přesnosti ve stovkách, ale i desítkách metrů. V případě využívání jsou vhodné jako doplňující informace pro navigační systém GPS a urychlit tak počáteční inicializaci zařízení [16].

Metoda	Přesnost – město	Přesnost – okolí
Cell ID + TA	Od 100m do 500m	Jednotky až desítky kilometrů
E-OTD	Lze dosáhnout až desítek metrů	Stovky až tisíce metrů
AOA	Lze dosáhnout až desítek metrů	Od 300m po jednotky kilometrů
E-CGI	Od 50m do 550m	Od 250m do 8km

Tabulka 3 Srovnání přesnosti metod geolokace u GSM sítí

(zdroj: <http://access.feld.cvut.cz/view.php?cisloclanku=2006022801>)

Omezení

Nutností je připojení k mobilní síti, takže zařízení musí být vybaveno potřebnou technologií. U přesnějších metod ovlivňuje dosaženou přesnost dostupnost signálu, tj. například v místech s menší hustotou zástavby a infrastruktury stanic [16].

1.2.3 Lokální bezdrátové sítě

Mezi využitelné technologie lze také zařadit technologie spadající do skupiny bezdrátových sítí. Lze využívat bezdrátové technologie sítí PAN⁴, kde mezi nejčastěji využívanými lokálními sítěmi jsou technologie Bluetooth, ZigBee atp. Samozřejmostí je využití technologie WiFi sítí.

Principem geolokace je využití matematické metody triangulace síly signálu. Díky tomu lze zmíněné technologie využít pro určení geografických informací. Jak název napovídá, jedná se o lokální určování polohy, tedy na relativně malé ploše, tj. budovy, výrobní haly atp. Přesnost je dána dobrým pokrytím a dobrou infrastrukturou sítě [10].

⁴ Personal Area Networks

GNSS	
+ Globální dostupnost. + Vysoká přesnost lokace. + Nezatěžuje síťový přenos.	– Nutnost LOS ⁵ . – Delší doba inicializace. – Vyšší spotřeba energie.
BTS apod.	
+ Dobrá pokrytí. + Funkčnost při NLOS ⁶ . + Nezatěžuje síťový přenos. + Krátká doba inicializace.	– Nepřesná lokace.
Lokální bezdrátové sítě	
+ Přesná lokace. + Funkčnost při NLOS. + Krátká doba inicializace.	– Nutná velmi dobrá infrastruktura. – Zatížení síťového přenosu. – Lokální funkčnost.

Tabulka 4 Srovnání metod využívající el. mag. Signály

(zdroj: <http://www.internetprovsechny.cz/webove-aplikace-s-geolokaci/>)

1.3 Přímé metody určení pozice

Při přímém postupu zjišťování aktuální zeměpisné souřadnice je využíváno prostředků, které přímo měří veličiny pomocí který lze určit aktuální polohu k předem známé poloze. Tyto prostředky mohou vykazovat nahodilé chyby, případně může být měření závislé na individuálním přístupu k měření, například lidský faktor. To znamená, že získaná souřadnice může být při opakování postupu jiná [3, 4].

1.3.1 Inerciální navigační systém

Inerciální navigační systém slouží k měření okamžité zeměpisné souřadnice. Dokáže určit aktuální polohu pohybujícího se objektu vzhledem k jeho známé počáteční poloze. Navigace je plně autonomní, tedy nezávislá na vnějším zdroji informace nebo signálu. To je v prostředích, kde není možné přijímat žádný signál.

Inerciálních systémů se využívá zejména v letectví. Zde inerciální navigační systém slouží kromě poskytování navigačních parametrů (zeměpisné souřadnice, rychlost a zrychlení pohybu), také k určení kurzu, náklonu a sklonu a k učení uhlových rychlostí vzhledem k osám pohyblivého prostředku. Tím v sobě zahrnuje všechny parametry potřebné pro bezpilotní režim letu [3].

⁵ Line Of Sight – nutnost přímé viditelnosti na vysílač

⁶ Non Line Of Sight – bez nutnosti přímé viditelnosti na vysílač

Princip určování polohy

Základním principem inerciální navigace je postupná dvojitá časová integrace signálu zrychlení a tím získání signálu dráhy viz rovnice 1, kde $s(t)$ je okamžitá hodnota naměřeného vektoru trajektorie pohybu, $a(t)$ je okamžitá hodnota měřeného vektoru zrychlení, g je vektor tíhového zrychlení, v_0 je počáteční rychlost pohybu a s_0 je počáteční hodnota trajektorie.

$$s(t) = \int_0^t \left\{ \int_0^t [a(t) - g] dt + v_0 \right\} dt + s_0$$

Rovnice 1 Výpočet dráhy

Pokud tedy máme k dispozici informace o akceleraci a rotaci v jednotlivých osách v čase, můžeme vypočítat novou pozici vzhledem k počáteční známé pozici. K zjišťování těchto informací se využívá inerciálních senzorů. Nejčastější konstrukce inerciální navigace se skládá ze soustavy gyroskopů nebo akcelerometrů.

Konstrukčním základem je tříosá gyroskopicky stabilizovaná základna nesoucí dvouosou (případně tříosou) soustavu akcelerometrů a soustavu podpůrných komponentů. Citlivostní osy akcelerometrů jsou orientovány tak, aby zaujímaly směry světových stran a byly navíc kolmé na směr místní vertikály. Při průmětu složky tíhového zrychlení do citlivých os akcelerometrů indikuje rychlost a dráhu [3, 4].

Přesnost

Přesnost měření inerciální navigace je ovlivněna konstrukcí a použitými senzory. Při měření informací o akceleraci a rotaci vznikají odchylky, které se s rostoucím časem výrazně zvětšují. Nejlepších výsledků dosahují prostředky využívané v letectví. V současné době probíhá velký vývoj inerciálních MEMS senzorů, které by mohly nahradit používané technologie [3].

Omezení

Velkým omezením inerciální navigace je především vysoká cena a velikost samotného zařízení. Zařízení vyžaduje dlouhou dobu ke své kalibraci a je prakticky nemožné využívat ho pro běžné úkony. Inerciální MEMS senzory, které jsou mnohem levnější, mají menší rozměry a menší spotřebu energie, jsou pro tento účel prozatím nepoužitelné pro svou velkou nepřesnost [3].

1.3.2 Výpočet aktuální pozice

Určení aktuální zeměpisné souřadnice je založeno na přímém měření směru a vzdálenosti. Postup lze využít, když je známa počáteční pozice a jsou známy informace o délce a směru dalšího pohybu. Díky tomu lze přibližně odhadnout aktuální pozici. Metoda je závislá na přesnosti měření a vzhledem k explorativnímu sběru dat a lidskému faktoru jsou výsledky pozice odhadem aktuální polohy [13].

Princip určování polohy

Prvním krokem je nastavení mobilního zařízení do výchozí pozice. Většinou za pomoci jiné metody geolokace. V dalším kroku je určen směr následujícího postupu, a to například vzhledem k zeměpisnému severu pomocí kompasu. Konečným krokem je změření vzdálenosti mezi výchozí pozicí a aktuální pozicí. Toho lze dosáhnout pomocí přesných elektronických zařízení (dálkoměry) nebo jednoduššími metodami (metr, měření času, krokoměr atp.) [13].

Přesnost

Metoda je založena na jednoduchém principu a závislá na metodách, které mohou podléhat nepřesnostem. Určování směru pomocí magnetického kompasu nemusí být dostupné nebo může být rušeno jinými nežádoucími vlivy. Stejně tak měření vzdálenosti podléhá přesnosti zařízení, kterým je vzdálenost měřena. Výhodou je částečná autonomnost metody a lze ji tedy využít v prostoru, kde není dostupný žádný signál.

Omezení

Vzhledem k nepřesnostem v měření není vhodné tuto metodu využívat pro získávání přesných souřadnic. Metoda je vhodná pro přibližné získávání výsledků. V případě užití primitivnějších metod, jako je například krokoměr, musíme počítat s individuálními výsledky pro jednotlivá měření.

Inerciální navigace	
+ Přesné výsledky. + Autonomní systém.	– Nutné kvalitní komponenty zařízení. – Drahé a nedostupné.
Výpočet aktuální polohy	
+ Usnadnění orientace.	– Pouze orientační výsledky. – Přesnost je individuální.

Tabulka 5 Srovnání přímých metod

(zdroj: autor)

1.4 Alternativní metody geolokace

K získání geografické informace mohou sloužit i orientační body, které jsou buď pro dané místo charakteristické, nebo jsou na daném místě k těmto účelům připravené.

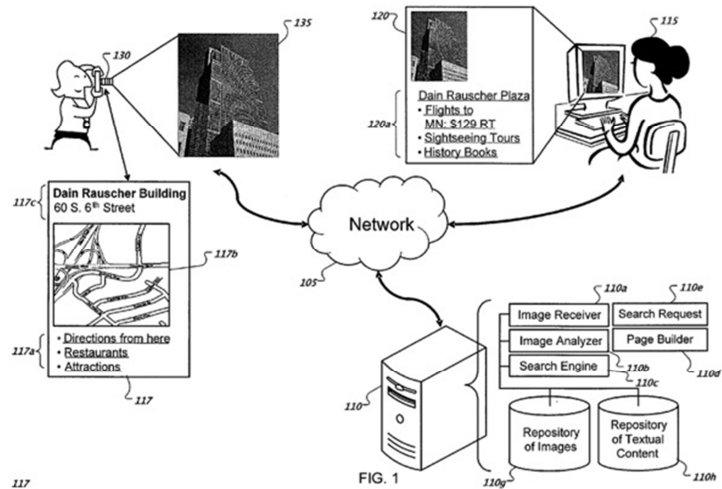
1.4.1 Využití QR – kodů

Technologie QR kódů (Quick Response Code) je v podstatě obdoba klasických čárových kódů. Na rozdíl od čarového kódu je složen z černobílých bloků, které se skládají do tvaru čtverce. Díky své konstrukci dokážou přenášet daleko více informací. Do jednoho QR kódu je možné uložit až 3 000 bajtů dat, tzn. do jedné QR matice se vejde až 7 000 číslic nebo text o délce 4 300 znaků.

V praxi je možné vytvořit pomocí QR kódu síť orientačních bodů. Každý QR kód v sobě ponese informaci o své geografické poloze. Mobilním zařízením, které je vybavené čtečkou QR kódů pak stačí kód vyfotit a překladem kódu informaci o poloze získat. Výhodou je využití offline čtečky kódů. Tím je možné získat informace bez jakéhokoli připojení. Nevýhodou je nutnost QR kód předem připravit. Další nevýhodou je, že většina QR kódů, které jsou k dispozici, často zpřístupňují webové stránky, kde je samotný obsah [21].

1.4.2 Využití fotografie

Společnost Google pracuje na geolokaci pomocí rozpoznávání známých objektů na fotografii. V praxi to znamená, že na základě pořízené fotografie, bude metoda schopna určit geografickou souřadnici místa, odkud byla fotografie pořízena. Problém je náročnost dané metody, a tak je lepší spolehnout se na ověřenou GPS nebo BTS.



Obrázek 4 Geolokace pomocí fotografie

(zdroj: <http://www.androidmarket.cz/>)

QR kódy	
<ul style="list-style-type: none"> + Přesné. + Možnost získání většího množství informací. 	<ul style="list-style-type: none"> – Závislost na vytvořené síti QR kódů.
Foto geolokace	
	<ul style="list-style-type: none"> – Náročný výpočet. – Závislost na externí databázi. – Časová náročnost. – Není zaručena přesnost.

Tabulka 6 Srovnání alternativních metod

(zdroj: autor)

2 Využití geolokace

Geolokace se v mobilních zařízeních využívá v širokém množství aplikací a služeb nativních i webových. Takové aplikace a služby, které jsou založeny na využití geolokace jsou souhrnně nazývány Lokálně Kontextové Služby, tedy Local Based Services [9].

2.1 Lokálně kontextové služby

Lokálně kontextové služby zkráceně LBS, zpřístupňují svůj obsah v závislosti na aktuální geografické poloze uživatele. Díky znalosti geografické polohy, ať už přesné nebo přibližné, mohou uživateli poskytovat pouze ty služby nebo informace, které jsou spjaté s jeho aktuální polohou. Toho se v současnosti velmi využívá v následujících odvětvích [9].

Reklamní služby

Vzhledem k velkému využívání Internetu v mobilních zařízeních je možné s využitím geolokace zajistit velmi dobré šíření reklamy. Díky známé geografické poloze mobilního zařízení lze zvýšit přesnost zaměření reklamy na správný okruh uživatelů, například nabízet reklamní informace pouze v konkrétní oblasti.

Pro tento typ služeb není nutné znát přesnou geografickou polohu. Pro šíření reklamy stačí znát údaje o oblasti, kde se mobilní zařízení nachází, například město, obec atp. Vzhledem k tomu stačí využít metody, které jsou snadno dostupné a to i z webového prohlížeče, například metody založené na síťovém připojení [9, 19].

Informační služby

Výsledek geolokace může sloužit i k získání aktuálních informací spjatých s polohou mobilního zařízení. Lze tak snadno získávat užitečné informace, například o dopravním spojení, o počasí v dané lokalitě, nebo dopravní situaci. Takové informace jsou často využívány i u jednoúčelových zařízení, například GPS navigace v automobilu.

Požadavek na přesnost geografické polohy je oproti reklamním službám daleko vyšší. V některých případech je vyžadována co nejpřesnější informace o poloze mobilního zařízení, např. při hledání autobusových spojení z konkrétní autobusové zastávky. V takových případech jsou často instalovány nativní aplikace a lze tak využívat všechny metody. V některých případech se nabízí i možnost využití QR kódů [9, 19].

Bezpečnost

Geografickou informaci lze velmi dobře využít z hlediska bezpečnostních služeb. Určitý typ činností lze spojit s určitým typem geografické informace a na základě toho ověřit bezpečnostní riziko. To například platí při ochraně zneužití citlivých dat nebo pro monitorování činnosti. U ochrany citlivých dat, například kreditních karet, přihlašovacích údajů nebo uživatelských účtů atp. se může využít informace o místech obvyklého použití a v případě vychýlení od standardu je provedeno bezpečnostní opatření. Lze také efektivněji bojovat proti spamu, kde je možné lokalizovat nebezpečné proudy dat a ty blokovat.

Přesnost geolokace nemusí být u bezpečnostních služeb vázána ke konkrétní geografické poloze, ale k rozsáhlejší oblasti. V některých případech stačí určit území státu. Při monitorování už je požadavek na přesnost daleko větší [9, 19].

Zábava

Geografické informace jsou využívány v různých geo-sociálních sítích, které geolokaci využívají k zábavě. Příkladem může být aplikace Foursquare, jejíž podstatou je záznam aktivit a míst, kde uživatel byl. V dalších případech může aplikace zpřístupňovat informaci, kde se nachází aktuálně připojený uživatel.

V této kategorii mohou být aplikace, které vyžadují přesné určení geografické polohy, ale také aplikace, které nejsou z hlediska přesnosti tak náročné. Může se jednat o nativní aplikace, které využívají přesnou lokaci pomocí GPS, ale také o webové aplikace, které používají méně přesné metody [9, 19].

2.2 Využití geolokace ve webových aplikacích

Mobilní webová aplikace je jako multiplatformní aplikace, která není vyvíjena pro konkrétní platformu. Z hlediska vývoje, snadné údržby a aktualizaci zdrojových kódů jsou webové aplikace výhodným řešením při tvorbě mobilní aplikace. Využívání geolokace webovou aplikací, ale není tak snadné jako u nativních aplikací.

Samotná webová aplikace nemá přístup k systémovým službám a hardwaru mobilního zařízení stejně jako nativní aplikace. Z toho důvodu není možné využití některých metod geolokace přes webový prohlížeč. K účelu geolokace ve webových aplikacích jsou

konsorciem W3C vydány standardy pro API⁷ specifikace, které webové aplikaci umožní přístup k potřebným funkcionalitám a umožní tak přístup k informaci o geografické poloze zařízení [9].

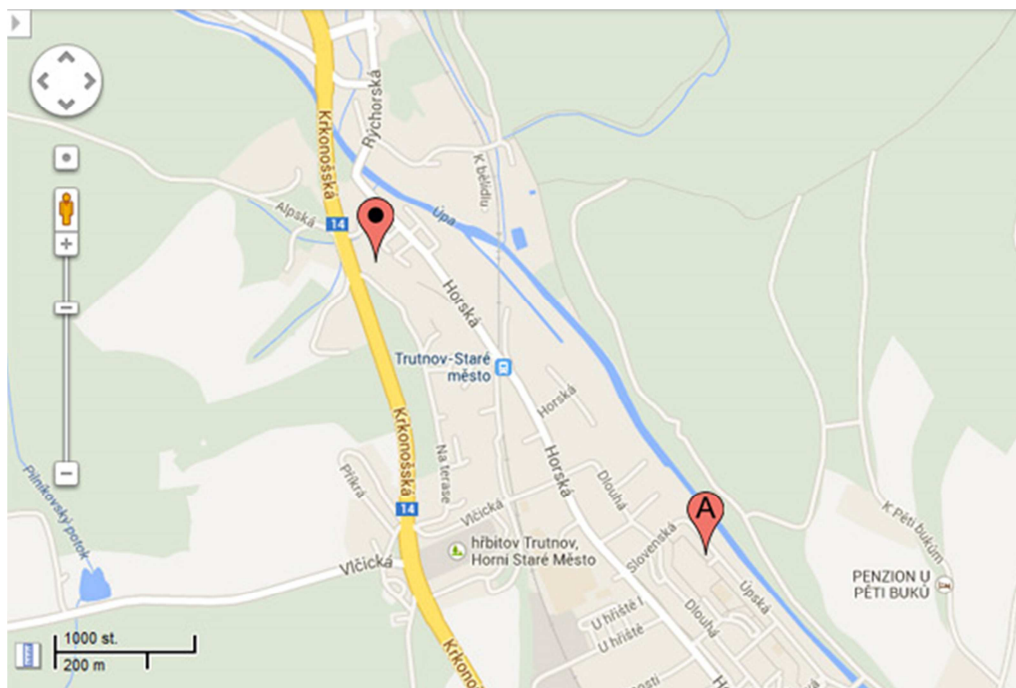
2.2.1 Geolocation API

Umožňuje webové aplikaci využívat geolokaci a získat údaje o geografické poloze mobilního zařízení. Specifikace neurčuje jakou metodu geolokace budou údaje o poloze získány. Záleží na možnostech mobilního zařízení a na dostupnosti technologií. V případě, kdy není dostupná GPS geolokace, využívá se geolokace síťových zařízení. Pokud je k zařízení připojeno pomocí WiFi, je možné dosáhnout velmi přesného výsledku. Pokud není dostupné WiFi připojení nebo jiné připojení k internetu, budou údaje získány z mobilní sítě. Prohlížeč zjistí dostupnost okolních bezdrátových sítí nebo vysílačů mobilní sítě a s požadavkem geolokace využije například Google.

Přesnost geolokace je různá pro každou využitou technologii. Následující obrázky ukazují rozdíly v přesnosti geolokace ve webovém prohlížeči mobilního telefonu. Na každém obrázku je zobrazen bod A, který určuje skutečnou polohu mobilního zařízení, a bod, který je získán použitím různých technologií.

Na obrázku 5 je výsledek získán prostřednictvím mobilní sítě GSM. Výsledná pozice je získána zaměřením stanic BTS. Webový prohlížeč získá geografické údaje na základě vytvoření balíčku ve formátu JSON, který obsahuje informace o dostupných základových stanicích BTS a síle dostupného signálu. Balíček je odeslán na adresu příslušného serveru, v tomto případě serveru Googlu, který na základě získaných informací porovná data se svou databází známých poloh stanic a výpočtem triangulace síly signálu určí přibližnou polohu mobilního zařízení [2, 10]

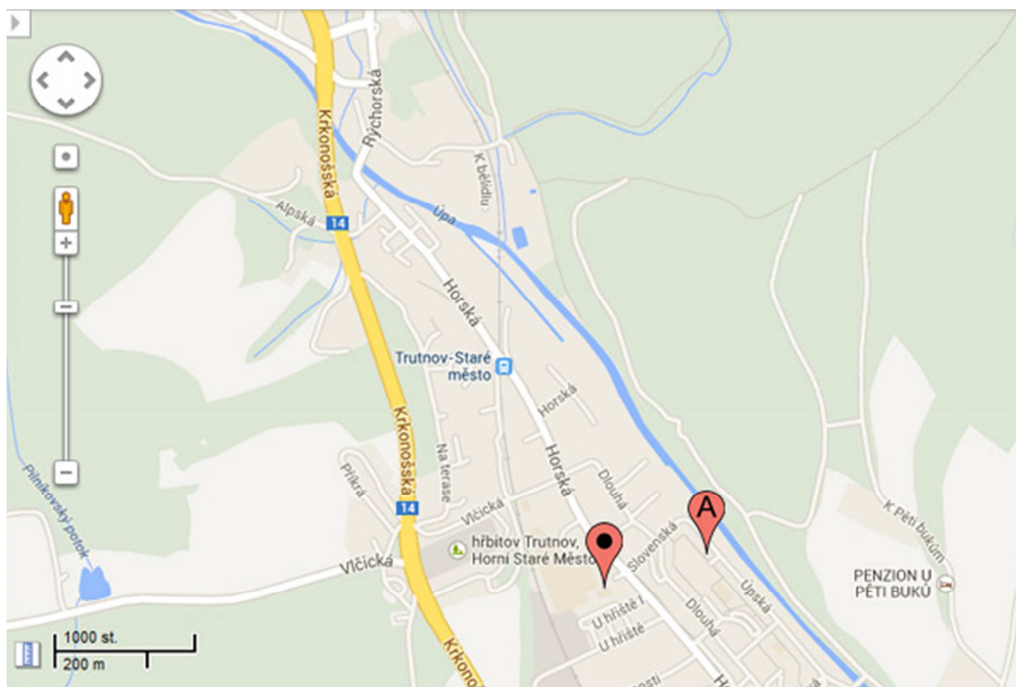
⁷ Application Programming Interface



Obrázek 5 Geolokace pomocí BTS

(zdroj: <http://robertnyman.com/html5/geolocation/current-location.html>)

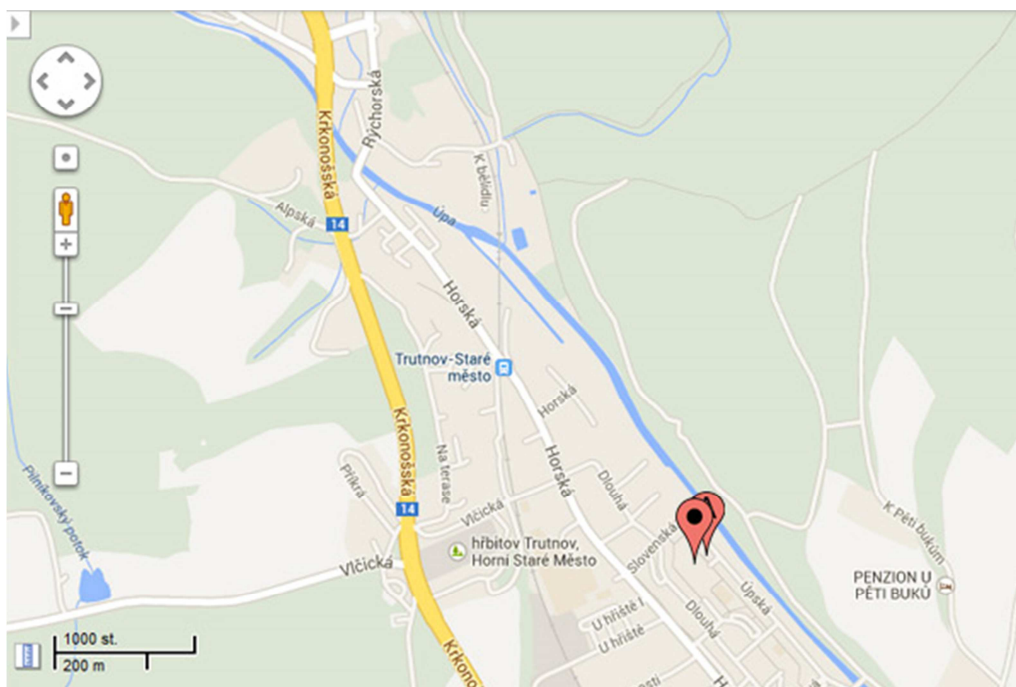
Obrázek 6 ukazuje přesnost výsledku po připojení mobilního zařízení v mobilní síti WCDMA. Výsledná pozice je získána pomocí stanic Node-B. Postup geolokace je stejný jako u stanic BTS.



Obrázek 6 Geolokace pomocí Node B

(zdroj: <http://robertnyman.com/html5/geolocation/current-location.html>)

Na posledním obrázku 6 je výsledná pozice získána po připojení mobilního zařízení k dostupné síti WiFi. Zde se místo seznamu dostupných základových stanic vytvoří balíček se seznamem dostupných WiFi sítí a síly dostupného signálu.



Obrázek 7 Geolokace pomocí sítě WiFi

(zdroj: <http://robertnyman.com/html5/geolocation/current-location.html>)

Nejpřesnějšího výsledku bylo dosaženo po připojení mobilního zařízení k síti WiFi. Při připojení k mobilní síti dostáhla lepšího výsledku síť WCDMA a síť GSM byla nejméně přesná. Přesnějších výsledků by mohlo být dosaženo v případě, kdy bude zařízení připojeno k více sítím najednou. Nejpřesnější budou výsledky v případech, kdy se mobilní zařízení bude nacházet například v centru města. Naopak méně přesné výsledky budou pravděpodobně získány mimo město [2, 10].

2.3 Využití geolokace nativních aplikací

Nativní aplikace jsou vyvíjeny pro konkrétní platformu mobilního zařízení. To znamená, že nevýhodou nativních aplikací je nutnost vývoje aplikace pro různé platformy zvlášť. Proti nenativním aplikacím, ale mají výhodu v plném přístupu k systémovým službám a hardwaru mobilního zařízení.

Díky tomu mohou nativní aplikace bez problému využívat geolokaci například pomocí modulu GPS, který dosahuje velmi přesných výsledků. Pro nevýhody této geolokace,

především pak pomalá inicializace, se ale i v nativních aplikacích využívají ostatní metody. Metody lze vhodně kombinovat a zpřesňovat tak dosažené výsledky. U zařízení, které lze připojit k mobilní síti, je možné zkombinovat informace získané prostřednictvím mobilní sítě se systémem GPS a vytvořit tak systém A-GPS, tedy Assisted-GPS. Samozřejmostí je i využívání informací získaných síťovým připojením [2, 10].

Například u zařízení s mobilní platformou Android, jsou při vytváření nativní aplikace k dispozici dva zdroje geolokace, tzv. provider.

- NETWORK_PROVIDER – zdroj, který umožňuje získat informace o geografickém připojení pomocí mobilní sítě, nebo využít jiné dostupné připojení. Princip je stejný jako u geolokace ve webovém prohlížeči.
- GPS_PROVIDER – zdroj využívající příjem signálu ze satelitů systému GPS.

Metody lze kombinovat získávat výsledky v závislosti na tom jak dobrý je přijímaný signál.

3 Mobilní zařízení

Mezi mobilní zařízení lze zařadit všechna přenosná bezdrátová zařízení, která jsou jednoúčelová nebo víceúčelová. Lokálně kontextové služby nacházejí největší uplatnění hlavně ve víceúčelových zařízeních, kterými jsou hlavně notebooky, tablety a chytré mobilní telefony. Využití LBS lze uplatnit i v jednoúčelových zařízeních, kterými jsou například GPS navigace a různá monitorovací zařízení. V rámci této diplomové práce jsou využívány pouze víceúčelové mobilní zařízení a to hlavně chytré mobilní telefony a tablety, vybavené potřebnými komponenty, které jsou v rámci práce využity [9, 10].

3.1 Mobilní platformy

Mobilní zařízení jsou nejčastěji vybaveny specifickými platformami, které jsou často charakteristické pro dané zařízení. Jednotlivé platformy mají různé přístupy k tvorbě nativní aplikace. V současnosti mezi nejznámější a nejrozšířenější mobilní platformy patří následující operační systémy.

Google Android

Nejrozšířenější a nejrychleji se vyvíjející mobilní operační systém vyvíjený společností Google. Jedná se o open source platformu, která je založena na Linuxu. Vzhledem k nulové ceně a velkému množství aplikací, je hojně využíván v mobilních zařízeních a v současnosti začíná být využíván i v jiných zařízeních. Aplikace pro tento operační systém jsou vyvíjeny pomocí Android SDK s využitím jazyka Java.

Apple iOS

Operační systém iOS je používaný výhradně mobilních zařízení Apple. Platforma vychází z operačního systému Mac OS X. Aplikace jsou vyvíjeny v jazyce C nebo Objective-C, ale systém je uzavřený a to s sebou přináší jistá omezení. Instalace aplikací je možná pouze přes oficiální App Store, kde lze najít pouze schválené aplikace. Neoficiální metodou je tzv. Jailbreak, který umožňuje nahrávat i neschválené aplikace.

Microsoft Windows Phone

Platforma od firmy Microsoft využívaná na různých mobilních zařízeních. Oproti předchozím platformám byl Windows Phone 6 v době vydání (r. 2007) nepokrokový a problémový. Ani upgrade systému nebyl dostačující a byl vydán koncepčně přepracovaný Windows Phone 7

(r. 2010). Aplikace jsou vyvíjeny prostřednictvím vývojářské platformy Microsoft XNA, který umožňuje snadný vývoj.

3.2 Komponenty mobilních zařízení

Většina moderních mobilních zařízení disponuje vedle prostředků zpřístupňujících geografické údaje i množstvím čidel a senzorů, které mohou sloužit k různým účelům. Mohou sloužit k samotné obsluze mobilního zařízení, ale také například k měření pohybu, orientace atp. V rámci diplomové práce budou využívány senzory, které jsou v dnešní době již běžnou součástí mobilních zařízení.

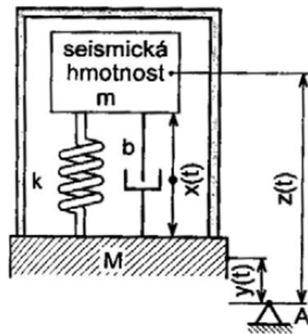
3.2.1 Akcelerometr

Je jednou ze základních součástí mobilních zařízení. Akcelerometr je senzor sloužící k určování statického zrychlení, tedy síly vzniklé působením zemské gravitace nebo dynamického zrychlení, síla vzniklá změnou rychlosti pohybujícího se předmětu (senzoru). Vzhledem k tomu je akcelerometr využíván ve velkém množství odvětví a lze s jeho pomocí řešit například [4]:

- a) měření úhlu náklonu,
- b) měření setrvační síly (měření rychlosti, vzdálenosti nebo síly),
- c) měření nárazů a vibrací,
- d) monitorování pohybu předmětů,
- e) ovládání multimediálních zařízení,
- f) inerciální navigační systém atp.

Princip akcelerometru

Základní princip akcelerometru je znázorněn na obrázku. Senzor je tvořen základnou M , která je pevně spojena s objektem. K základně je pružinou k , připojena setrvačná hmota o hmotnosti m , jejíž odchylka vzhledem k základně je vyhodnocována. Celá soustava je popsána rovnováhou setrvačné, direktivní a tlumící síly, tj. pohybovou rovnicí [4].



Obrázek 8 Základní princip akcelerometru

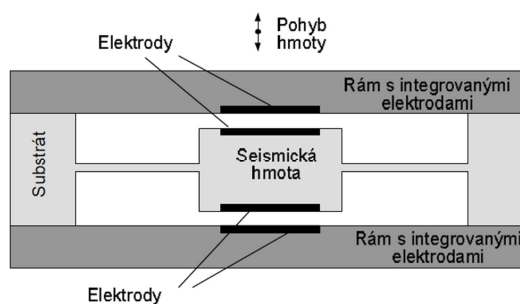
(zdroj: Sensory a měřící obvody, Ďad' o S.)

Existuje několik konstrukčních řešení akcelerometru a od toho se odvíjejících principů měření. Typy akcelerometrů:

- a) Piezoelektrické,
- b) Piezodoporové (Tenzometrické),
- c) Tepelné,
- d) Kapacitní.

V případě mobilního zařízení, kde je požadavek na minimální velikost senzoru, se využívá kapacitních akcelerometrů, konstruovaných plošnou mikromechanickou technologií, tj. MEMS technologií.

MEMS akcelerometr je založen na proměnné kapacitě tříelektrodového vzduchového kondenzátoru. Využívá se zde známé nelineární závislosti kapacity na vzdálenosti elektrod kondenzátoru. Jedna elektroda bude tvořena seismickou hmotou a její pohyb bude závislý na působícím zrychlení, viz obrázek X. Měřením kapacity získáme potřebné údaje. Součástky konstruované technologií MEMS představují moderní generaci levných senzorů [4].



Obrázek 9 Kapacitní akcelerometr

(zdroj: přednáška ČVUT Prof. Ing. M. Husák, CSc.)

3.2.2 Magnetometr

Magnetometr je v současné době součástí většiny chytrých mobilních zařízení. Slouží k měření složky lokálního magnetického pole. To je tvořeno součtem geomagnetického pole a magnetického pole, které je tvořeno okolím a jeho parametry. Magnetometr lze využít při konstrukci elektronického kompasu, nebo ve spojení s jinými senzory může zajišťovat přesné informace o směru nezávisle na orientaci [4].

Princip magnetometru

Pro měření magnetického pole existují různé metody, konkrétně:

- a) indukční metody,
- b) silové metody,
- c) a metody závislé na změně materiálových vlastností.

Nejznámějšími a nejběžnějšími senzory jsou Hallovy sondy, které jsou závislé na změně materiálových vlastností. Nejjednodušším příkladem je snímací cívka, v níž se indukuje napětí dle Faradayova zákona indukce.

$$\oint_s \vec{E} \widehat{ds} = - \frac{d}{dt} \iint_s \vec{B} \overline{dS}$$

Rovnice 2 Faradayův zákon indukce

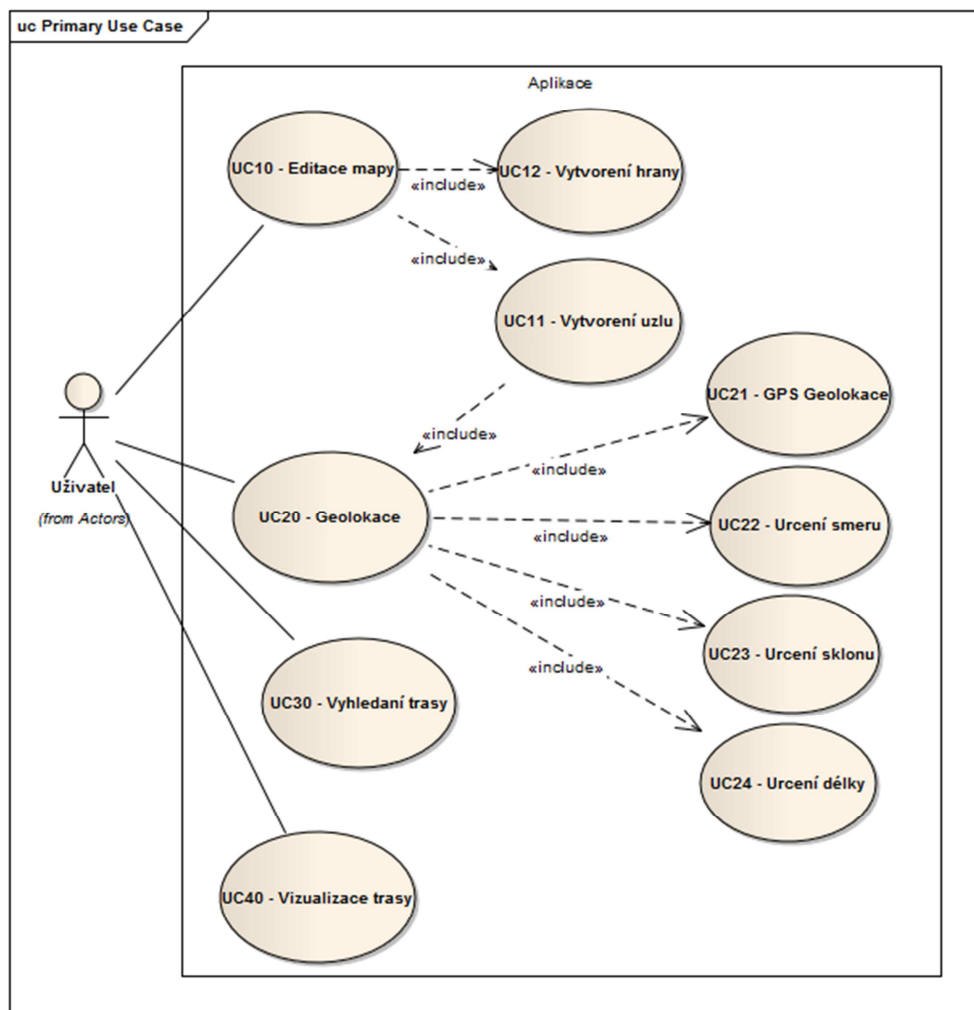
Napětí cívky odpovídající pravé straně rovnice je zpracováno v integračním zesilovači, jehož výstupní napětí odpovídá hledané hodnotě magnetického toku nebo indukci [4].

3.2.3 Modul GPS

Modul GPS, který je v současnosti také standardní komponentou v mobilních zařízeních, přijímá a zpracovává signál satelitního navigačního systému GPS, který je popsán v kapitole Možnosti geolokace. Modul předává informace o aktuální poloze konkrétnímu softwaru, který informace dále zpracovává, např. navigační software atp.

4 Návrh aplikace

Jedním z cílů diplomové práce je navrhnout mobilní nativní aplikaci využívající metody geolokace. Výsledkem je návrh mobilní aplikace umožňující na základě předem známé geografické polohy zaznamenávat postup a zjišťovat aktuální geografickou souřadnici. Záznam postupu by měl obsahovat změny směru a výskyt orientačních bodů. Na základě takto získaných informací by měla být aplikace schopna vizualizovat vykonané trasy a zmapovat postup v neznámém prostředí.



Obrázek 10 UseCase diagram navrhované aplikace

(zdroj: autor)

4.1 Funkční požadavky

Navrhovaná aplikace by měla obsahovat následující funkcionality.

Editace mapy:

- a) Vytvoření nové mapy, která bude obsahovat údaje o vykonaných trasách.
- b) Výběr uložené mapy k editaci nebo navigaci.
- c) Vytvoření bodové značky s aktuální geografickou souřadnicí.
- d) Vytvoření liniové značky s popisem a ohodnocením.

Geolokace:

- a) Umožnit uživateli přesné určení geografické souřadnice pomocí dostupné metody GPS.
- b) Určit aktuální směr vzhledem k magnetickému severu Země.
- c) Zpřístupnit přibližné údaje o vykonané trase.
- d) Určit novou geografickou polohu.

Navigace:

- a) Vyhledání nejkratší trasy.
- b) Znázornění přibližné vykonané vzdálenosti na vyhledané trase.

Vykreslení mapy:

- a) Vykreslení struktury mapy.
- b) Vykreslení informací o bodových a liniových značkách.

4.2 Návrh datových struktur

Základem navrhované aplikace bude uchovávání geografických dat v podobě mapy, která obsahuje bodové a liniové značky. Každá bodová značka bude charakterizována svou geografickou souřadnicí a seznamem liniových značek. Návrhem vhodné datové struktury, pro uchování těchto geografických dat, se umožní efektivní provádění operací s těmito daty.

4.2.1 Uchování geografických souřadnic

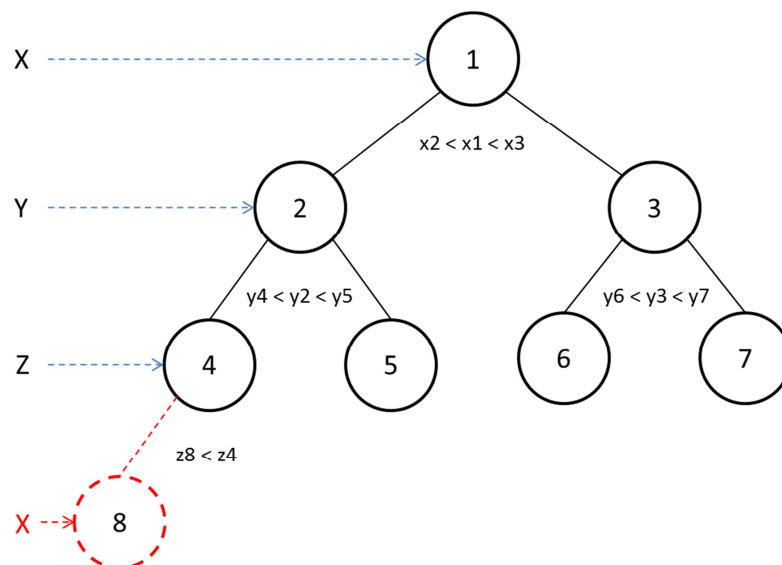
Každá bodová značka obsažená v mapě, bude reprezentována svou geografickou souřadnicí systému WGS84. Ta je složena z hodnoty zeměpisné šířky a hodnoty zeměpisné délky. Tyto

hodnoty budou označovány jako X a Y. Aby nenastala shoda dvou souřadnic v různých nadmořských výškách, je doplněna třetí hodnota Z, která slouží k uchování hodnoty nadmořské výšky v daném bodě.

Takto zadaná souřadnice má k-dimenzionální tvar a určuje jednoznačný multidimenzionální klíč. Klíčem v tomto případě bude, pro uchování a třídění jednotlivých bodů, 3D souřadnice disponující hodnotou X, Y a Z. K uchování geografických dat můžeme využít následující datovou strukturu [7, 12].

Struktura kD strom

Stromová datová struktura kD strom sloužící k ukládání k-dimenzionálních dat. Třídění dat se ve stromu realizuje vždy podle jedné z dimenzí klíče. V tomto případě se jedná o 3D klíč. Princip třídění takových dat je znázorněný na obrázku.



Obrázek 11 Princip třídění 3D stromu

(zdroj: autor)

Při vybudování struktury 3D stromu jsou tedy data, v každé první úrovni, seřazena podle hodnoty první dimenze klíče. V seřazených datech je určen medián množiny a je vybrán prvek odpovídající mediánu, který určuje kořen stromu nebo podstromu. V každé druhé úrovni stromu jsou data seřazena podle hodnoty druhé dimenze klíče. Je opět vybrán medián množiny a vybrán prvek. V každé třetí úrovni stromu, jsou data tříděna podle hodnoty třetí dimenze klíče. Každá další úroveň má určené pořadí úrovní a postupy se cyklicky opakují, dokud nejsou všechna data uložena ve struktuře stromu. Stejným principem třídění, jsou realizovány operace *Vlož*, *Odeber* a *Najdi*.

Operace *Vlož* a *Najdi*, jsou realizovány postupným traverzováním od kořene stromu je hledána příslušná pozice. V každém vrcholu je rozhodnuto, jakým směrem bude pokračovat následující postup porovnáním příslušné dimenze klíče s diskriminátorem daným pro danou úroveň stromu. Při operaci *Najdi* je porovnáván celý klíč a v případě shody je vrácen aktuální vrchol stromu. Operace *Vlož* při nálezů příslušné pozice prvek vloží.

Při operaci *Odeber* mohou nastat rozdílné situace. Nejsnadnější je případ, kdy se prvek nachází v listu stromu a může být odebrán přímo. Složitější situace nastává, existuje-li pravý nebo levý podstrom prvku. V případě existence pravého podstromu je vybrána náhrada za odebíraný prvek hledáním prvku s nejnižší hodnotou dimenze klíče, která odpovídá úrovni odebíraného prvku. Existuje-li levý podstrom, je vybrána náhrada hledáním prvku s maximální hodnotou dimenze klíče [12].

4.2.2 Struktura mapy

Realizovanou mapu si lze představit jako síť vykonatelných tras. Z toho vyplývá odpovídající struktura síťového grafu. ADT Graf představuje heterogenní bipartitní strukturu pracující se dvěma odlišnými třídami prvků – vrcholy a hrany. Na výběr jsou různé způsoby implementace grafu [7].

Vrcholově orientovaný přístup

Vstupní branou datové struktury je vrchol. S ním jsou spojeny další informace o ostatních vrcholech (předchůdci/následníci) a o příslušných hranách, které jsou při používání k dispozici [7]

Hvězdy - prvotní resp. druhotná struktura může být reprezentovaná buď polem, nebo tabulkou. Mohou být realizovány jako následující abstraktní typy:

- a) pole – pole
- b) pole – tabulka
- c) tabulka – pole
- d) tabulka – tabulka (kde tabulka může být realizována jako pole nebo seznam)

Při reprezentaci struktury polem, jsou vrcholy identifikovány pomocí n přirozených čísel. U tabulky jsou vrcholy reprezentovány klíčovou položkou.

Křížové reprezentace – jedná se o paměťově méně náročnou reprezentaci grafu, která umožňuje přístup k následníkům i předchůdcům.

- a) pole – tabulka (seznam) x pole – tabulka (seznam)
- b) tabulka (seznam) – tabulka (seznam) x tabulka (seznam) – tabulka (seznam)

Hranově orientovaný přístup

Vstupní branou datové struktury je hrana a vyhledávací operace jsou orientovány na vyhledání hran. Mohou být realizovány jako následující abstraktní typy – pole hran a tabulka hran [7].

4.2.3 Uložení objektů mapy

Výstupem navrhované aplikace je vytvořená mapa uchovávaná v navržených datových strukturách. Pro účely opětovného použití vytvořené mapy mohou být data uložena následujícími způsoby:

- a) uložení do souboru v interní nebo externí paměti zařízení,
- b) využití relační databáze,
- c) uložení na webovém serveru,
- d) využití objektové databáze.

Z hlediska výhodnosti přístupu a uchování celých objektů mapy je výhodným řešením využití objektové databáze.

Databáze db4o

Typickým představitelem objektově orientovaných databází je databáze db4o, tedy database for object. Je to open source⁸ objektová databáze vytvořená pro jazyky Java a NET. Databáze db4o nabízí velmi jednoduché použití a pomocí jednoho řádku kódu umožňuje ukládání a načítání objektů. Db4o je primárně určena pro jednouchyvatelské stand-alone aplikace, například pro chytré mobilní telefony [11, 20].

4.3 Použité metody geolokace

Při používání navrhované aplikace se předpokládá využití v místech, kde není dostupný žádný vnější zdroj informace použitelné ke zjištění geografické polohy. Dostupnost signálu se

⁸ Otevřený software

předpokládá pouze ve chvíli, kdy se bude určovat počáteční pozice při mapování. Poté bude využita metoda přímého měření a další pozice bude odhadem vzhledem k počáteční pozici.

4.3.1 Zaměření mobilního zařízení

Metoda pro počáteční kalibraci mobilního zařízení je metoda satelitního navigačního systému GPS. Systém GPS dosahuje dobré přesnosti a je dostupný. Vzhledem k předpokladu využití aplikace mimo obytnou oblast je v podstatě nejpřesnější a nejspolehlivější. Poloha, která bude metodou získána, bude uložena jako počáteční bod.

4.3.2 Výpočet nové pozice

Při určování aktuální pozice zpracovává aplikace pouze informace o vykonaném pohybu. V ideálním případě by se jako nejlepší metoda nabízela možnost inerciální navigace. S ohledem na mobilní zařízení, které bude chytrý mobilní telefon nebo tablet, je tato metoda nepoužitelná. Současné komponenty takových mobilních zařízení nedosahují takové přesnosti a kvality pro získání kvalitních výsledků.

Proto je aplikace navržena na principu výpočtu nové aktuální pozice, z dostupných informací o směru a délce pohybu, které získá pomocí senzorů mobilního zařízení. Vypočítání nové souřadnice a ostatních potřebných údajů se provede dosazením hodnot získaných do následujících rovnic [18].

Výpočet zeměpisné souřadnice

Při znalosti výchozí souřadnice, známé hodnotě délky vykonaného pohybu a známém směru vykonaného pohybu lze nejprve vypočítat hodnotu zeměpisné šířky pomocí následující rovnice:

$$Lat_2 = \arcsin\left(\sin Lat_1 * \cos \frac{d}{R} + \cos Lat_1 * \sin \frac{d}{R} \cos \theta\right)$$

Rovnice 3 Výpočet zeměpisné šířky

kde Lat_1 a Lat_2 jsou hodnoty zeměpisné šířky, θ je hodnota azimutu, d je vzdálenost a R je rádius země⁹.

Po výpočtu zeměpisné šířky lze vypočítat hodnotu zeměpisné délky. To se provede dosazením do následující rovnice:

⁹ Poloměr Země je přibližně 6371m.

$$Long_2 = Long_1 + \arctan2\left(\sin \theta * \sin \frac{d}{R} * \cos Lat_1, \cos \frac{d}{R} - \sin Lat_1 * \sin Lat_2\right)$$

Rovnice 4 Výpočet zeměpisné délky

kde $Long_1$ a $Long_2$ jsou hodnoty zeměpisné délky, θ je hodnota azimutu, d je vzdálenost a R je rádius země a Lat_2 je nová hodnota zeměpisné šířky.

Výpočet azimutu

Při zadání dvou známých bodů lze vypočítat hodnotu azimutu určující směr z počátečního bodu do druhého. Výpočet hodnoty se provede dosazením do následující rovnice:

$$\theta = \arctan2(\sin \Delta Lon * \cos Lat_2, \cos Lat_1 * \sin Lat_2 - \sin Lat_1 * \cos Lat_2 * \cos \Delta Lon)$$

Rovnice 5 Výpočet hodnoty azimutu z bodu 1 do bodu 2

kde ΔLon je rozdíl zeměpisných délek druhého a prvního bodu a Lat_1 a Lat_2 jsou hodnoty zeměpisných šířek.

Výpočet vzdálenosti

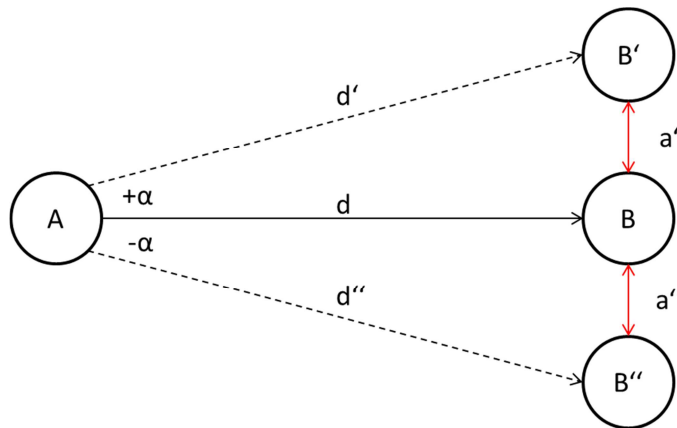
Při zadání dvou známých bodů lze vypočítat vzdálenost mezi nimi podle následující rovnice:

$$d = \arccos(\sin Lat_1 * \sin Lat_2 + \cos Lat_1 * \cos Lat_2 * \cos \Delta Lon) * R$$

Rovnice 6 Výpočet vzdálenosti mezi zadanými body

kde Lat_1 a Lat_2 jsou hodnoty zeměpisných šířek, ΔLon je rozdíl zeměpisných délek druhého a prvního bodu a R je rádius země.

Všechny uvedené rovnice počítají s dvourozměrnou zeměpisnou souřadnicí. U výpočtu nových souřadnic a výpočtu ostatních hodnot, v případě navrhované aplikace, je ale využíváno trojrozměrných souřadnic. Kromě směru a délky může být zadána ještě hodnota inklinace, tedy jestli trasa je ve sklonu klesání, nebo stoupání. Tím se mění některé postupy při výpočtu nové souřadnice a vzdálenosti mezi dvěma body, jak je vidět na obrázku.



Obrázek 12 Změna délky trasy při nastavené inklinaci

(zdroj: autor)

- $\alpha = 0$
 - o výsledkem výpočtu vzdálenosti je hodnota d ;
 - o při výpočtu nového bodu je výsledkem bod B;
- $\alpha > 0$
 - o výsledkem výpočtu vzdálenost je hodnota d' ;
 - o při výpočtu nového bodu je výsledkem bod B';
- $\alpha < 0$
 - o výsledkem výpočtu vzdálenost je hodnota d'' ;
 - o při výpočtu nového bodu je výsledkem bod B'';

V případech, kdy je nastavena hodnota inklinace na nenulovou hodnotu, provede se výpočet potřebných hodnot pomocí Pythagorovy věty o pravoúhlém trojúhelníku.

Určování směru

Směr pohybu bude určován pomocí elektronického kompasu. Minimálním předpokladem je správné určení směru pohybu z počátečního bodu. Další směr může být nastaven jako odchylka k předchozímu nastavenému směru. Vzhledem k tomu, že při dalším postupu hrozí ovlivnění kompasu nežádoucími vnějšími vlivy, bude nastavený pevný přímočarý směr, dokud nedojde k jeho změně. V průběhu pohybu budou v nastaveném intervalu zaznamenávány referenční body s reálnou hodnotou azimutu.

Určování vzdálenosti

K určení délky vykonané dráhy bude sloužit krokomeř realizovaný pomocí akcelerometru. V současné době krokoměry dosahují poměrně přesných výsledků a při správném nastavení

průměrné délky kroku je možné získat přibližnou vzdálenost. Aplikace předpokládá využívání v místech, kde bude dráha vykonaná v jednom směru, přesahovat desítky metrů.

Při postupu mohou nastat i speciální případy, kdy bude trasa vykonávaná pod úhlem stoupání nebo klesání. Tento údaj je potřebný pro výpočet nadmořské výšky v daném bodě. Příkladem takového stoupání nebo klesání mohou být např. schody, výtah atp. V tomto případě bude délka kroku závislá na hodnotě nastavené inklinace a podle toho bude ovlivněna i dosažená vzdálenost.

4.4 Vizualizace mapy

Jedním z úkolů aplikace je možnost vizualizovat vykonanou trasu, tzn. znázornit orientační mapu vykonaných tras. Vizualizovaná mapa by měla obsahovat potřebné informace pro co největší usnadnění orientace. K dispozici se také nabízí velké množství volně přístupných mapových podkladů, které by mohly sloužit k promítnutí dosažených bodů přímo v mapě.

Google Maps

Mapová služba společnosti Google, která nabízí možnost využití mapových služeb prostřednictvím Google Maps API. Jedná se o jednu z nejvyužívanějších mapových služeb, která poskytuje velké objemy dat a přesné výsledky. Google Maps obsahuje SDK¹⁰ pouze pro operační systémy Android a iOS. Mapové služby lze používat i ve webových aplikacích díky Google Maps JavaScript API [17].

Bing Maps

Mapová služba od společnosti Microsoft. Funkcionalita je obdobná se službou Google Maps. Vzhledem k poskytovateli je služba provázána s operačním systémem Windows Phone, kde je služba nativně zahrnuta. Využití je ale možné i pro operační systémy Android a iOS [17].

Open Street Map

Jedná se o open source projekt mapové služby, který je tvořen dobrovolníky z celého světa. Data jsou poskytována pod licencí Open Database License. Není oficiálně vydáno žádné SDK, ale možnost využití poskytuje jeden z mnoha open source projektů, například Osmroid, vývojové nástroje pro Android, RoutMe, nástroje pro iOS, a Leaflet, pro webové

¹⁰ Software development kit

aplikace. Vzhledem k otevřenosti kódu jsou dány menší záruky funkčnosti, ale služba je více flexibilní [17].

V rámci této diplomové práce se nepředpokládá využití externího mapového podkladu. Vizualizace mapy vznikne vykreslením síťového grafu.

4.5 Navigace

Při práci s již hotovým mapovým podkladem je další možností, jak usnadnit pohyb v prostoru, vyhledání nejkratší trasy mezi dvěma body. Při dodržení pokynů navigace lze pomocí krokoměru určovat aktuální pozici na vyhledané trase a zobrazit zbývající vzdálenost.

4.5.1 Problém nejkratší cesty

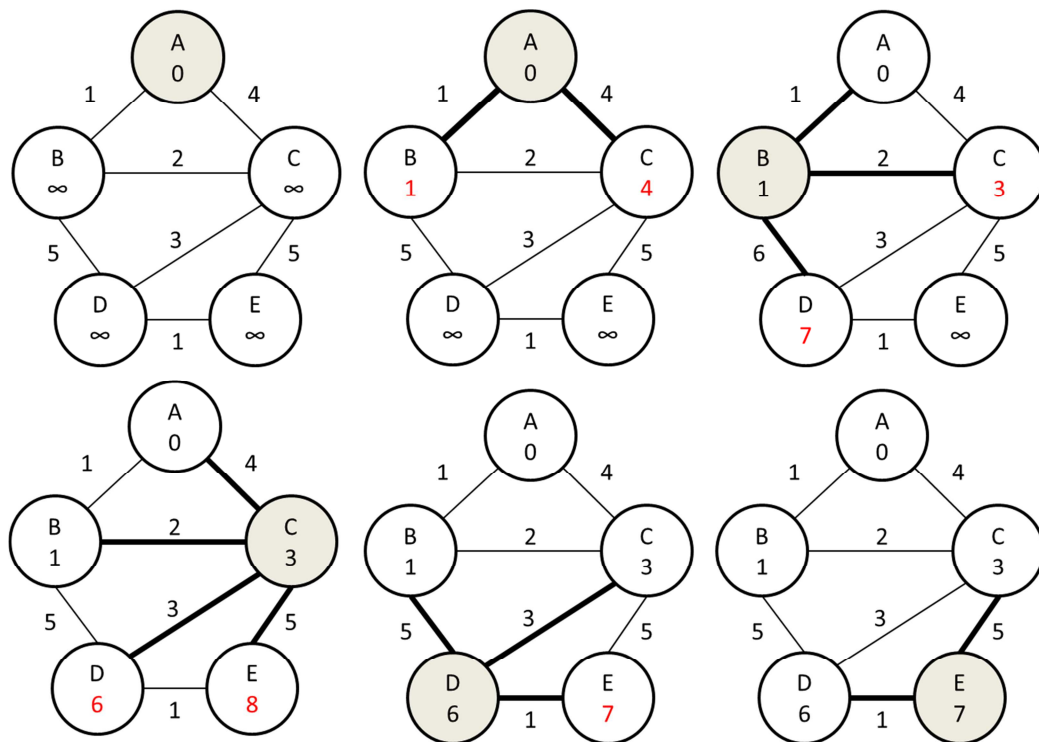
Problém nejkratší cesty je grafová úloha, jejímž základním cílem je nalézt v zadaném grafu nejkratší cesty z jednoho zadaného uzlu. Řešení této základní úlohy pak lze aplikovat i na další varianty hledání trasy (např. trasa mezi dvojicí uzlů).

Dijkstrův algoritmus

Nejefektivnější algoritmus pro hledání nejkratší cesty ze zadaného uzlu do ostatních uzlů grafu. Dijkstrův algoritmus je možné chápat jako určité zobecnění prohledávání grafu do šířky. Počínaje prvním uzlem se šíří systematicky posouvaná vlna, která pohlcuje takové uzly, k nimž byla nalezena nejkratší cesta. Hlavním předpokladem použitelnosti je existence grafu, který obsahuje pouze kladné ohodnocení hran [8].

Princip algoritmu:

- 0) Všechny uzly jsou seřazeny v prioritní frontě podle priority, která je určena vzdáleností od počátečního uzlu.
- 1) Počátečnímu uzlu nastavena priorita na hodnotu 0 a všem ostatním prvkům je nastavena priorita na hodnotu nekonečna.
- 2) V každém následujícím kroku je vybrán prvek s nejvyšší prioritou, tedy nejnížší hodnotou vzdálenosti, a zařazen do množiny zpracovaných prvků.
- 3) Algoritmus pokračuje výběrem jeho nezpracovaných potomků a přiřazuje prvkům novou prioritu. Nová priorita se ověřuje i vůči zpracovaným prvkům.
- 4) Algoritmus skončí po prověření všech prvků prioritní fronty, nebo neexistuje-li kratší varianta cesty.



Obrázek 13 Princip Dijkstrova algoritmu

(zdroj: autor)

Bellman-Fordův algoritmus

Bellman-Fordův algoritmus lze využívat v případě, kdy je potřeba zpracování grafů se záporně hodnocenými hranami. Vyhledá nejkratší trasu a současně dokáže zjistit, zda je splněna podmínka týkající se záporných cyklů. Pracuje na stejném principu jako Dijkstruv algoritmus s rozdílem ve způsobu výběru hran pro techniku relaxe. Proti Dijkstrovu algoritmu jde o výpočetně složitější algoritmus vyhledání nejkratší trasy [8].

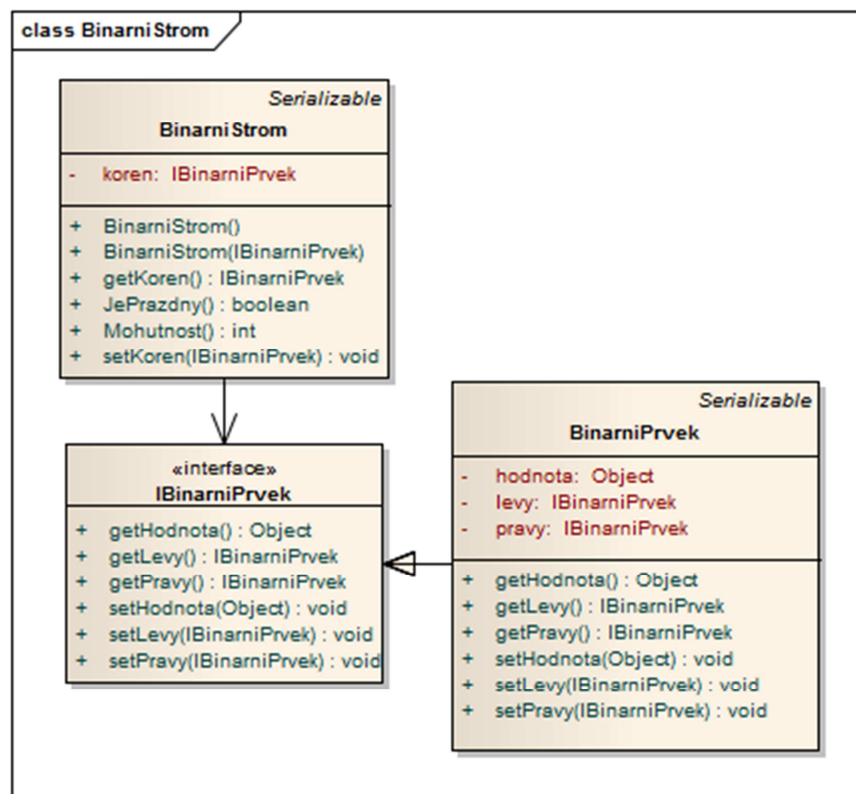
5 Implementace mobilní aplikace

V rámci této diplomové práce je nativní mobilní aplikace realizována pro mobilní platformu OS Android. Vzhledem k tomu je aplikace implementována v programovacím jazyce Java s využitím vývojového prostředí Eclipse.

5.1 Implementace datových struktur

5.1.1 Třída Binární strom

Základem datové struktury 3D stromu je datová struktura binárního stromu. Pro tu je navržena třída *BinarniStrom*. V binárním stromu jsou uloženy binární prvky, pro které je navržena třída *BinarniPrvek*.



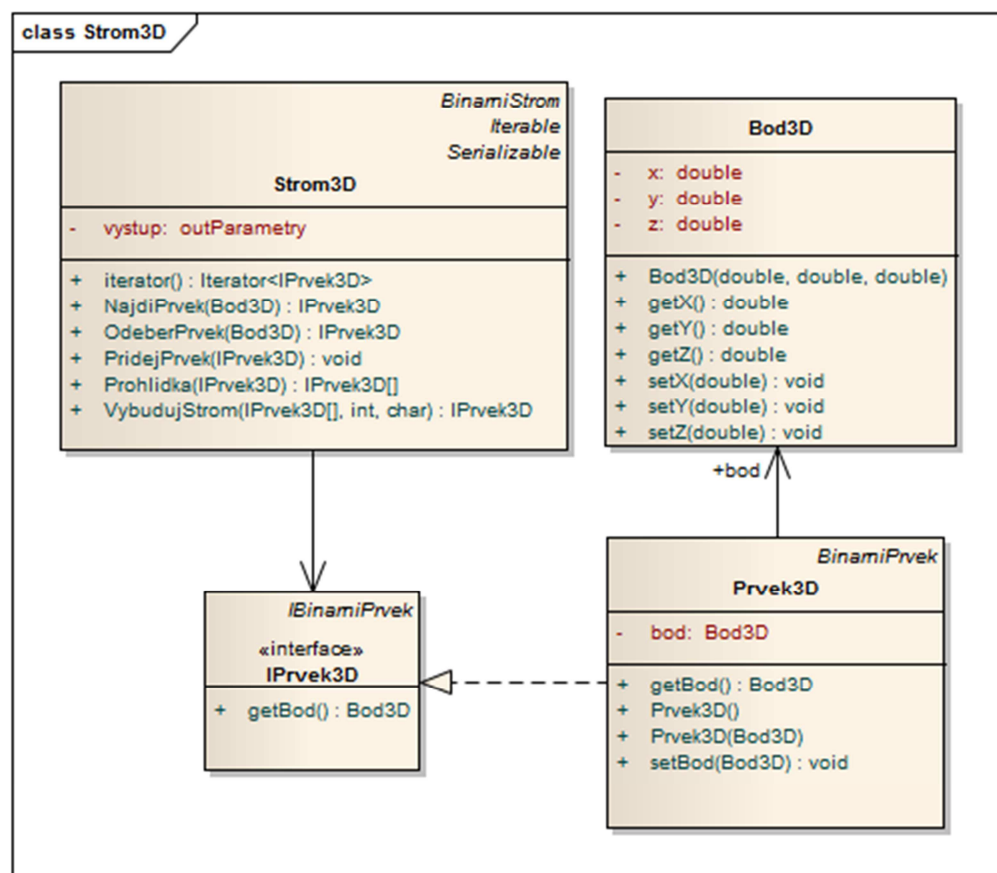
Obrázek 14 Implementace třídy binárního stromu

(zdroj: autor)

Třída *BinarniStrom* obsahuje pouze jeden atribut *koren*, který obsahuje kořenový prvek stromu. Prvek binárního stromu obsahuje jako atributy *levy* a *pravy*, které obsahují reference na levého a pravého potomka.

5.1.2 Třída 3D strom

Pro uchovávání trojrozměrné souřadnice je vytvořena datová struktura 3D strom jehož zobecněním je struktura kD strom. Datová struktura je navržena třídou *Strom3D*, která rozšiřuje třídu *BinarniStrom*. Třída *Strom3D* byla doplněna o atribut typu *outParametry*, který obsahuje parametry využívané při provádění operací nad stromem. Prvek uchovávaný ve struktuře 3D stromu je navržen třídou *Prvek3D*. Ta rozšiřuje třídu *BinarniPrvek*. Třída *Prvek3D* je doplněna o atribut *bod*, která nese hodnotu geografické souřadnice a hodnotu nadmořské výšky. Pro uložení trojrozměrné souřadnice byla navržena pomocná třída *Bod3D*, která má tři hlavní atributy X, Y a Z.



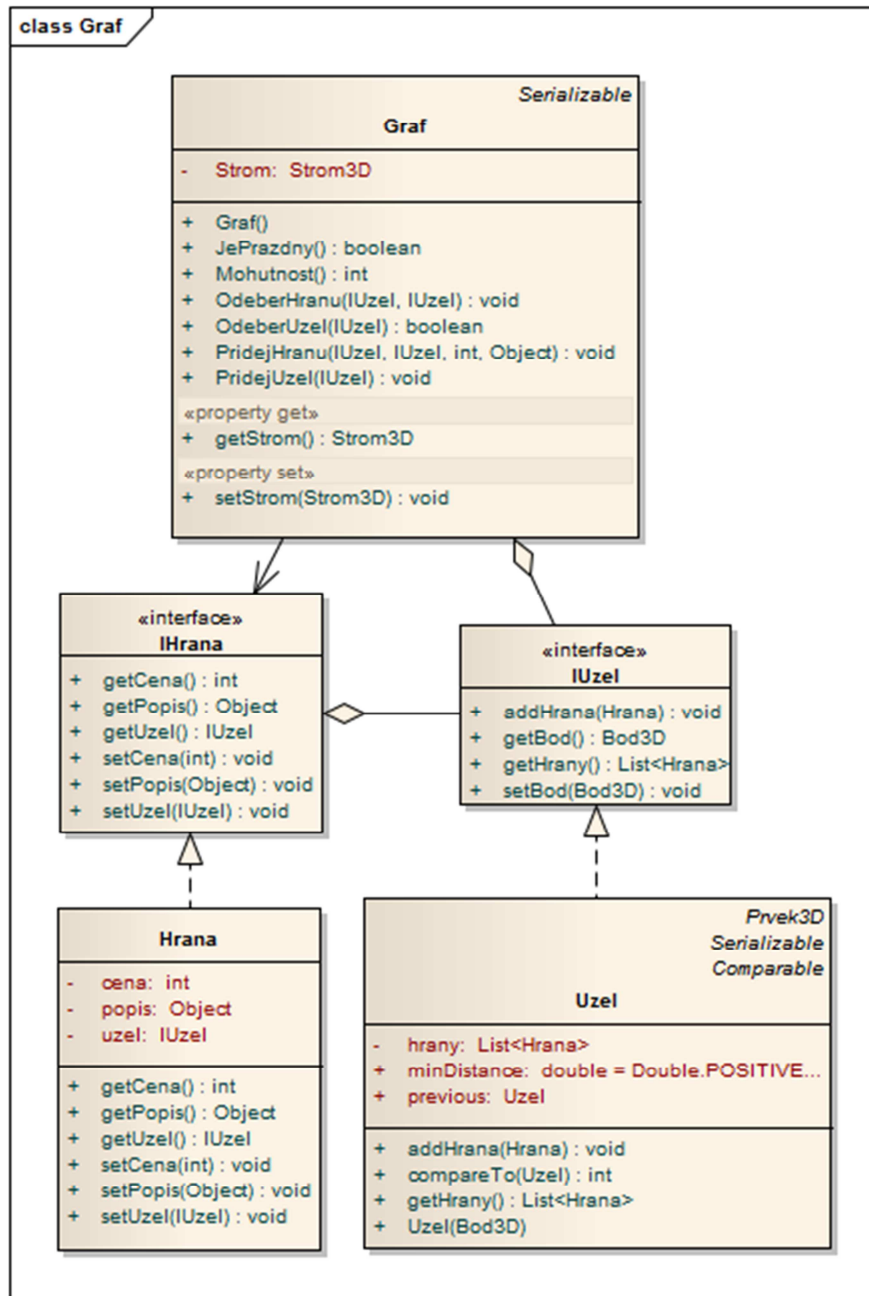
Obrázek 15 Implementace třídy 3D stromu

(zdroj: autor)

5.1.3 Třída Graf

Navrhovaná struktura *Graf* má podobu vrcholově orientovaného grafu implementovaného jako hvězda. Struktura je navržena třídou *Graf*. Ve struktuře grafu je obsažen atribut *Strom*, který obsahuje objekt stromu *Strom3D* uchovávající uzlové body. Uzlové body grafu jsou navrženy třídou *Uzel* vycházející z třídy *Prvek3D*. Každý uzlový bod obsahuje atribut *hrany*.

Ten uchovává datovou strukturu *List<>*, která slouží k uchování hran grafu. Hrany jsou navrženy třídou *Hrana*. V datové části hran je atribut *popis* a *cena*, které uchovávají informace o hraně a atribut *uzel*, který je referencí na následující uzel, s nímž je daná hrana spojena.



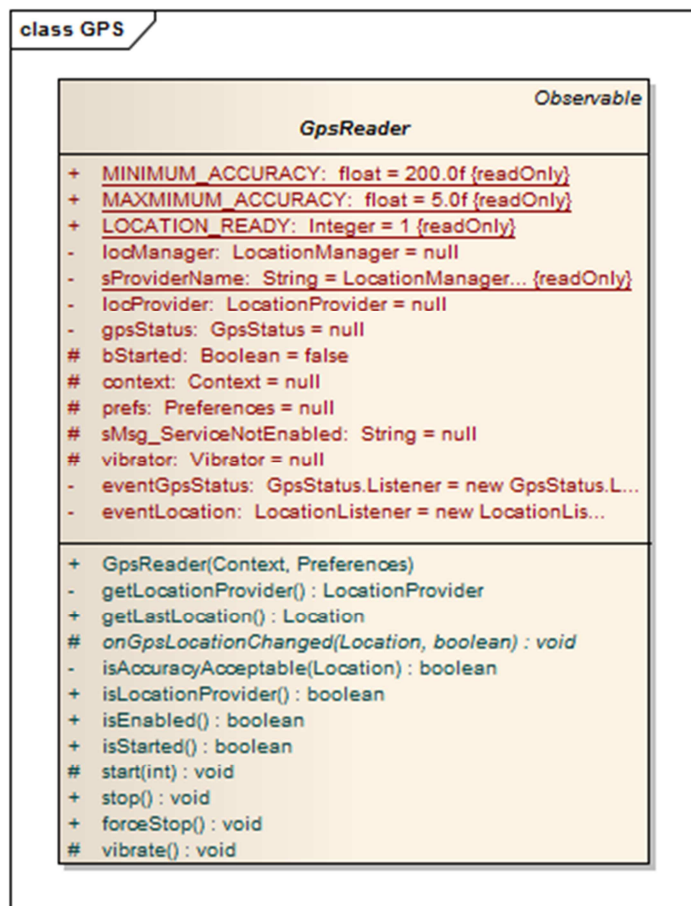
Obrázek 16 Implementace třídy Graf

(zdroj: autor)

5.2 Implementace geolokace

5.2.1 Třída GPS reader

Pro přístup k informacím o geografické poloze mobilního zařízení je pro platformu Android definováno Location Services API. Metody a související služby jsou umístěny v balíčku *android.location*. V rámci aplikace je tato API využívána při implementaci třídy *GpsReader*.



Obrázek 17 Třída *GpsReader*

(zdroj: autor)

Abstraktní třída *GpsReader* je základní třídou, která slouží k získání geografické souřadnice pomocí GPS modulu. Třída přistupuje k systémovým službám prostřednictvím objektu *LocationManager*. V tomto případě přistupuje k systémové službě *LOCATION_SERVICE*, která umožňuje získávat aktualizace o změně polohy mobilního zařízení.

Získání polohových informací je možné z několika různých zdrojů, které API nabízí. K tomuto účelu je definován provider, tj. zdroj, který obsahuje soubor kritérií každého zdroje. Android nabízí dva typy zdrojů a to *NETWORK_PROVIDER* a *GPS_PROVIDER* (viz

kap. 2.3) Je možné využít více zdrojů, ale v rámci této práce je využíván pouze zdroj *GPS_PROVIDER*.

Hlavní částí třídy je definování metod listeneru. Listener je objekt, který slouží k zachycení a zpracování zprávy o změně polohy. Listener má několik předdefinovaných veřejných metod, které jsou po zaregistrování listeneru volány při konkrétní změně stavu. Důležitá je obslužná metoda *onLocationChanged()*, která je volána při změně polohy zařízení. Protože je třída odvozena z třídy *Observable*, lze v případě zachycení zprávy o změně polohy zařízení volat metodu *setChanged()*. Tato metoda informuje o změně stavu skupinu registrovaných objektů typu *Observer*.

Proto, aby aplikace mohla přijímat informace z uvedených zdrojů, je potřeba nastavit povolení přístupu k těmto systémovým službám. Nastavení se provádí v souboru *AndroidManifest.xml* doplněním oprávnění přístupu ke službám *ACCESS_FINE_LOCATION*.

Detailnější popis metod a služeb Location Services API lze najít na stránkách Android Developers¹¹.

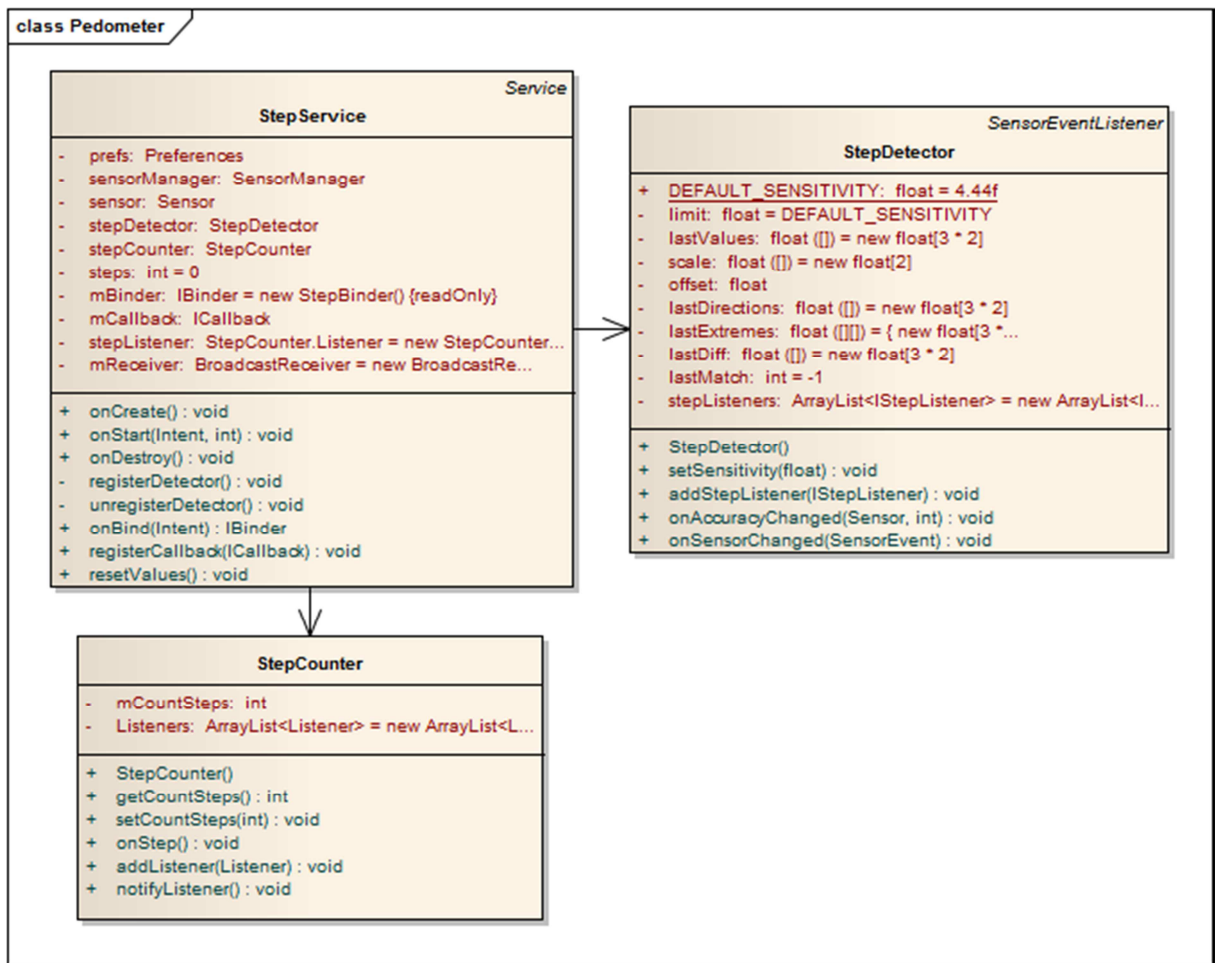
5.2.2 Třída *Step detector* a třída *Compass*

V těchto třídách je realizována práce se senzory mobilního telefonu. Pro práci se senzory a čidly mobilního zařízení je definována Sensors API. Metody a související služby jsou umístěny v balíčku *android.hardware*. Standardní metody balíčku mají za úkol identifikovat senzory, jejich vlastnosti a monitorování běhu událostí senzorů. Identifikování senzorů určí závislosti aplikace na mobilním zařízení, které musí obsahovat dané typy senzorů.

Třída Step detector

Ohodnocení vzdálenosti mezi jednotlivými body se provádí podle údajů získaných pomocí krokoměru. Krokoměr je založen na detekci kroků, která provádí analýzu dat získávaných pomocí Akcelerometru. Základní třídou detekce kroků je třída *StepDetector*.

¹¹ <http://developer.android.com/reference/android/location/package-summary.html>



Obrázek 18 Implementace krokoměru

(zdroj: autor)

Třída *StepDetector* implementuje rozhraní *SensorEventListener*. Toto rozhraní definuje dvě metody, které se využívají pro příjem zpráv o změně hodnot měřených na zadaném senzoru. První metoda *onAccuracyChanged()* je volána při změně přesnosti senzoru. Druhá metoda *onSensorChanged()* je volána při změně stavu měřených dat.

V metodě *onSensorChanged()* implementována samotná detekce kroku. Metoda má vstupní parametr *SensorEvent*. Ten v sobě obsahuje data ze senzoru, který hlásí změnu. V tomto případě jsou zpracována data pouze ze senzoru *TYPE_ACCELEROMETER*. Na výstupní data z akcelerometru je aplikován vstupní filtr a na základě výpočtu je detekován krok. Počet vykonaných kroků se ukládá ve třídě *StepCounter*, která implementuje třídu *IStepListener*.

Celá detekce kroku je spuštěna jako služba definována třídou *StepService*. Služba definuje použití defaultního senzoru a také nastavuje režim objektu *SensorManager*. Ten je nastaven

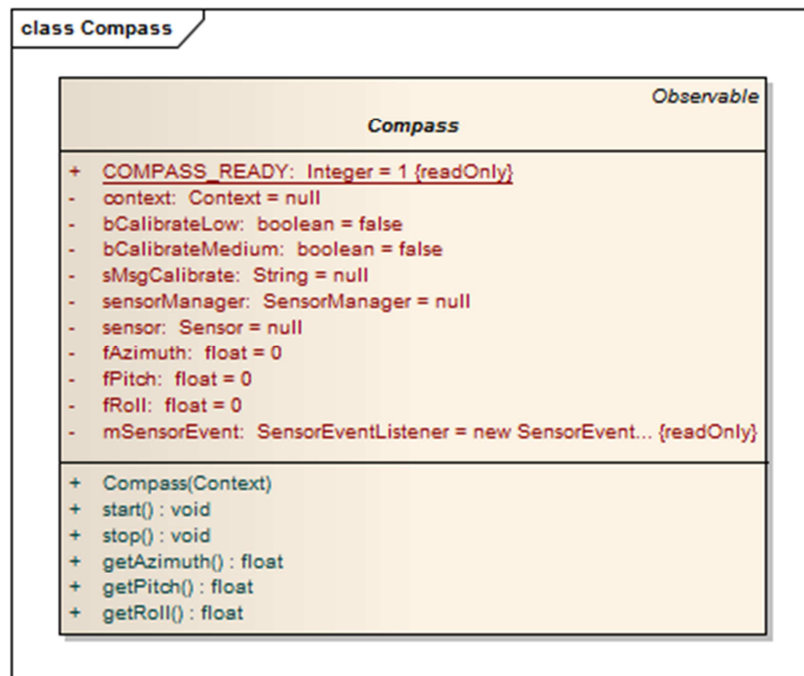
tak, aby jeho odezva byla co nejrychlejší. Aby mohla být služba spuštěna, musí být uvedena v souboru *AndroidManifest.xml* stejně jako všechny *ActivityClass* aplikace.

Třída *Kompas*

K určování směru je v aplikaci implementován magnetický kompas. Hodnoty azimutu jsou získávány z hodnot, která vrací senzor *TYPE_ORIENTATION*. Ten vrací hodnoty úhlů, které svírají jednotlivé osy senzoru s osou směru magnetického severu. Hodnoty mají tvar:

- a) Azimut – úhel mezi osou y a směrem magnetického severu.
 - Hodnoty od 0 – 359;
- b) Pitch – rotace kolem osy x.
 - Hodnoty od -180 – 180;
- c) Roll – rotace kolem osy x
 - Hodnoty -90 – 90;

Kompas je realizován třídou *Compass*, která zajišťuje čtení hodnot ze senzoru. Ve třídě je implementován *SenzorEventListener*, který zajistí naplnění proměnných daty o orientaci mobilního zařízení.

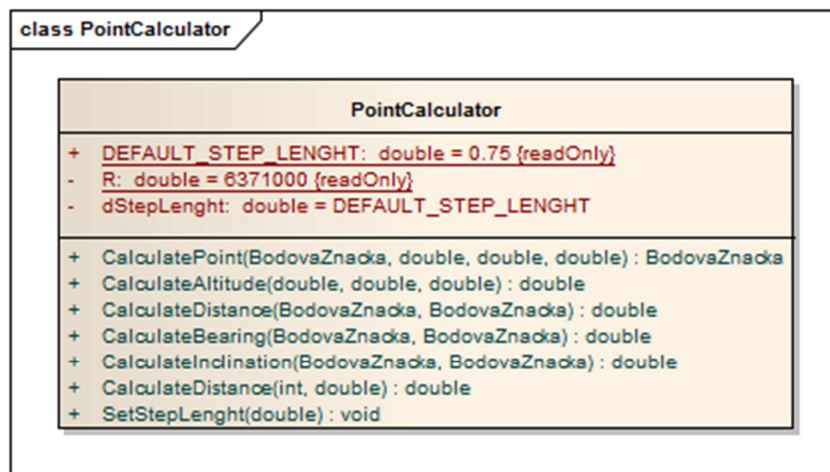


Obrázek 19 Implementace kompasu

(zdroj: autor)

5.2.3 Třída Point calculator

Třída *PointCalculator* obsahuje metody k získání souřadnice nového bodu ze zadaných hodnot délky a směru. Ostatní metody vrací ostatní výsledky výpočtu rovnic uvedených v návrhu aplikace (viz kap. 4.3.2).



Obrázek 20 Třída PointCalculator

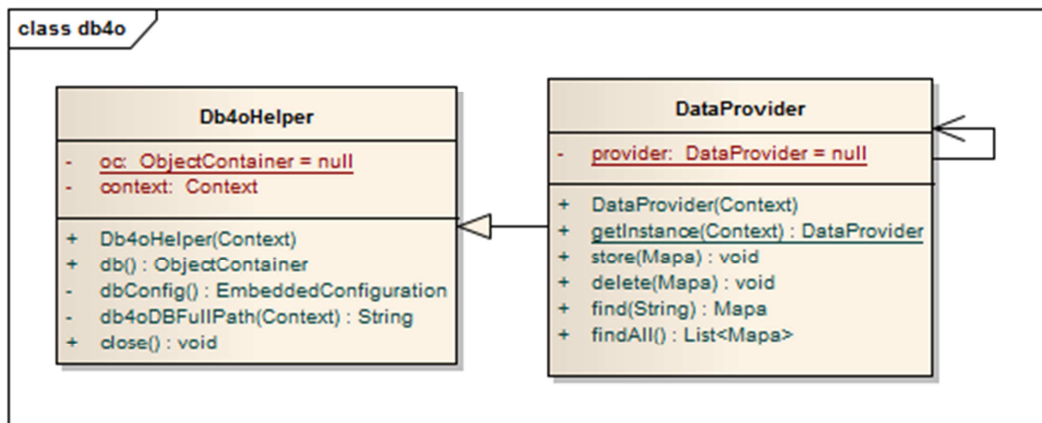
(zdroj: autor)

5.3 Implementace mapy

5.3.1 Uložení mapy

Uložení mapy je realizováno pomocí objektové databáze db4o. Protože db4o není součástí standardních knihoven, musí se potřebné knihovny stáhnout a zahrnout do aplikace. Potřebné knihovny je možné stáhnout například na webových stránkách společnosti Versant, viz zde: <http://community.versant.com/>.

Funkce db4o jsou využívány ve třídě *Db4oHelper*. Obsahuje objekt typu *ObjectContainer*, který představuje samotnou databázi. Z třídy je odvozena třída *DataProvider*, která zpřístupňuje metody pro práci daty v databázi.



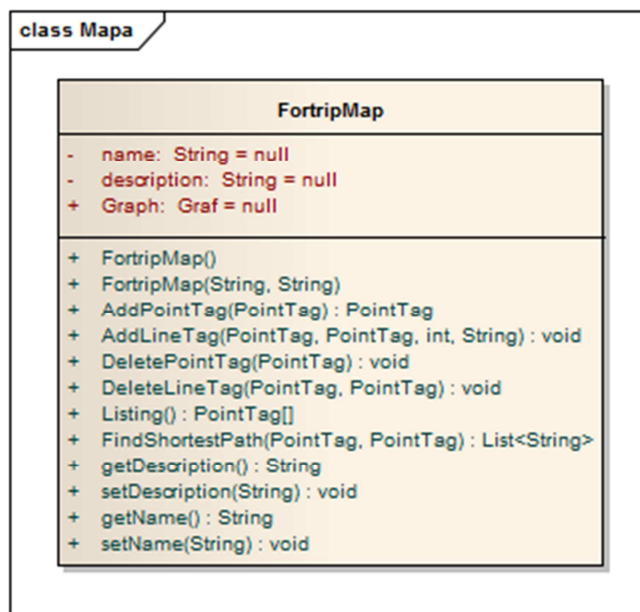
Obrázek 21 Implementace databáze db4o

(zdroj: autor)

5.3.2 Struktura mapy

Třída *FortripMap* uchovává strukturu mapy. Obsahuje atribut *Graph* typu *Graf*, který slouží k uchovávání objektů třídy *PointTag*. Každý objekt třídy *PointTag* rozšiřuje třídu *Uzel* o atributy *name* a *type*, které zaznamenávají informaci o názvu a typu daného bodu. Objekt třídy *PointTag* uchovává množinu objektů třídy *LineTag*, která rozšiřuje třídu *Hrana* o atribut *azimuth*.

Třída *FortripMap* obsahuje metodu *FindShortestParh*, která implementuje problém nejkratší trasy. Metoda má dva vstupní parametry *start* a *end*, které určují dva uzlové body, mezi kterými je pomocí Dijkstrova algoritmu vyhledána nejkratší trasa.



Obrázek 22 Třída FortripMap

(zdroj: autor)

5.4 Aktivity aplikace

Jednotlivé položky uživatelského rozhraní jsou realizovány pomocí tříd odvozených dědičností od třídy *Activity*. Každá z těchto tříd představuje jednu z činností aplikace. S každou třídou aktivity je vytvořen soubor xml, který definuje podobu uživatelského rozhraní, tj. layout. Oba tyto soubory jsou zahrnuty v definici souboru *AndroidManifest.xml* s potřebným nastavením, které může ovlivnit spouštění, vzhled, ovládání aj. Základní třídou aktivit této aplikace je třída *CommonActivity*, která je odvozena od třídy *Activity*. Obsahuje základní obslužné metody všech ostatních aktivit. Všechny ostatní třídy jsou vytvořeny dědičností od této třídy.

MainActivity

Hlavní aktivitou aplikace je aktivita Main, která je definována třídou *MainActivity*. Tato aktivita má na starosti základní obsluhu aplikace a jejím prostřednictvím jsou spouštěny ostatní důležité aktivity aplikace. Layout této aktivity obsahuje základní zobrazení vybrané mapy a v případě hledání nejkratší trasy, zobrazuje i vyhledaný postup mapou a přibližný průběh postupu. Vzhled aktivity je definován souborem *activity_main.xml*. Ten v sobě obsahuje položky tříd *View* pro zobrazení obsahu. Pro vykreslování mapy a vyhledané trasy je vytvořena třída *MainView* odvozená od třídy *View*. Třída *MainView* obsahuje metody sloužící k vykreslování a obsluhu zobrazené mapy.

MapsActivity

Pro vybrání nebo vytvoření nové mapy slouží aktivita definovaná třídou *MapsActivity*. Tato aktivita je spouštěna z aktivity *Main* s požadavkem vrácení informací o vybrané mapě. Layout aktivity, definovaný souborem *activity_maps*, obsahuje položku pro zobrazení seznamu vytvořených map, které lze vybrat a použít k zobrazení nebo editaci. Po vybrání konkrétní mapy aktivita předá informaci o vybrané mapě a ukončí svou činnost.

EditMapActivity

Po vybrání nové nebo již použité mapy je možná editace, která je realizována aktivitou *EditMapActivity*. Aktivita je spouštěna z aktivity *Main* s předanou informací o vybrané mapě. V aktivitě je realizováno vytváření struktury síťového grafu pomocí určeného směru a délky vykonané trasy. V aktivitě je využíváno služby *StepDetectorService* a objektu typu *Compass*. Aktivita spouští další obslužné aktivity.

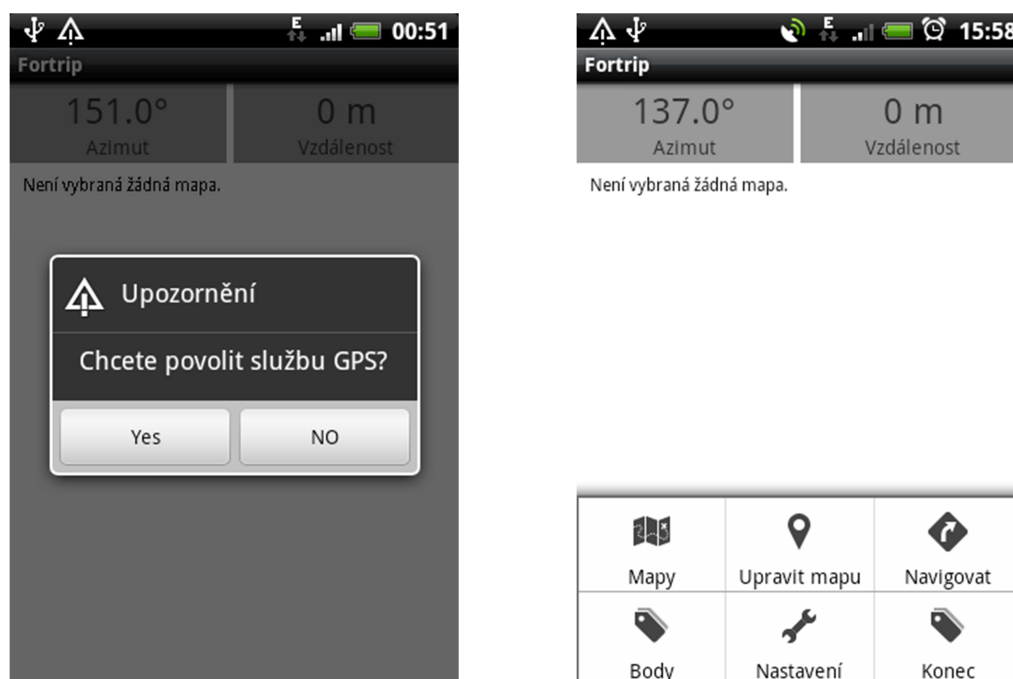
- *PointActivity* – slouží k výběru aktuálního bodu. Je spouštěna s informací o vybrané mapě a umožňuje výběr bodu v mapě. Jako výsledek vrací hodnotu vybraného bodu a ukončuje svojí činnost.
- *NewPointActivity* – slouží k vytvoření nového bodu. Je možné vytvářet bod pomocí předaných informací o směru a délce pohybu, pomocí GPS nebo je možné zadat hodnoty ručně. Jako výsledek je předána hodnota nového bodu, která je spojena hranou s předchozím vybraným bodem.

6 Výsledek

Aplikace byla vyvíjena pro mobilní platformu Android verze 2.3.3 Gingerbread. Všechny činnosti spojené s vývojem aplikace a testováním byly prováděny na mobilním zařízení HTC Wildfire S A510e.

6.1 Uživatelské rozhraní

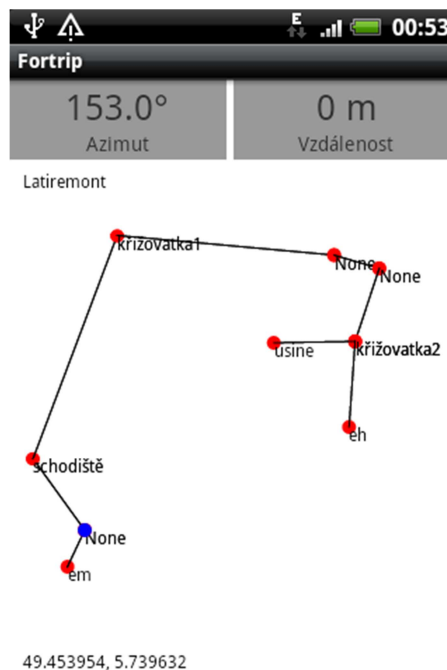
Úvodní obrazovka aplikace je znázorněna na obrázku 23. Při spuštění aplikace je v případě, kdy není povolena služba GPS, uživatel upozorněn na dostupnost služby a je umožněna její aktivace. V dalším kroku je znázorněno hlavní menu aplikace. Při prvním spuštění aplikace je dostupná pouze aktivita pro výběr mapy, která se spustí zvolením záložky *Mapy*. Další možnosti práce s mapou jsou dostupné po vybrání nové nebo již vytvořené mapy. Možnost nastavení je v této verzi aplikace pouze orientační a volba záložky *Konec* způsobí ukončení aplikace.



Obrázek 23 Úvodní obrazovka aplikace

(zdroj: autor)

Po zvolení možnosti *Mapy* je spuštěna aktivita výběru mapy. Je zobrazen seznam aktuálně vytvořených map nebo prázdný seznam. Po úspěšném zvolení mapy z nabídky dostupných map nebo po vytvoření nové mapy se aplikace vrací zpět do hlavního okna, kde dojde k vykreslení aktuálně vybrané mapy. Vykreslení mapy je znázorněno na obrázku 24.

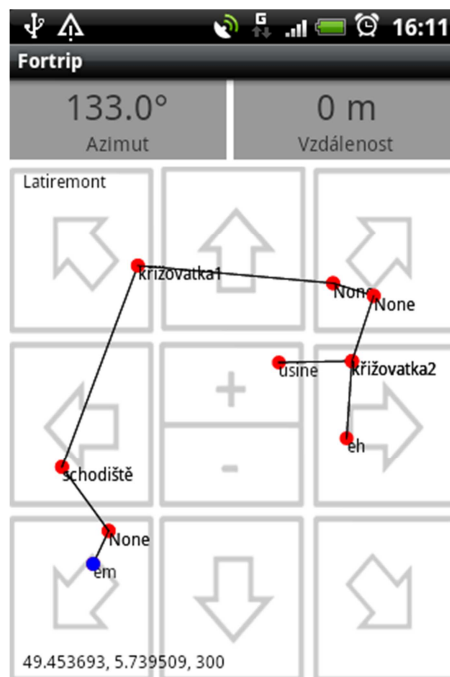


Obrázek 24 Zobrazení mapy v aplikaci

(zdroj: autor)

Ve vykreslené mapě jsou znázorněny všechny dobové značky prostřednictvím červených bodů. Všechny liniové značky, které znázorňují trasy mezi bodovými značkami, jsou znázorněny černými úsečkami. Modrý bod znázorňuje aktuálně vybraný bod, nebo v případě dostupného GPS signálu, aktuální polohu. Pohyb, po aktuálně zobrazené části mapy, je umožněn prostřednictvím dotykového displeje. Možnosti posunu jsou znázorněny na obrázku 25. Ten znázorňuje sektory v ploše zobrazené mapy, které po stisknutí posunou plochu daným směrem. Prostřední část je rozdělena na dvě části a slouží ke změně měřítka mapy.

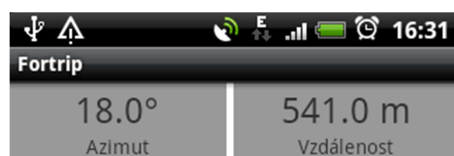
Další možností pohybu v mapě je pomocí záložky *Body* v hlavním menu. Po stisknutí je vytvořen seznam bodů, ze kterého lze vybrat bod s unikátním názvem. Ten je pak vyznačen a umístěn na střed obrazovky.



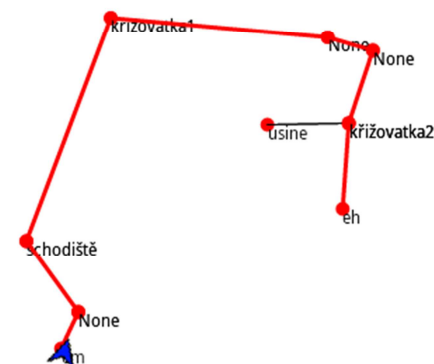
Obrázek 25 Rozložení ovládání zobrazení mapy

(zdroj: autor)

Když mapa obsahuje již vytvořenou strukturu mapy, lze zvolit záložku *Navigovat* v hlavním menu. Je otevřeno dialogové okno, kde je zvolen výchozí a cílový bod mezi kterými dojde k vyhledání nejkratší trasy. Na obrázku 26 je znázorněno vyhledání trasy mezi zadanými body. V momentě, kdy je trasa vyhledána, je nastavena hodnota vzdálenosti a je spuštěna služba krokoměru. Ta určuje hodnotu zbývajících vzdálenosti, která chybí k dosažení cílového bodu. V dolní části obrazu jsou zobrazovány informace o směru dané liniové značky a údaj určující vzdálenost do další změny směru. Modrá značka označuje aktuální přibližnou pozici a údaj zobrazený pod informacemi o směru, je aktuální souřadnicí daného bodu.



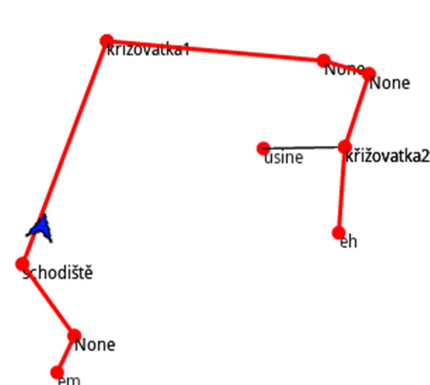
Latiremont



Trasa: , směr 17.0°, 0.0°. Za 29.25m změna směru
49.453699450191436, 5.7395120335882845, 300



Latiremont



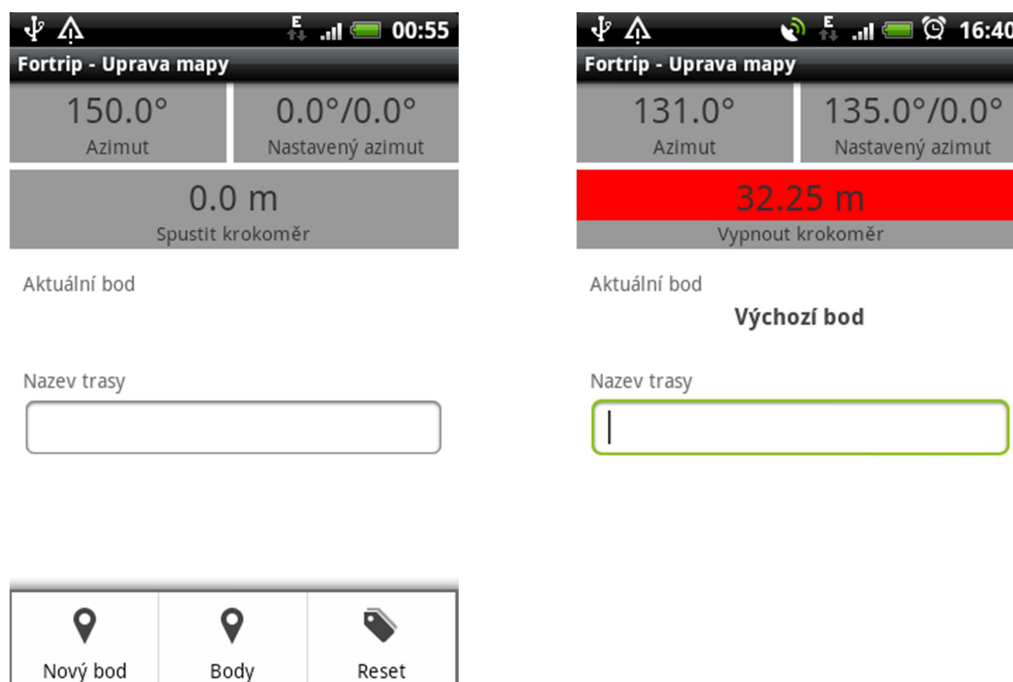
Trasa: , směr 14.0°, 0.0°. Za 147.75m změna směru
49.45477059422855, 5.739379978759172, 300

Obrázek 26 Zobrazení vyhledané trasy

(zdroj: autor)

Poslední záložkou hlavního menu je volba *Upravit mapu*. Prostřednictvím této aktivity je možné upravovat strukturu již existující mapy nebo vytvoření mapy nové. V případě, kdy je struktura mapy prázdná, je prvním krokem vytvoření nového bodu, který bude nastaven jako výchozí bod. U vytvoření tohoto bodu se předpokládá dostupnost příjmu signálu GPS nebo ruční zadání známého bodu. Zadaný bod se stane aktuálním bodem a z jeho známé polohy je možné vypočítat následující bod podle zjištěných informací o vzdálenosti a směru.

V případě, kdy již mapa obsahuje vytvořené body, je možné pomocí záložky *Body* vybrat bod s unikátním názvem a ten zvolit jako výchozí.



Obrázek 27 Editační obrazovka

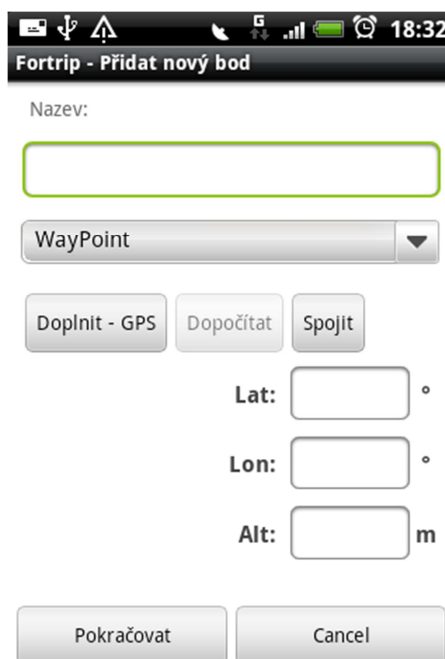
(zdroj: autor)

Mezi možnosti editačního nástroje patří nastavení aktuálního směru a ovládání služby krokoměru. Služba krokoměru se ovládá stiskem v oblasti zobrazující přibližnou aktuální vzdálenost. Po prvním kliknutí do pole je služba spuštěna. V aktuální verzi aplikace je nastavena defaultní hodnota průměrné délky kroku na hodnotu 0,75m a počet vykonaných kroků je násoben touto hodnotou. Dalším kliknutím do pole se služba deaktivuje a dokud není znovu spuštěna, zůstává nastavena hodnota vzdálenosti.

Nastavení aktuálního směru se provádí stisknutím oblasti zobrazení nastaveného azimutu. Po stisku oblasti je spuštěna aktivita nastavení směru. V dialogovém okně je možné nastavit hodnotu azimutu a hodnotu inklinace dané trasy. Nastavená hodnota inklinace ovlivňuje hodnotu průměrné délky kroku. To platí hlavně pro případy, kdy trasa vede například směrem nahoru nebo dolů (schody atp.). Stejně jako hodnota průměrné délky kroku, jsou tyto hodnoty v této verzi aplikace nastaveny pevně. Maximální hodnota inklinace je maximálně $\pm 90^\circ$.

V případě, kdy je aktivní služba detekce kroku, je při každém nastavení změny směru nebo inklinace vytvořen orientační bod s neunikátní hodnotou názvu. Takový bod nelze vybrat pro navigaci ani jako aktuální. Pokud by měla být aktuální pozice zaznamenána jako bod s unikátním názvem, musí být vytvořen pomocí záložky *Nový bod*.

Vybráním záložky *Nový bod* je spuštěna aktivita definování nového bodu. Při vytváření bodu je zadáno unikátní jméno, podle kterého bude bod v mapě identifikován. Další možností identifikace bodu je výběr typu bodu. V aktuální verzi aplikace nemá tato hodnota žádný význam, ale v budoucnu by mohla sloužit například k určení značky, která bude v místě bodu vykreslena. Hodnotu souřadnice bodu je možné doplnit pomocí GPS (když je dostupný signál), dopočítáním (v případě, kdy jsou předány informace o vykonané trase), ručně (když jsou známy souřadnice uzlových bodů) nebo je možné spojit aktuální pozici s již uloženým bodem v mapě.



Obrázek 28 Přidání nového bodu

(zdroj: autor)

6.2 Využití a přesnost aplikace

Navrhovaná aplikace byla vytvořena za účelem využívání v podzemních prostorech s rozsáhlou infrastrukturou tras. Pro tyto prostory jsou charakteristické převážně velmi dlouhé podzemní chodby s minimální členitostí a s minimálními změnami směru. Příkladem mohou být rozsáhlejší důlní díla nebo civilní i vojenské podzemní objekty (například opevnění, kryty).

U takových objektů je možné získat pomocí dostupných metod geografické souřadnice vstupního objektu nebo případně více povrchových objektů. Vzhledem k překonávání velkých vzdáleností mezi jednotlivými objekty a to hlavně podzemními trasami, je výhodné znát

přibližné zobrazení podzemní trasy. V případě, kdy není dostupný žádný mapový podklad, je výhodné tvořit po vstupu do neznámého prostředí strukturu tras. Po dosažení cíle lze využít tento obraz k usnadnění navigace zpět do výchozího bodu, případně k jinému bodu ve vytvořené mapě. Na obrázku 29 je ukázka plánu vstupních objektů podzemního prostoru. Na předchozích obrázcích znázorňující obrazovky aplikace je znázorněna přibližná trasa mezi těmito objekty.



Obrázek 29 Ukázka plánu vstupních objektů podzemního prostoru

(zdroj: www.darkplaces.org)

Protože jsou ke zjišťování geografických souřadnic používány prostředky, které mohou být ovlivněny daným prostředím (jiné magnetické pole) nebo individuálním použitím (lidský faktor), nelze očekávat dosažení velké přesnosti. Aplikace je vhodná k použití v případech, kde nezáleží na velké přesnosti, ale využitelné jsou i orientační výsledky. Pro tyto účely aplikace splňuje požadavky pro použití v uvedených prostorech.

Aplikace by mohla obsahovat automatické funkce pro detekci změny směru a plnit tak některé aktivity automaticky. Od tohoto úmyslu bylo upuštěno z důvodu velké náchylnosti na vnější vlivy ovlivňující měření.

Závěr

Hlavním úkolem v teoretické části diplomové práce bylo podle zadání popsat dostupné geolokační metody, které lze využít prostřednictvím mobilního zařízení. Obsahem popisu jednotlivých metod je srovnání metod z hlediska přesnosti, dostupnosti a možností využití v mobilních aplikacích webových i nativních. Dále popis obsahuje možnost využití lokálně kontextových služeb.

Druhým hlavním cílem diplomové práce bylo v praktické části navrhnout a realizovat nativní mobilní aplikaci, která využívá možnosti geolokace a zpracovává získaná geografická data. Za tímto účelem byla navržena a implementována nativní mobilní aplikace pro mobilní platformu Android. Aplikace umožňuje stavbu síťového grafu ze získaných geografických údajů, který představuje mapu tras v prostorech, kde není možné využít dostupné vnější geolokační metody. Aplikace slouží jako nástroj pro usnadnění orientace v neznámém prostředí a znázornění přibližné aktuální geografické polohy uživatele. Pro tyto účely byly realizovány všechny navržené základní funkcionality aplikace. Díky tomu splňuje požadavky na použití v předpokládaných prostorech a je možné díky jejímu použití dosáhnout snadnější orientace.

Při návrhu aplikace bylo využito dosažených znalostí o datových strukturách, které jsou vhodné pro uchování geografických dat, a možností uložení těchto dat v objektové databázi. Dále při realizaci aplikace byly získány zkušenosti s vývojem navigační mobilní aplikace v prostředí platformy Android za pomoci využití dostupných senzorů a čidel mobilního zařízení.

Současná verze mobilní aplikace nabízí svou základní funkcionality, která by v budoucnu, s nástupem kvalitnějších komponent mobilního zařízení, mohla být doplněna o přesnější metody měření nebo zpřesňováním pomocí jiných metod. Dále je nutné dokončit většinu doplňkových funkcionalit a upravit a přizpůsobit uživatelské rozhraní aplikace, pro snadnější přístup a ovládání aplikace. Aplikace by mohla být vhodným způsobem testována za účelem získání informací o dosažitelné přesnosti získaných dat.

Použitá literatura

- [1] BALEJ, Jiří. Srovnání přesnosti aktivních geolokačních technik. In: *Access server* [online]. 11. 07. 2012 [cit. 2013-07-26]. Dostupné z: <http://access.feld.cvut.cz/view.php?navezclanku=srovnani-presnosti-aktivnich-geolokacnich-technik&cislolanku=2012070001>
- [2] ČÍŽEK, Jakub. Hackujeme Google: vím, kde je tvůj Wi-Fi router. In: *Živě.cz* [online]. 16.12.2010 [cit. 2013-05-01]. Dostupné z: <http://www.zive.cz/clanky/hackujeme-google-vim-kde-je-tvuj-wi-fi-router/sc-3-a-155025/default.aspx>
- [3] ČÍŽMÁR, Jan. *Současnost a budoucnost inerciálních MEMS senzorů*. Brno, 2011, 9 s. Dostupné z: <http://www.roznovskastredni.cz/dwnl/pel2011/06/cizmar.pdf>
- [4] ĎAĎO, Stanislav a Marcel KREIDL. *Senzory a měřicí obvody*. Praha: ČVUT, 1996. ISBN 80-01-01500-9.
- [5] GLONASS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-25]. Dostupné z: <http://cs.wikipedia.org/wiki/GLONASS>
- [6] HOJGR, Radek a Jan STANKOVIČ. *GPS: praktická uživatelská příručka*. Vyd. 1. Brno: Computer Press, 2007, 221 s. ISBN 978-80-251-1734-7.
- [7] KAVIČKA, Antonín. *Graf: Elektronické sylaby přednášek k předmětu Datové struktury a algoritmy*. Pardubice, 2012, 35 s.
- [8] KOLÁŘ, Josef. *Teoretická informatika*. 2. vyd. Praha: Česká informatická společnost, 2004. ISBN 80-900853-8-5.
- [9] KYSELA, Jiří. Lokálně kontextové služby v praxi. In: *Internet pro všechny* [online]. 5.8.2011 [cit. 2013-05-29]. Dostupné z: <http://www.internetprovsechny.cz/lokalne-kontextove-sluzby-v-praxi/>
- [10] KYSELA, Jiří. Webové aplikace s geolokací. In: *Internet pro všechny* [online]. 5.11.2012 [cit. 2013-05-29]. Dostupné z: <http://www.internetprovsechny.cz/webove-aplikace-s-geolokaci/>

- [11] MANEIRO, Damasia. Using db4o in an Android application. In: *DZone: Javalobby* [online]. © 1997-2012 [cit. 2013-08-01]. Dostupné z: <http://java.dzone.com/articles/using-db4o-android-application>
- [12] MICHEK, Tomáš. *Datové struktury pro uchování geografických dat*. Pardubice, 2011. Diplomová práce. Univerzita Pardubice. Vedoucí práce doc. Ing. Antonín Kavička, Ph.D.
- [13] MIKLOŠÍK, František. *Základy užití map*. 1. vyd. Brno: Vojenská akademie, 1999.
- [14] MO ČR. *Standardizovaná vojenská topografická mapa*. Dobruška: Vojenský geografický a hydrometeorologický úřad, 2005.
- [15] ODBOR KOSMICKÝCH TECHNOLOGIÍ A DRUŽICOVÝCH SYSTÉMŮ. *Český kosmický portál* [online]. © 2013 [cit. 2013-08-05]. Dostupné z: <http://www.czechspaceportal.cz/>
- [16] ORLICH, M. Základní lokalizační metody v GSM. In: *Access server* [online]. 28. 02. 2006 [cit. 2013-07-26]. Dostupné z: <http://access.feld.cvut.cz/view.php?cisloclanku=2006022801>
- [17] VANDROL, Jan. *VÝVOJ NAVIGAČNÍ APLIKACE V PROSTŘEDÍ ANDROID*. Ostrava, 2013.
- [18] VENESS, Chris. Calculate distance, bearing and more between Latitude/Longitude points. *Movable Type Scripts* [online]. © 2002-2012 [cit. 2013-08-01]. Dostupné z: <http://www.movable-type.co.uk/scripts/latlong.html>
- [19] VERNER, Lukáš a Dan KOMOSNÝ. Geolokace síťových zařízení v internetových sítích. *Elektrorevue*. 17.6.2011, roč. 2011, č. 33. Dostupné z: <http://www.elektrorevue.cz/file.php?id=200000615-26db227d50>
- [20] VERSANT CORP. *Db4objects* [online]. © 2000-2013 [cit. 2013-08-01]. Dostupné z: <http://www.db4o.com/>
- [21] *QR kódy* [online]. 2007 [cit. 2013-07-26]. Dostupné z: <http://www.qrkody.info/>

Příloha A – CD s vytvořenou aplikací

Obsah přiloženého CD

- Text práce ve formátu PDF;
- Soubor formátu ZIP obsahující adresář UML, ve kterém je soubor EAP obsahující návrhový model aplikace;
- Soubor formátu ZIP obsahující adresář s názvem Fortrip, ve kterém je umístěn projekt nativní aplikace pro platformu Android;

Příloha B – programátorská příručka

Příloha obsahující popis klíčových tříd, které byly implementovány v navržené mobilní aplikaci. Diagramy jednotlivých tříd lze najít v kapitole 5.

Rozhraní *IBinarniPrvek*

Rozhraní definující metody, které mají být implementovány v datové struktuře dané jako prvek binárního stromu.

Metody mají vlastnosti metod *Getter* a *Setter* a nastavují hodnoty atributů sktruktury prvku binárního stromu.

Třída *BinarniPrvek*

Třída *BinarniPrvek* sloužící k vytváření prvků, uchovávaných v datové struktuře binárního stromu.

Atributy třídy:

- *hodnota* – atribut typu *Object*, který bude uchovávat hodnotu prvku,
- *levy* – reference na prvek stejného typu, který představuje levého potomka,
- *pravy* – reference na prvek stejného typu, který představuje pravého potomka.

Metody třídy:

Implementuje všechny metody uvedené v rozhraní *IBinarniPrvek*.

Třída *BinarniStrom*

Představuje datovou strukturu binárního stromu, která je základem pro datovou strukturu 3D stromu.

Atributy třídy:

- *koren* – kořen stromu typu *BinarniPrvek*.

Metody třídy:

- *BinarniStrom()* – bezparametrický konstruktor třídy, který nastavuje hodnotu kořene na *null*,
- *BinarniStrom(IBinarniPrvek koren)* – parametrický konstruktor, který nastavuje hodnotu kořene podle předaného parametru,
- *JePrazdny()* – metoda vracující hodnotu typu *Boolean* podle toho je-li struktura prázdná,
- *Mohutnost()* – metoda vracující hodnotu *Integer*, jenž určuje počet prvků nacházejících se ve struktuře stromu.

Ostatní metody mají vlastnosti metod *Setter* a *Getter*.

Třída Bod3D

Pomocná třída *Bod3D* sloužící k vytvoření objektu uchovávajícího hodnoty trojrozměrné souřadnice.

Atributy třídy:

- *x* – atribut typu *double* obsahující klíčovou hodnotu *x*,
- *y* – atribut typu *double* obsahující klíčovou hodnotu *y*,
- *z* – atribut typu *double* obsahující klíčovou hodnotu *z*.

Metody třídy:

- *Bod3D(double x, double y, double z)* – konstruktor třídy, který podle předaných parametrů nastaví klíčové hodnoty,

Ostatní metody mají vlastnosti metod *Setter* a *Getter*.

Rozhraní IPrvek3D

Rozhraní obsahuje metody, které musí být implementovány ve třídě prvku 3D stromu. Implementuje rozhraní *IBinarniPrvek*.

Třída Prvek3D

Třída sloužící k vytváření objektů prvků, uchovávaných v datové struktuře 3D stromu. Rozšiřuje třídu *BinarniPrvek*.

Atributy třídy:

- *bod* – uchovává hodnotu bodu prvku typu *Bod3D*.

Metody třídy:

- *Prvek3D()* - bezparametrický konstruktor třídy,
- *Prvek3D(Bod3D prvek)* – parametrický konstruktor, který nastaví hodnotu bodu s předaného parametru.

Třída **OutParametry**

Pomocná třída k třídě *Strom3D*, která slouží k vytváření pomocných objektu při operacích s prvky stromu *NajdiPrvek* a *OdeberPrvek*.

Atributy třídy:

- *znak* – atribut typu *char*, který určuje znak třídění na dané úrovni stromu,
- *rodic* – atribut uchovávající prvek, který představuje rodiče aktuálního prvku.

Metody třídy mají vlastnosti metod *Setter* a *Getter*.

Třída **Strom3D**

Třída představující datovou strukturu 3D strom, která rozšiřuje třídu *BinarniStrom*.

Atributy třídy:

- *outParametry* – atribut typu *OutParametry* pro provádění operací s prvky *Odeber* a *Najdi*.

Metody třídy:

- *VybudujStrom(IPrvek3D[] body, int hloubka, char znak)* – metoda sloužící k vybudování struktury stromu na hlavní úrovni případně na úrovni podstromu, k čemuž slouží další předané parametry *hloubka* a *znak*,
- *PridejPrvek(IPrvek3D prvek)* – metoda sloužící k přidání prvku do struktury stromu,
- *OdeberPrvek(IPrvek3D prvek)* – metoda sloužící k odebrání prvku ze struktury stromu vracující hodnotu odebíraného prvku,

- *NajdiPrvek(Bod3D bod)* – metoda vracející hodnotu prvku, který je vyhledán na základě zadaného bodu,
- *Prohlidka(IPrvek3D prvek)* – metoda zpřístupňuje pole prvků, které jsou obsaženy ve struktuře stromu, případně pole prvků odpovídající podstromu, který je určen předaným prvkem.

Rozhraní IUzel

Rozhraní obsahující metody, které mají být obsaženy ve struktuře, která je uchovávána v datové struktuře grafu. Metody rozhraní mají vlastnosti metod *Setter* a *Getter*.

Rozhraní IHrana

Rozhraní obsahující metody, které mají být obsaženy ve struktuře, která je uchovávána v datové struktuře seznamu každého uzlu grafu. Metody rozhraní mají vlastnosti metod *Setter* a *Getter*.

Třída Uzel

Třída implementuje rozhraní *IUzel* a rozšiřuje třídu *Prvek3D*. Slouží k vytváření objektů uchovávaných ve struktuře grafu. Třída implementuje také třídu *Comparable*, která umožňuje porovnávání jednotlivých uzlů.

Atributy třídy:

- *hrany* – atribut uchovávající seznam hran, které vycházejí z tohoto uzlu a jsou spojeny s dalším uzlem ve struktuře grafu.

Metody třídy:

- *Uzel(Bod3D)* – parametrický konstruktor vytvoří objekt uzlu na základě předaného bodu a vytvoří nový prázdný seznam hran,
- *compareTo(Uzel other)* – metoda sloužící k porovnávání jednotlivých uzlů.

Implementuje metody rozhraní *IUzel*.

Třída Hrana

Třída implementuje rozhraní *IHrana* a slouží k vytváření objektů hran, které jsou uchovány v seznamu každého z uzlů, které spojují.

Atributy třídy:

- *popis* – atribut typu *Object*, který v sobě uchovává popisné informace o hraně,
- *uzel* – reference na prvek typu *IUzel*, který určuje prvek, s nímž je hrana spojena z jedné strany,
- *cena* – atribut uchovávající hodnotu ohodnocení hrany.

Metody jsou implementací metod z rozhraní *IHrana*.

Třída Graf

Třída představující datovou strukturu grafu.

Atributy třídy:

- *strom* – atribut uchovávající stromovou strukturu 3D stromu.

Metody třídy:

- *Graf()* – bezparametrický konstruktor, který vytvoří objekt grafu a vytvoří novou instanci stromu,
- *JePrazdny()* – metoda vrací výsledek metody *JePrazdny()* stromu 3D,
- *Mohutnost()* – metoda vrací výsledek metody *Mohutnost()* stromu 3D,
- *PridejUzel(IUzel uzel)* – metoda vloží nový uzel do struktury grafu,
- *OdeberUzel(IUzel uzel)* – metoda odebere zadaný uzel ze struktury grafu,
- *NajdiUzel(Bod3D bod)* – metoda hledající zadaný body ve struktuře stromu, vracející uzel grafu,
- *Prohlidka()* – metoda vracející pole všech uzlů grafu,
- *PridejHranu(IUzel aUzel, IUzel bUzel, int cena, Object popis)* – metoda vytvoří hranu mezi dvěma uzly grafu a přidá hranu k daným uzlům,

- *OdeberHranu(IUzel aUzel, IUzel bUzel)* – odebere hranu, která existuje mezi dvěma danými uzly grafu.

Třída **PointCalculator**

Třída obsahuje metody pro výpočet geografických údajů na základě zadaných hodnot.

Atributy třídy:

- *DEFAULT_STEP_LENGTH* – statický atribut obsahující defaultní hodnotu délky kroku,
- *R* – statický atribut obsahující hodnotu přibližného poloměru země,
- *dStepLength* – atribut typu *double* uchovávající hodnotu průměrné délky kroku.

Metody třídy:

- *CalculatePoint(PointTag point, double distance, double bearing, double inclination)* – metoda vypočítá souřadnici nového bodu ze zadaných hodnot *point*, známý bod, *distance*, vzdálenost od známého bodu, *bearing*, hodnota azimutu od známého bodu k novému, a *inclination*, hodnotou inklinace směrem k od známého bodu k novému,
- *CalculateAltitude(double altitude, double distance, double inclination)* – metoda vypočítá hodnotu nadmořské výšky z hodnot *altitude*, hodnota známé výšky, *distance*, vzdálenost od bodu se známou výškou, a *inclination*, hodnota inklinace od bodu se známou výškou,
- *CalculateDistance(PointTag pointA, PointTag pointB)* – metoda vypočítá vzdálenost mezi dvěma zadanými body,
- *CalculateBearing(PointTag pointA, PointTag pointB)* – vypočítá hodnotu azimutu mezi dvěma body, směrem z bodu *pointA* do bodu *pointB*,
- *CalculateInclination(PointTag pointA, PointTag pointB)* – vypočítá hodnotu inklinace mezi dvěma body, směrem od *pointA* k bodu *pointB*,
- *CalculateDistance(int stepCount)* – metoda vypočítá vzdálenost po zadání počtu kroků,
- *SetStepLength(double inclination)* – nastaví délku kroku v závislosti na hodnotě inklinace.

Třída FortripMap

Třída reprezentující strukturu mapy.

Atributy třídy:

- *name* – atribut typu String obsahující jméno mapy,
- *description* – atribut typu String obsahující popis mapy,
- *Graph* – objekt typu Graf obsahující strukturu grafu, které slouží k uchování prvků grafu.

Metody třídy:

- *FortripMap()* – bezparametrický konstruktor třídy,
- *FortripMap(String nazev, String popis)* – parametrický konstruktor třídy nastavující hodnotu atributu jména a popisu,
- *AddPointTag(PointTag bod)* – metoda sloužící pro vložení nové bodové značky,
- *AddLineTag(PointTag pointA, PointTag pointB, double distance, String name, double azimuth, double inclination)* - metoda vložení nové liniové značky mezi dva zadané body,
- *DeletePointTag(PointTag uzal)* – metoda sloužící k smazání bodové značky,
- *DeleteLineTag(PointTag uzela, PointTag uzalB)* – metoda sloužící ke smazání liniové značky z mapy,
- *FindPoint(Bod3D bod)* – metoda sloužící k vyhledání bodové značky,
- *Listing()* – metoda vracející hodnotu všech bodových značek obsažených ve struktuře grafu,
- *FindShortestPath(PointTag start, PointTag end)* – metoda obsahující Dijkstruv algoritmus vyhledání nejkratší trasy.

Třída Db4oHelper

Třída vytvářející objektovou databázi db4o, která slouží k ukládání struktury mapy.

Atributy třídy:

- *objectContainer* – atribut obsahující objekt typu *ObjectContainer*, který představuje databázi db4o,
- *context* – atribut obsahující kontext třídy, ve které je databáze používána.

Metody třídy:

- *Db4oHelper(Context context)* – parametrický konstruktor třídy,
- *db()* – metoda vracející hodnotu atributu *objectContainer*,
- *dbConfig()* – metoda obsahující nastavení databáze,
- *db4oDBFullPath(Context ctx)* – metoda vracející hodnotu adresářové cesty souboru s uloženou databází,
- *close()* – metoda zavírající otevřenou databázi.

Třída DataProvider

Třída rozšiřující třídu Db4oHelper a obsahující metody pro obsluhu dat v databázi.

Atributy třídy:

- *provider* – obsahující objekt typu *DataProvider*.

Metody třídy:

- *DataProvider(Context ctx)* – parametrický konstruktor třídy,
- *getInstance(Context ctx)* – metoda zajišťující vytvoření pouze jedné instance třídy podle návrhového vzoru singleton,
- *store(DataContainer exercise)* – metoda sloužící k uložení dat do databáze,
- *delete(DataContainer exercise)* – metoda sloužící k smazání dat z databáze,
- *find(String name)* – metoda sloužící k vyhledání dat v databázi podle názvu mapy,
- *findAll()* – metoda vracející všechny záznamy v mapě.

Třída **DataContainer**

Třída sloužící k vytváření objektů ukládaných do databáze. Z důvodu problému s velkým zanořením objektu stromu, byla vytvořena tato třída.

Atributy třídy:

- *sName* – atribut obsahující název uložené mapy,
- *sDesc* – atribut obsahující popis uložené mapy,
- *aPoints* – atribut obsahující objekt *ArrayList*, který obsahuje všechny prvky ukládané mapy.

Metody třídy mají vlastnosti metod Setter a Getter.

Třída **Command**

Třída sloužící k definování a uchování kódu jednotlivých příkazů a stavů v aplikaci. Většina atributů třídy jsou veřejné hodnoty kódů typu *Integer*.

Třída **Main**

Třída slouží k obsluze jednotlivých příkazů obsažených ve třída *Command*.

Atributy třídy:

- *mParent* – atribut určující rodičovskou třídu aktivity, která vyvolá obsluhu příkazu.

Metody třídy:

- *Main(CommonActivity parent)* – parametrický konstruktor třídy,
- *runCommand(int idCmd)* – metoda obsluhy daného příkazu z třídy *Command*.

Ostatní metody jsou metody patřící k obsluze jednotlivých příkazů.

Třída **CommonActivity**

Třída odvozena dědičností od třídy *Activity*, sloužící jako základ všech dalších aktivit aplikace. Obsahuje všechny společné metody a atributy.

Atributy třídy:

- *mContext* – atribut typu *Context*,
- *mIntent* – atribut typu *Intent*,
- *main* – atribut třídy *Main*,
- *buttons* - atribut třídy *CommonButtons*, které obsahuje obsluhu standardních tlačítek aplikace,
- *db4oHelper* – atribut obsahující databázi db4o,
- *selectMap* – atribut typu *DataContainer* obsahující vybranou mapu,
- *sSelectMap* – hodnota atributu obsahuje název vybrané mapy,
- *calculator* – atribut třídy *PointCalculator* sloužící k výpočtu nových souřadnic,
- *selectPoint* – atribut obsahující aktuálně vybraný bod v mapě,
- *dDistance* – hodnota aktuální vzdálenosti,
- *dAzimuth* – hodnota aktuálního azimutu,
- *dSetAzimuth* – hodnota aktuálně nastaveného azimutu,
- *dSetInclination* – hodnota aktuálně nastavené inklinace.

Metody třídy:

- *showActivity(Class<?> classToShow, int requestCode, Bundle data)* – metoda sloužící ke spuštění aktivity s očekáváním výsledku spuštěné aktivity,
- *showActivity(Class<?> classToShow, Bundle data)* – spuštění aktivity bez očekávání výsledku, ale s předáním dat v objektu typu *Bundle*,
- *showActivity(Class<?> classToShow)* – prosté spuštění aktivity,
- *closeActivityWithResult(int resultCode, Bundle data)* – ukončení aktivity s předáním výsledku aktivitě, která vyvolala spuštění,
- *closeActivityWithResultOK(Bundle data)* – ukončení aktivity kladným výsledkem,
- *closeActivityWithResultCancel(Bundle data)* – ukončení aktivity se záporným výsledkem.

Ostatní metody mají vlastnosti metod setter a getter.

Třída MainActivity

Třída odvozena dědičností od třídy *CommonActivity*. Obsahuje obsluhu hlavního obrazu aplikace.

Atributy umožňující uchování objektů sloužících k obsluze vykreslení mapy a obsluze navigace po vyhledané nejkratší trase.

Klíčové metody třídy:

- *update()* – metoda aktualizující zobrazení objektu View,
- *updateDistance()* – metoda sloužící k úpravě vzdálenosti při navigaci po vyhledané trase,
- *update(Observable observable, Object data)* – metoda aktualizace stavu objektu *GpsReader* a *Compass*,
- *onActivityResult(int requestCode, int resultCode, Intent data)* – metoda přebírající výsledky spuštěných aktivit.

Třída MapsActivity

Třída odvozena dědičností od třídy *CommonActivity*, sloužící k vytvoření seznamu dostupných map. Umožňuje spuštění aktivity *NewMapActivity*. Vrácená hodnota při ukončení aktivity je název vybrané mapy.

Třída NewMapActivity

Třída odvozena dědičností od třídy *CommonActivity*, umožňující vytvoření a uložení nové mapy.

Třída NavigationActivity

Třída odvozena dědičností od třídy *CommonActivity*, sloužící k výběru dvou bodů ve vybrané mapě, mezi kterými má být vyhledána nejkratší trasa. Aktivita vrací hodnotu dvou vybraných bodů.

Třída EditMapActivity

Třída odvozena dědičností od třídy *CommonActivity* umožňující editaci mapy.

Klíčové metody třídy:

- *onActivityResult(int requestCode, int resultCode, Intent data)* - metoda přebírající výsledky spuštěných aktivit,
- *addPointTag(PointTag point)* – metoda umožňující vytvoření nového bodu podle zadaných podmínek,
- *addLineTag(PointTag point)* – metoda umožňující vytvoření liniové značky.

Třída PointsActivity

Třída odvozena dědičností od třídy *CommonActivity*, sloužící k vytvoření seznamu dostupných bodů v mapě. Vrácená hodnota při ukončení aktivity je název vybraného bodu.

Třída NewPointActivity

Třída odvozena dědičností od třídy *CommonActivity*, sloužící k vytvoření nového bodu podle zadaných informací, nebo pomocí GPS. Vrácená hodnota při ukončení aktivity je hodnota nového bodu.

Třída AzimuthActivity

Třída odvozena dědičností od třídy *CommonActivity*, sloužící k nastavení hodnoty azimutu a inklinace. Vrací hodnotu nového azimutu a inklinace.