

**Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta**



Mapy do PDA

Bakalářská práce

Vlastimil Almer

Školitel: Ing. Václav Novák, CSc.

České Budějovice 2011

Bibliografické údaje

Vlastimil Almer, 2011: Mapy do PDA.

[Maps to PDA. Bc.. Thesis, in Czech.] – 46 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Abstrakt

Bakalářská práce se zabývá návrhem a programováním aplikace určené ke zobrazování mapových podkladů a získávání GPS souřadnic objektů na mapě. V práci jsou popsány metody získávání mapových podkladů a GPS souřadnic. Práce dále obsahuje návrh souřadnicového systému, výpočet GPS souřadnice a polohy v aplikaci, jejich zpřesnění a algoritmus určený ke zobrazování mapových podkladů.

Abstract

This thesis deals with design and programming of applications designed to display maps and obtain GPS coordinates of objects on the map. The work describes methods for obtaining maps and GPS coordinates. The work includes design coordinate system, calculate the GPS position in the application of more accurate and the algorithm designed to display maps.

Klíčová slova

PDA, Windows Mobile, .NET Framework, GIS, GPS, mapové podklady

Keywords

PDA, Windows Mobile, .NET Framework, GIS, GPS, maps

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

.....

Vlastimil Almer

14.12.2011

Poděkování

Rád bych poděkoval Ing. Václavu Novákovi, CSc. a Bc. Vilému Jandovi za jejich rady při tvorbě této bakalářské práce.

Obsah

1	ZADÁNÍ PRÁCE	1
2	ÚVOD	2
3	TEORETICKÁ ČÁST	3
3.1	PDA (PERSONAL DIGITAL ASSISTANT)	3
3.2	WINDOWS MOBILE	3
3.3	PLATFORMA .NET	4
3.4	GLOBÁLNÍ POLOHOVACÍ SYSTÉM (GPS)	5
3.5	UNIVERZÁLNÍ TRANSVERZÁLNÍ MERCATORŮV SYSTÉM SOUŘADNIC (UTM)	7
3.6	GEOGRAFICKÝ INFORMAČNÍ SYSTÉM (GIS)	7
4	ANALÝZA PROBLÉMU	8
4.1	POŽADOVANÉ FUNKCE APLIKACE	8
4.2	ZOBRAZENÍ MAPOVÝCH PODKLADŮ	8
4.3	ZÍSKÁNÍ MAPOVÝCH PODKLADŮ	8
4.4	ZÍSKÁNÍ A ULOŽENÍ GPS SOUŘADNIC	9
4.5	METODY ZPŘESNĚNÍ	11
4.6	VÝPOČET VZDÁLENOSTI DVOU GPS SOUŘADNIC	11
5	NÁVRH ŘEŠENÍ	12
5.1	DIAGRAM UŽITÍ	12
5.2	UŽIVATELSKÉ ROZHRANÍ	13
5.3	BLOKOVÉ SCHÉMA PROGRAMU	14
5.4	TŘÍDY	15
5.5	SOUŘADNICOVÝ SYSTÉM	16
5.6	ZOBRAZENÍ MAPOVÝCH PODKLADŮ	19
5.7	ADRESÁŘOVÁ STRUKTURA	21
6	IMPLEMENTACE ŘEŠENÍ	22
6.1	PROJEKT	22
6.2	TŘÍDY	22
6.3	ZOBRAZENÍ MAPOVÝCH PODKLADŮ	31
6.4	VÝPOČET GPS SOUŘADNICE NA MAPĚ	33
6.5	ZPŘESNĚNÍ GPS SOUŘADNIC NA MAPĚ	33
6.6	VÝPOČET VZDÁLENOSTI DVOU GPS SOUŘADNIC	34
7	TESTY PŘESNOSTI	35
8	ZÁVĚR	36
8.1	ZHODNOCENÍ PRÁCE	36
8.2	NÁVRH PRO BUDOUCÍ ŘEŠENÍ	37
	PŘEHLED LITERATURY	39
	SEZNAM PŘÍLOH	41
A	OBSAH PŘILOŽENÉHO CD	42
B	UŽIVATELSKÁ PŘÍRUČKA	43
B.1	POŽADAVKY PŘED INSTALACÍ	43
B.2	INSTALACE	43
B.3	ODINSTALACE	43
B.4	POUŽITÍ PROGRAMU	43
C	NAMĚŘENÉ GPS SOUŘADNICE	44
C.1	POROVNÁNÍ GPS SOUŘADNIC	44
C.2	VIZUALIZACE NAMĚŘENÝCH HODNOT	46

1 Zadání práce

Pro biologická pozorování a měření v přírodě je mnohdy potřeba přesné označení bodu v terénu, ve kterém se vyskytuje příslušný sledovaný objekt. Standardní souřadnice GPS často nestačí svoji přesností k označení místa. Například objektu jako je jeden strom. Student má za úkol navrhnout mapové podklady do Smart metering device tak, aby mohla obsluha zaznamenat s přesností na 1 m polohu sledovaného objektu v krajině.

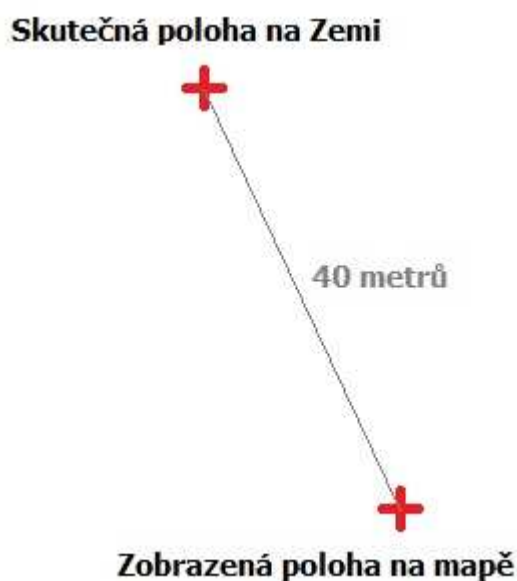
Student navrhne a naprogramuje software do PDA. Zvolí pro mapové podklady do PDA vhodný souřadnicový systém, Implementuje příklady map, které optimalizuje na velikost uložení v paměti PDA. V zobrazení krajiny bude možno provádět zoom a označovat dotykem stylusu přesné místo na mapě.

Dále bude software realizovaný jako knihovna např.: DLL, kterou bude možné připojit k jiným vyvíjeným programům. S tím souvisí pochopitelně definice a implementace vhodného rozhraní.

Provede zkušební testy a vyhodnotí přesnost měření. Součástí práce je i dokumentace užití vytvořeného díla.

2 Úvod

V dnešní době je často potřeba určit přesnou polohu na Zemi a zobrazit jí na mapě. Může to být například za účelem nalezení daného místa a následného zaznamenání informací o daném místě. Zdroj GPS souřadnic může být GPS přijímač nebo databáze objektů na mapě obsahující GPS souřadnice. Je jedno jaký zdroj GPS souřadnic je k dispozici, vždy je potřeba přesně zobrazit GPS polohu na mapě, což představuje problém, pokud zobrazená poloha na mapě vůči skutečné poloze na Zemi má dosahovat přesnosti jednoho metru. V praxi to pak vypadá následujícím způsobem. Je potřeba zobrazit bod na mapě a rozdíl polohy bodu na mapě vůči skutečné poloze na Zemi může být v řádech desítek až stovek metrů. V této práci se bere zdroj GPS souřadnic jako přesný a ovlivňuje se pouze přesnost jeho zobrazení na mapě.



Obrázek 2.1: Rozdíl polohy na Zemi vůči poloze na mapě

Cílem této práce je tedy navrhnout a naprogramovat aplikaci do PDA na platformě Windows Mobile 6, která bude sloužit k zobrazování mapových podkladů a získávání přesných GPS souřadnic daných bodů v krajině. Pro tuto aplikaci je nutné navrhnout souřadnicový systém, metody zpřesnění GPS souřadnic na mapě a algoritmus zobrazování mapových podkladů.

3 Teoretická část

Teoretická část obsahuje základní stavební kameny nutné k vypracování této práce a na základě kterých jsem budoval aplikaci určenou ke zobrazení mapových podkladů. Jedná se o pojmy PDA, Windows Mobile, .NET, GPS, UTM a GIS.

3.1 PDA (Personal Digital Assistant)

PDA je kapesní počítač ovládaný dotykovou obrazovkou a perem (stylusem). Současná PDA jsou velmi výkonná a zvládají v podstatě to samé jako osobní počítač. Skládají se ze základní desky s procesorem, z flash paměti pro ukládání dat, operační paměti RAM a barevného dotykového displeje. Současná PDA mají možnost připojení k internetu. PDA může obsahovat i slot na SIM kartu a modul GSM/GPRS díky čemuž jej lze využívat také jako mobilní telefon. Některé PDA mají vlastní GPS přijímač, k jiným lze tento přijímač připojit např. pomocí Bluetooth. [1] [3] [7]

3.1.1 Operační systém

Nejpoužívanější operační systémy do PDA jsou:

- Windows Mobile
- PalmOS
- Symbian

3.1.2 Synchronizace s PC

Synchronizace slouží ke zkopírování souborů a informací z osobního počítače do PDA a naopak. PDA se vkládá do tzv. kolébky, která slouží k napájení a k připojení k PC. Samotnou synchronizaci obstarávají např. tyto programy:

- ActiveSync
- Windows Mobile Device Center
- HotSync

3.1.3 Vývoj aplikací

- Visual Studio
- Embedded Visual C++
- Netbeans

3.2 Windows Mobile

Windows Mobile je operační systém určený pro mobilní zařízení (PDA, smartphony) vyvíjený společností Microsoft. Windows Mobile je založen na jádru Windows CE a obsahuje sadu základních aplikací vyvinutých pomocí Windows API. Využívá vzhled přejatý z klasických Windows, malou část jejich Win32 API, ale zcela nové hybridní jádro. Aplikace do systému lze zakoupit pomocí Windows Marketplace. [14]

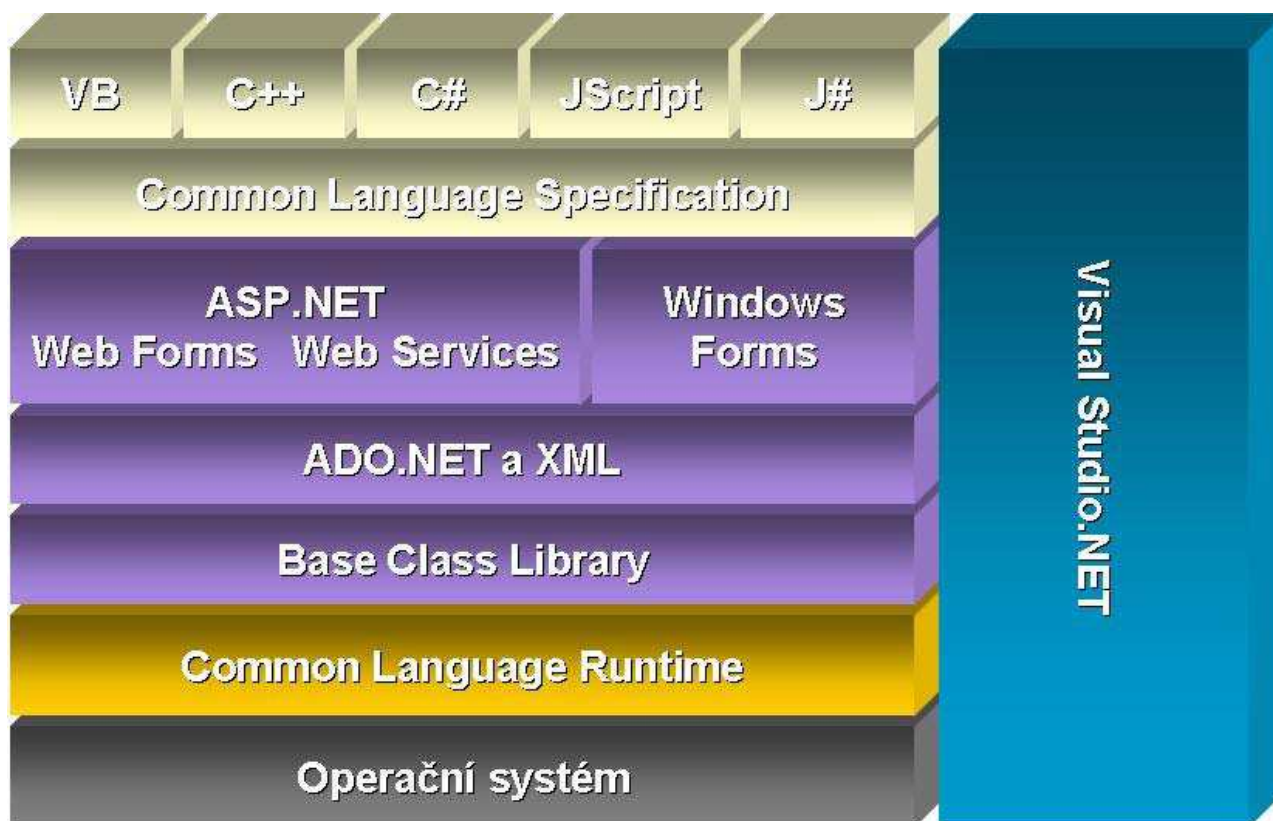
3.3 Platforma .NET

Platforma .NET je soubor technologií v softwarových produktech, jejímž základem je .NET Framework.

3.3.1 Dostupnost Platformy

- .NET Framework – platforma pro osobní počítače s operačním systémem Microsoft Windows od verze 98
- .NET Compact Framework – platforma určená pro mobilní zařízení s operačním systémem Windows Mobile
- .NET Micro Framework – platforma určená pro embedded zařízení

3.3.2 .NET Framework



Obrázek 3.1: Architektura .NET [16]

.NET Framework je platforma pro vytváření a provozování aplikací. Jejimi základními komponenty jsou CLR (Common Language Runtime - společný běhový modul) a FCL (Framework Class Library - základní knihovna tříd). CLR abstrahuje služby operačního systému a slouží jako vykonávací jádro pro řízené aplikace. FLC poskytuje objektově orientované rozhraní API. Jednou z výhod .NETu je podpora různých programovacích jazyků. [15]

3.3.2.1 Common Language Runtime (CLR)

CLR se nachází nad operačním systémem a poskytuje virtuální prostředí pro hostování řízených aplikací. Každý zapsaný bajt kódu běží v CLR nebo mu CLR povolí běžet mimo tento modul. Zdrojové kódy se nepřekládají do nativního kódu, ale do intermediárního jazyka (CIL). Instrukce CIL se na požádání (just-in-time) zkompilují do nativního strojového kódu za běhu. Důležitou částí CLR je společný typový systém (CTS). [15]

3.3.2.2 Programovací jazyky

Výhodou provozování řízených aplikací v CLR je to, že veškerý kód se omezuje na CIL. CLR nečiní mezi jazyky žádný rozdíl. Jazyk je jen syntaktický prostředek pro vytváření CIL. Framework dokonce umožňuje napsat určitou třídu v jednom jazyku a použít ji v jiném.

Microsoft nabízí kompilátory CIL pro jazyky C#, J#, C++, Visual Basic a Jsript. Aplikace samozřejmě může být psána rovnou v CIL. Slouží k tomu ILASM assembler. Jiné společnosti nabízejí kompilátory pro další jazyky jako Perl, Python a dokonce COBOL. [15]

3.3.2.3 Framework Class Library (FCL)

Jedná se o rozhraní API respektive knihovnu tříd .NET Frameworku s více než 7000 typy (třídami, strukturami, rozhraními, výčty a delegáty). Některé třídy obsahují až 100 metod, vlastností a dalších členů.

Knihovna tříd je rozdělena do hierarchických jmenných prostorů, má kolem 100 jmenných prostorů. Každý jmenný prostor obsahuje třídy a další typy. Příklady hlavních jmenných prostorů FCL: [15]

- System – základní datové typy a pomocné třídy
- System.Collections – hešovací tabulky, pole a další kontejnery
- System.Data – třídy ADO.NET pro přístup k datům
- System.Drawing – třídy pro práci s grafikou
- System.IO – vstupně/výstupní operace s proudy a soubory
- System.Net – třídy obalující síťové protokoly
- System.Threading – třídy pro vytváření a spravování vláken
- System.Web – třídy podpory http
- System.Xml – třídy pro čtení a zápis XML dat

3.3.2.4 Visual Studio .NET

V praxi se používá k sestavení aplikací nástroj Visual Studio .NET. Mezi mnoho jeho prvků patří integrovaný editor, kompilátor a debugger, kontextová nápověda ve formě IntelliSense a zabudovaný návrhář formulářů.

3.3.3 .NET Compact Framework

Jedná se o verzi .NET Frameworku určenou pro běh na mobilních zařízeních. Používá některé shodné knihovny tříd jako .NET Framework a pár knihoven určených pouze pro mobilní zařízení.

3.4 Globální polohovací systém (GPS)

Jedná se primárně o vojenský systém, ale v dnešní době hojně využíván i v civilní sféře hlavně z důvodu jeho vysoké polohovací přesnosti, schopnosti určovat rychlost a čas

a v neposlední řadě dostupnosti 24 hodin denně kdekoliv na Zemi. [6]

3.4.1 Struktura GPS

GPS je tvořen třemi základními segmenty:

- Kosmickým – 24 družic rozmístěných na oběžných drahách vysílající navigační signály
- Řídícím – skládá se z pozemních stanic aktualizujících údaje obsažené v navigačních zprávách
- Uživatelským – GPS přijímače, uživatelé, vyhodnocovací nástroje a postupy

3.4.2 Určování polohy

Navigační přístroj např. mobilní telefon s podporou GPS přijímá od aktuálně viditelných družic navigační signály obsahující čas a identifikátor družice. Se znalostí pozice družice na obloze je možné vypočítat vzdálenost ke konkrétní družici. Pro určení přesné polohy polohy je nutno mít signál minimálně ze tří družic. Čím více signálů z družic je přístroj schopem zachytit, tím lze určit přesnější polohu. [17]

3.4.3 Souřadnicový systém GPS

Země je charakterizována tělesem zvaným geoid, který je ovšem pro matematický popis nevhodný, proto se používá jeho aproximace na kouli nebo elipsoid. Pro potřeby uživatelů GPS byla zvolena aproximace elipsoidem podle systému WGS-84 (World Geodetic System 1984), který lze aplikovat s poměrně vysokou přesností na celou Zemi. [6]

3.4.4 Zeměpisné souřadnice

Souřadnice v systému WGS-84 jsou udávány pomocí dvou hodnot – zeměpisná šířka a zeměpisná délka.

3.4.4.1 Zeměpisná šířka

Zeměpisná šířka (latitude, N) určuje polohu na povrchu Země směrem k severu nebo jihu od rovníku. Jde o úhel, který svírá rovina rovníku s přímkou procházející středem Země a příslušným bodem na povrchu Země. Zeměpisná šířka je úhel v rozsahu od 0 stupňů do 90 stupňů. [18]

3.4.4.2 Zeměpisná délka

Zeměpisná délka (longitude, E) určuje polohu na povrchu země směrem k západu nebo východu od Greenwichského (nultého) poledníku. Jde o úhel, který svírá rovina místního poledníku procházející příslušným bodem a rovina nultého poledníku. [19]

3.4.4.3 Formáty souřadnic

GPS souřadnice se udávají ve třech formátech:

- Stupně – 48.97731°N, 14.44788°E
- Stupně a minuty – N 48° 58.6384', E 14° 26.8730'
- Stupně, minuty a vteřiny – 48° 58' 38.306"N, 14° 26' 52.378"E

3.5 Univerzální transverzální Mercatorův systém souřadnic (UTM)

UTM systém mění naši trojrozměrnou Zemi na dvourozměrný systém, který umožňuje přesnější měření vzdálenosti, jedná se o systém založený na mřížkách. Nejedná se o jedno mapové zobrazení narozdíl od GPS, ale o síť šedesáti zón. Jedna zóna je zobrazení části elipsoidu do roviny. Díky tomu je možné měřit vzdálenost bodů pomocí Pythagorovy věty, ale pouze, pokud body leží v jedné zóně. [11]

Střed souřadnic je v každé zóně jiný. Je tvořen průsečíkem středového poledníku s rovníkem. Od tohoto průsečíku se měří vzdálenost po ose x směrem na východ (eastings) a po ose y směrem na sever (nordings). [12]

3.6 Geografický informační systém (GIS)

Definice (výkladový slovník ministerstva hospodářství): Organizovaná kolekce počítačového technického vybavení, programového vybavení, geografických dat a personálu určená k účinnému sběru, pamatování, údržbě, manipulaci, analýze a zobrazování všech forem geograficky vztažené informace.

3.6.1 Části GISu

GIS se skládá z pěti hlavních částí:

- Hardware
- Software
- Data
- Lidé
- Metody

Pro tuto bakalářskou práci jsou nejdůležitější části týkající se softwaru a dat. Softwarová část se skládá z vlastního softwaru pro práci s geografickými daty a datová část jsou popisné informace, časové informace a prostorové informace. Tato část je nejdůležitější a nejnákladnější.

3.6.2 Mapové vrstvy

Mapové vrstvy se dělí na dva typy:

- Rastrové
- Vektorové

V bakalářské je využito rastrového typu. Tato reprezentace se zaměřuje na danou lokalitu jako celek na rozdíl od vektorové reprezentace, která popisuje jednotlivé geografické objekty matematicky. Základním stavebním kamenem rastrové reprezentace je tzv. buňka. Buňky jsou organizovány do mozaiky. Jednotlivé buňky obsahují hodnoty zastupující zkoumanou lokalitu.

4 Analýza problému

V této části je popsáno, co bylo potřeba rozmyslet a učinit před vlastním návrhem a programováním aplikace do PDA určené ke zobrazení mapových podkladů.

V první řadě se jednalo o požadované funkce v aplikaci, získání mapových podkladů a jakým způsobem tyto mapové podklady zobrazit na displej PDA. Dále, jak k těmto mapovým podkladům uložit GPS souřadnice jednotlivých buňek (jeden konkrétní mapový podklad), jak tyto GPS souřadnice získat a zpřesnit. Nakonec se jedná o výpočet vzdálenosti dvou GPS bodů.

4.1 Požadované funkce aplikace

Aplikace by měla obsahovat okno pro výběr oblasti pro přiblížení, dále by měla obsahovat okno určené ke zobrazení mapových podkladů a funkce zajišťující pohyb po mapě, měření vzdálenosti mezi body na mapě a zjištění GPS souřadnic bodů na mapě.

4.2 Zobrazení mapových podkladů

Mapové podklady se dají zobrazovat různým způsobem. V případě rastrových map se jedná v podstatě o pokročilý prohlížeč obrázků. Existuje několik možností. Buď se zvolí pouze jeden mapový podklad reprezentující celou mapu. V tomto řešení bude problém s rychlostí načtení dané lokality, což je v případě kapesních zařízení nemyslitelné. Další a reálnější možností je danou mapu rozdělit na několik mapových podkladů a načítat pouze ty, které jsou v danou chvíli potřeba. Zde jsou další možnosti, jak zvolit parametry jednotlivých mapových podkladů. Je třeba zvolit velikost mapového podkladu tak, aby jich nebylo příliš mnoho, ale na druhou stranu, aby nebyly příliš velké. Malá velikost mapového podkladu znamená, že je při zobrazení dané oblasti na mapě potřeba načíst více mapových podkladů, což zatěžuje paměť a procesor. Při velké velikosti mapového podkladu je pak v paměti zabráno zbytečně mnoho místa. Já jsem zvolil velikost mapového podkladu 640 x 640 pixelů. Velikost displeje, na kterém budu zobrazovat je 240 x 290 pixelů. Z tohoto rozměru plyne, že minimální počet načtených mapových podkladů je jeden a maximální čtyři.

4.3 Získání mapových podkladů

Získání mapových podkladů je jedna z nejdůležitějších věcí na této práci. Bez mapových podkladů by práce vůbec neměla smysl, protože by nebylo co zobrazovat a na čem testovat přesnost GPS souřadnic. Mapové podklady je možno získat dvěma způsoby. První a jednodušší, za to finančně velmi nákladný způsob, je mapy si jednoduše zakoupit. Tyto mapy pak můžete používat bez problémů pro komerční účely. Druhý způsob je stažení mapových podkladů z internetu z nějakého serveru např. mapy.cz[8] pomocí specializovaného software. Dle mého názoru jediný použitelný software pro stahování mapových podkladů je MapGrabber[9], který je shodou okolností napsaný také v .NET Frameworku. Tyto mapové podklady nemohou být užity pro komerční účely, ale pro vlastní účely ano, což pro tuto bakalářskou práci stačí.

4.3.1 MapGrabber

Program MapGrabber automaticky roluje, snímá, sestavuje a ukládá mapy (nebo jiný

obsah prohlížeče) do JPG souboru. [9]

4.3.1.1 Práce s MapGrabberem

- Přejít na webové stránky pro zachycení, např. www.mapy.cz.
- Vyhledat požadovanou oblast určenou k zachycení a přiblížit na požadované měřítko.
- Zadat parametry *Left / Top Corner* (viz. Obrázek 4.1) kliknutím na první tlačítko *Mark* a na webové stránce vybrat levý horní roh oblasti určené ke snímání.
- Pravý dolní roh oblasti se zadá ekvivalentně, pouze je potřeba kliknout na druhé tlačítko *Mark*.
- Dále se zadá spoždění *Delays Mouse Drag a Scrolling* (v sekundách). Je to doba, po kterou *MapGrabber* nebude snímat a posouvat. Hodnoty záleží na velikosti snímané plochy a rychlosti připojení.
- V *Capture X / Y Screens* se zadá počet snímání resp. počet posunutí v ose X a Y.
- V sekci *Output Image Settings* je potřeba nastavit hodnoty *Max output size X / Y Screens* podle toho, z kolika obrázků má být výsledná oblast poskládaná. Pokud jsou nastavené hodnoty 1 a 1, tak výsledná oblast bude rozdělena do tolika obrázků, kolik je posunů v ose X a Y. Pokud jsou nastavené stejné hodnoty jako u posunů, pak výsledná oblast bude pouze jeden obrázek.
- Nastavení atributů *Map Path a File format*. Jedná se o cestu, kam se mají obrázky uložit resp. výsledný formát obrázků.

Jediný problém při získávání mapových podkladů pomocí softwaru *MapGrabber* byl, že názvy uložených obrázků nebyly podle mých představ. Naštěstí je na stránkách *The Source Project* ke stažení zdrojový kód programu *MapGrabber*, kde je možné si kód upravit tak, aby vyhovoval potřebám práce. Já jsem použil již upravený *MapGrabber* od pana Bc. Viléma Jandy, který výsledné obrázky ukládá ve formátu `\Map001001`, kde první trojčíslí znamená hodnotu X a druhé trojčíslí znamená hodnotu Y.

4.4 Získání a uložení GPS souřadnic

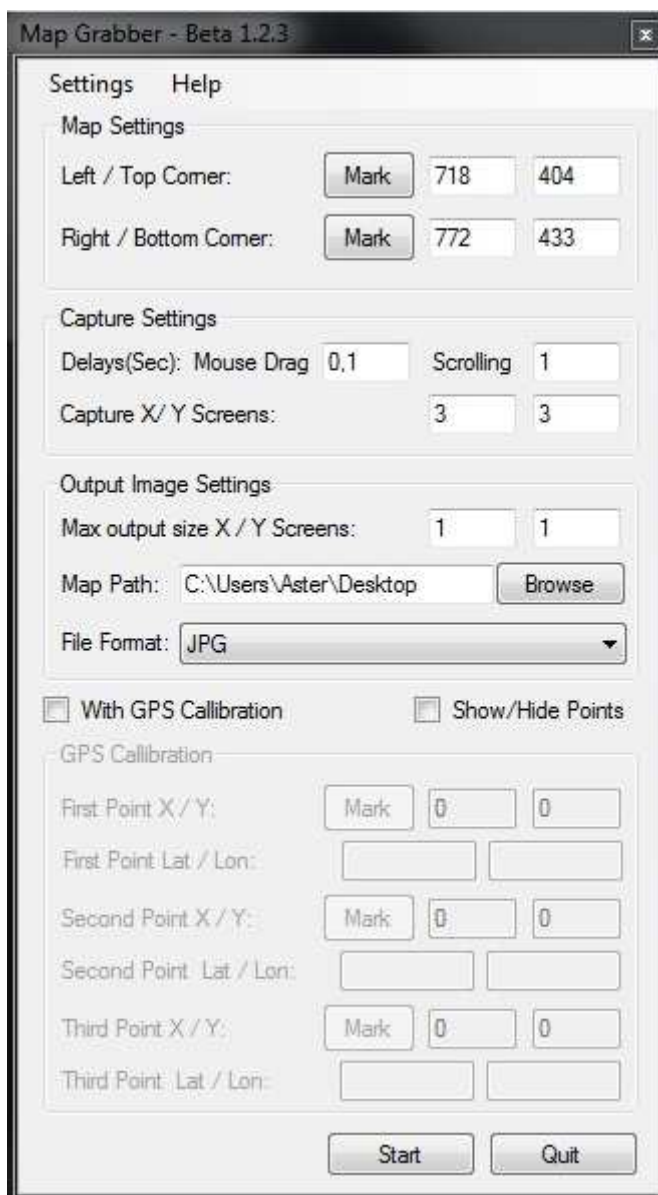
K získání a uložení GPS souřadnic se dá také použít program *MapGrabber*. Stačí v programu zaškrtnout volbu *With GPS Calibration*. Poté nastavit to samé jako v kapitole Práce s *MapGrabberem* a navíc ještě toto:

- Do pole *First Point X / Y* zadat polohu prvního bodu a do kolonky *First Point Lat / Lon* zadejte GPS souřadnice tohoto bodu.
- Do pole *Second Point X / Y* zadat polohu druhého bodu a do kolonky *Second Point Lat / Lon* zadejte GPS souřadnice tohoto bodu.

Problém spočívá v tom, že tyto výsledné GPS souřadnice nejsou příliš přesné. Na stránkách *The Source Project* je ke stažení zdrojový kód programu *MapGrabber*, kde je možné si kód upravit tak, aby vyhovoval potřebám práce. Já jsem použil upravený *MapGrabber* od pana Bc. Viléma Jandy, který upravil získávání GPS souřadnic k vyšší přesnosti. Zvolený formát uložení GPS souřadnic v souboru *GPSList.inf* je vidět na obrázku 4.2.

První řádek znamená, ke kterému obrázku záznam patří. Druhý řádek udává zeměpisnou šířku a zeměpisnou délku levého horního rohu obrázku. Třetí řádek udává zeměpisnou délku a zeměpisnou šířku pravého dolního rohu obrázku. Z tohoto formátu jsem

dále vycházel při načítání těchto hodnot v aplikaci MapyPDA.



Obrázek 4.1: Uživatelské rozhraní programu MapGrabber

```
\Map000000
0;0;14.400381944;49.645395556;
640;640;14.4181121821071;49.63388262535;
\Map001000
0;0;14.4181121821071;49.6454513000714;
640;640;14.4358424202143;49.6339383156054;
\Map002000
0;0;14.4358424202143;49.6455070441429;
640;640;14.4535726583214;49.6339940058607;
\Map003000
0;0;14.4535726583214;49.6455627882143;
640;640;14.4713028964286;49.6340496961161;
\Map004000
0;0;14.4713028964286;49.6456185322857;
640;640;14.4890331345357;49.6341053863714;
```

Obrázek 4.2: Formát souboru GPSList.inf

4.5 Metody zpřesnění

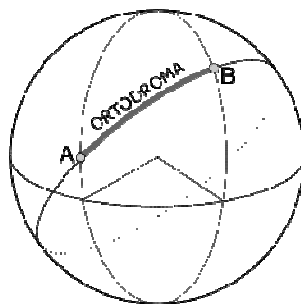
Nejdůležitější částí je bezpochyby získání co nejpřesnějších GPS souřadnic jednotlivých rohů mapových podkladů. Toho lze docílit v nejlepší případě tak, že se získají ručně, což je velmi náročné při představě, že počet mapových podkladů je v řádech stovek nebo tisíců. Čili jako snažší a méně přesné řešení se nabízí vytvořit jednoduchý program, který na vstup dostane přesné GPS souřadnice rohů celé mapy a na základě toho spočítá GPS souřadnice jednotlivých mapových podkladů. Tento případ není ideální z důvodu nelinearity GPS, ale já jej použiji v této bakalářské práci.

Další možnost, která mě napadla, je počítat polohu v systému UTM a pak převádět na GPS. Zde by byla potřeba naprogramovat konvertor z GPS na UTM a naopak. Poté by se v podstatě pokračovalo stejným způsobem.

Poslední uvedenou možností je zlinearizovat GPS souřadnice mapových podkladů. Toto řešení se jeví jako nejsložitější. Dalo by se toho docílit zkosením jednotlivých mapových podkladů tak, aby byly GPS souřadnice na mapových podkladech lineární.

4.6 Výpočet vzdálenosti dvou GPS souřadnic

K výpočtu vzdálenosti dvou GPS bodů na povrchu Země poslouží koule. Na povrchu koule lze počítat ortodroma, což je nejkratší vzdálenost mezi dvěma body (Obr. 4.3). [13]



Obrázek 4.3: Ortodroma

Určení délky ortodromy vychází ze sférické trigonometrie. Označme $|\varphi_1; \omega_1|$ a $|\varphi_2; \omega_2|$ souřadnice krajních bodů ortodromy a σ její délku. Délka se určí ze sférické kosinové věty takto: [13]

$$\sigma = \arccos(\sin\varphi_1\sin\varphi_2 + \cos\varphi_1\sin\varphi_2\cos(\omega_2 - \omega_1))$$

[13]

Vzdálenost bodů A a B (označeno jako d) se pak spočítá jako:

$$d = \sigma \cdot r$$

[13]

kde délku σ je v radiánech. Pro dosazení ve stupních by platilo:

$$d = \frac{2\pi}{360} \cdot \sigma \cdot r$$

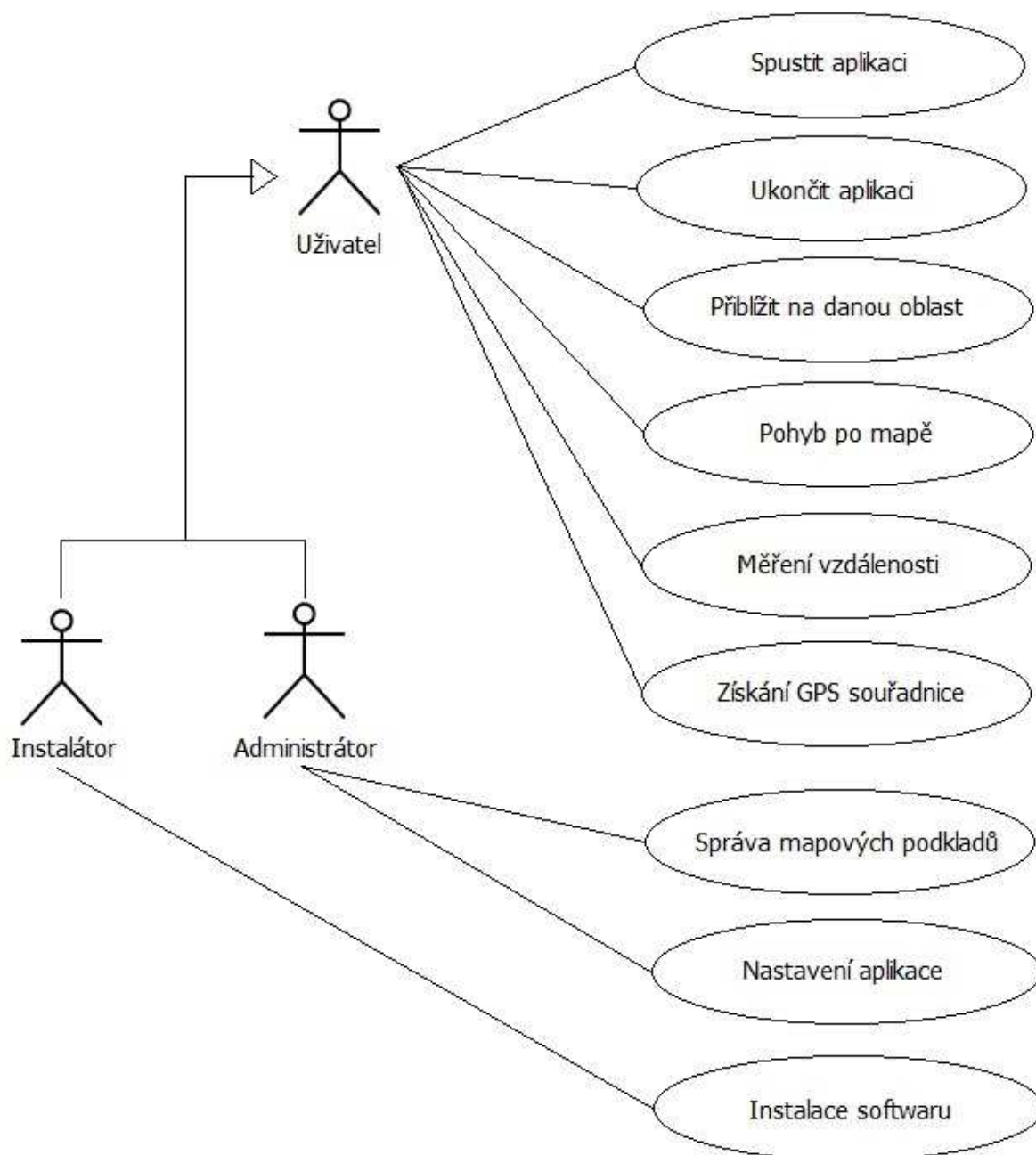
[13]

Hodnota r je poloměr Země v jednotkách, ve kterých je požadován výsledek.

5 Návrh řešení

Tato část popisuje návrh jednotlivých částí na základě analýzy z předchozí kapitoly. Jedná se o diagram užití, uživatelské rozhraní, třídy, zobrazení mapových podkladů a adresářovou strukturu aplikace.

5.1 Diagram užití

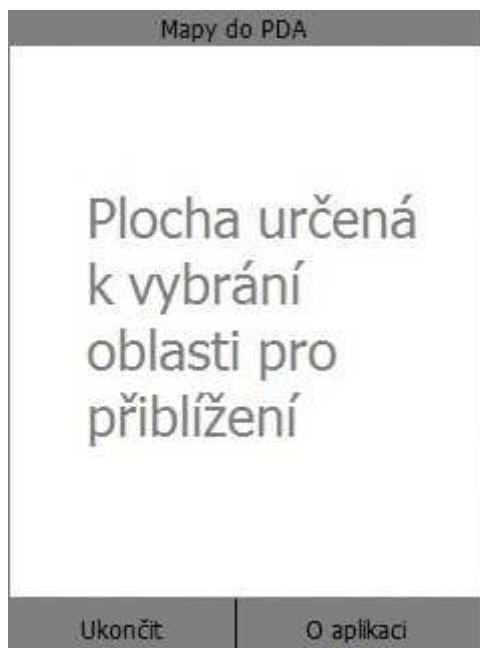


Obrázek 5.1: Diagram užití

5.2 Uživatelské rozhraní

Návrh uživatelského rozhraní aplikace MapyPDA je k vidění na obrázcích 5.2 a 5.3. Aplikace se skládá ze dvou oken (formulářů). Při spuštění aplikace se zobrazí úvodní okno (Obr. 5.2), kde si uživatel vybere oblast pro přiblížení a tato oblast se zobrazí v hlavním okně aplikace (Obr. 5.3), kde bude možné po mapě posouvat, určovat GPS souřadnice a vzdálenost konkrétních bodů na mapě.

5.2.1 Úvodní okno aplikace



Obrázek 5.2: Úvodní okno aplikace

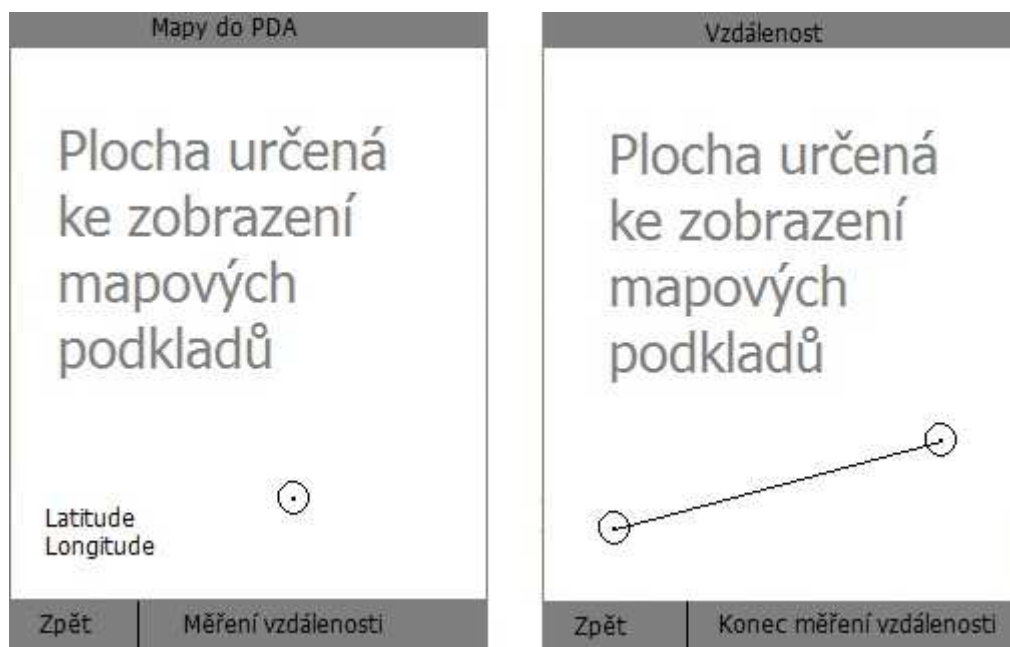
Úvodní okno (Obr.5.2) se zobrazí hned po spuštění aplikace. Toto okno obsahuje navigační lištu, která obsahuje dvě tlačítka, dále úvodní okno obsahuje panel pro vykreslení celé mapy a titulek s názvem aplikace.

První tlačítko v navigační liště se jmenuje *Ukončit* a slouží k ukončení celé aplikace, nikoliv pouze minimalizaci, jak je tomu u aplikací na Windows Mobile běžné. Druhé tlačítko v navigační liště se nazývá *O aplikaci* a po jeho stisku se zobrazí dialogové okno s informacema o aplikaci. Jedná se o popis aplikace, k čemu slouží, jméno autora, číslo verze a rok vzniku. Poslední částí úvodního okna je panel pro vykreslení celé mapy. Zde si uživatel vybere konkrétní místo pro přiblížení tak, že se tohoto místa dotkne stylusem a aplikace zobrazí hlavní okno (Obr. 5.3), které obsahuje další možnosti.

5.2.2 Hlavní okno aplikace

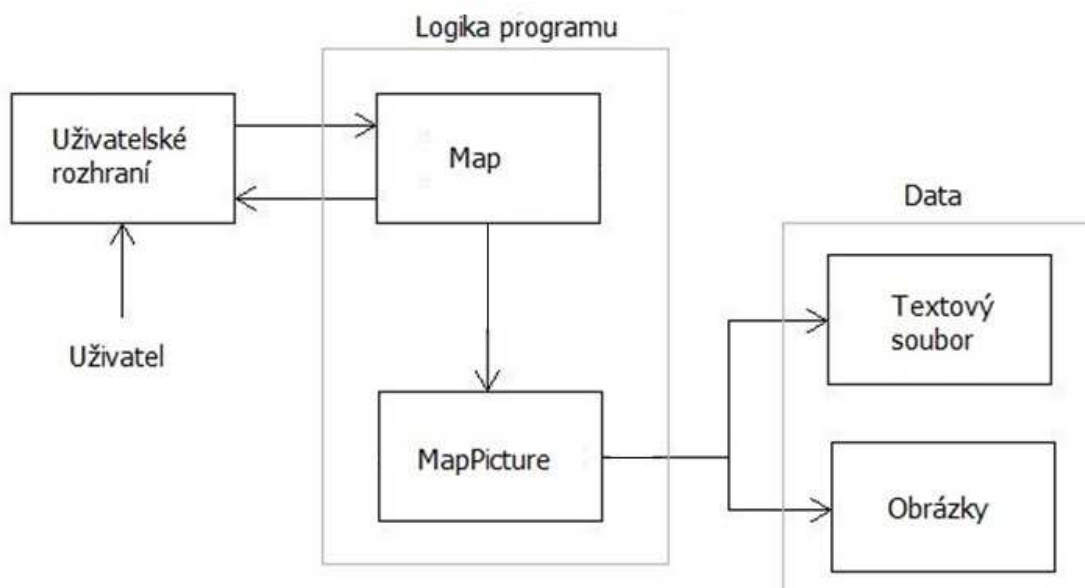
Hlavní okno aplikace (Obr. 5.3) nabízí dva režimy práce. První režim slouží k určování souřadnic a obsahuje titulek s názvem aplikace, label pro zobrazení souřadnic, panel k zobrazování mapových podkladů a navigační lištu, která se skládá ze dvou tlačítek. První tlačítko *Zpět* zavírá hlavní okno a zobrazuje úvodní okno. Druhé tlačítko *Měření vzdálenosti* přepíná do módu měření vzdálenosti dvou bodů na mapě. Vybrané místo na mapě je zobrazeno pomocí zaměřovače.

Druhý režim slouží k určování vzdálenosti dvou bodů na mapě. Obsahuje label určený k zobrazení vzdálenosti, panel k zobrazování mapových podkladů a navigační lištu, která se opět skládá ze dvou tlačítek. První tlačítko *Zpět* uzavírá hlavní okno a zobrazuje úvodní okno. Druhé tlačítko *Konec měření* přepíná zpět do módu určování GPS souřadnic. Vzdálenost na mapě je zobrazena přímkou, která spojuje dva zaměřovače, znázorňující místa na mapě, mezi kterými byla vzdálenost měřena.



Obrázek 5.3: Hlavní okno aplikace [8]

5.3 Blokové schéma programu



Obrázek 5.4: Blokové schéma programu

Aplikace je rozdělena do tří hlavních částí. První část je uživatelské rozhraní aplikace, druhá část je logika aplikace a třetí část jsou data užívané v aplikaci. Uživatel aplikace komunikuje prostřednictvím uživatelského rozhraní, které jeho příkazy akceptuje a zobrazuje mu výsledky. Uživatelské rozhraní komunikuje s logikou aplikace prostřednictvím rozhraní obsahující funkce jako posun po mapě, zobrazit mapové podklady apod. Data do logiky programu jsou načtena z textového souboru a složky mapových podkladů. Textový soubor obsahuje GPS souřadnice jednotlivých mapových podkladů.

5.4 Třídy

Základní třídy navržené v aplikaci jsou k vidění na obrázku 5.5. Při návrhu tříd jsem vycházel z objektů, které v aplikaci budu modelovat, a z jejich funkcí. Jedná se o objekt mapa, dále o objekt mapový obrázek, GPS souřadnice a dva formuláře pro grafické uživatelské rozhraní. Aplikace bude obsahovat pomocné třídy pro výpočet vzdálenosti, vykreslení terče při zaměření bodu a výpis informací o aplikaci. Dále jsem navrhl rozhraní, které bude obsahovat základní funkce, které by měla mapa obsahovat.



Obrázek 5.5: Diagram tříd

5.4.1 Třída Program

Třída `Program` je statická třída vygenerovaná Visual Studiem, obsahuje pouze jednu statickou metodu, která při spuštění aplikace vytváří instanci třídy `IntroForm`.

5.4.2 Třída IntroForm

Třída `IntroForm` představuje výchozí formulář při spuštění aplikace. Tato třída bude vytvářet instanci třídy `Map` a volat její metodu pro přiblížení na danou oblast na mapě.

5.4.3 Třída Map

Třída `Map` představuje celou mapu určenou ke zobrazení. Obsahuje důležité vlastnosti, jako velikost jednoho obrázku ke zobrazení, velikost displeje, místo pro vykreslení. Dále kolekci obrázků typu `MapPicture`, uchovávající obrázky, které se mají vykreslit a velikost posunu po mapě. Obsahuje také metody, jako posunutí mapy vpravo, vlevo, nahoru a dolů, získání GPS souřadnice, přiblížení na danou oblast a zobrazení mapy.

5.4.4 Třída MapsForm

Třída `MapsForm` představuje formulář, který se spustí při výběru místa pro přiblížení. Tento formulář obsahuje události vznikající při stisku tlačítka vpravo, vlevo, nahoru a dolů, přičemž se volají metody ze třídy `Map`, které provedou konkrétní posuny po mapě. Dále jsou zde události vznikající při označení místa na mapě, při stisku konkrétních tlačítek apod.

5.4.5 Třída MapPicture

Třída `MapPicture` představuje jeden konkrétní mapový podklad. Obsahuje vlastnosti jako jeho pozici, konkrétní mapový podklad, GPS souřadnice apod.

5.4.6 Třída Gps

Třída `Gps` slouží jako obal pro hodnoty latitude (zeměpisná šířka) a longitude (zeměpisná délka). Třída obsahuje statické metody určené k převedení GPS souřadnic z formátu ve stupních na formát ve stupních, minutách a vteřinách.

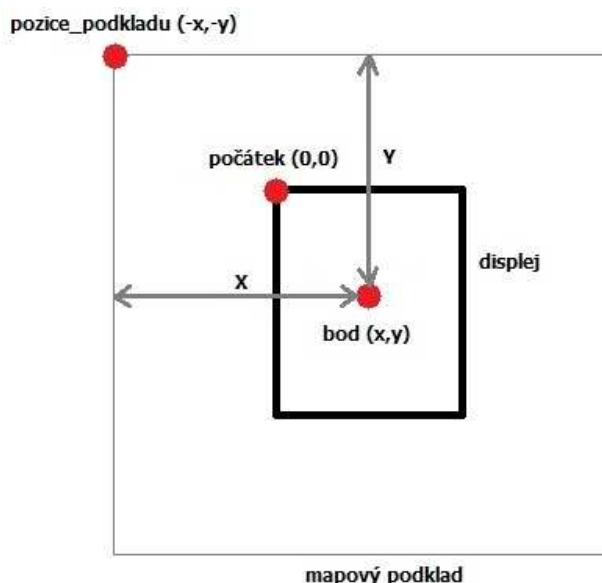
5.4.7 Rozhraní IMap

Rozhraní `IMap` obsahuje základní metody, které by měla každá mapa implementovat. Jedná se o vykreslení mapových podkladů, kolekce mapových podkladů, posun po mapě apod.

5.5 Souřadnicový systém

V .NET frameworku má souřadnicový systém počátek v levém horním rohu a kladné hodnoty jsou ve směrech doprava a dolů. Z tohoto systému jsem vycházel při návrhu souřadnicového systému do aplikace MapyPDA.

Velikost jednoho mapového podkladu je 640 x 640 pixelů a velikost plochy určené ke zobrazení je 290 x 240 pixelů. Počátek displeje a počátek mapového podkladu jsou ve stejném bodě (levý horní roh displeje). Při zobrazení hýbeme počátkem mapového podkladu. Cílem je získat souřadnice X a Y s počátkem v levém horním rohu mapového podkladu.



Obrázek 5.6: Souřadnicový systém aplikace

Výpočet souřadnic X a Y na mapovém podkladu:

$$X = bod(x) - pozice_podkladu(x)$$

$$Y = bod(y) - pozice_podkladu(y)$$

5.5.1 Výpočet GPS souřadnice bodu na mapě

Zeměpisná šířka levého horního rohu mapového podkladu

$$Lat_{levý_horní}$$

Zeměpisná šířka pravého dolního rohu mapového podkladu

$$Lat_{pravý_dolní}$$

Zeměpisná délka levého horního rohu mapového podkladu

$$Lon_{levý_horní}$$

Zeměpisná délka pravého dolního rohu mapového podkladu

$$Lon_{pravý_dolní}$$

Rozdíl levého horního a pravého dolního rohu mapového podkladu

$$Lat_{rozdíl} = Lat_{pravý_dolní} - Lat_{levý_horní}$$

$$Lon_{rozdíl} = Lon_{pravý_dolní} - Lon_{levý_horní}$$

5.5.1.1 Výpočet zeměpisné šířky a délky:

$$Lat = \frac{Lat_{rozdíl}}{velikost_podkladu(y)} \cdot Y + Lat_{levý_horní} + \varepsilon_{lat}$$

$$Lon = \frac{Lon_{rozdíl}}{velikost_podkladu(x)} \cdot X + Lon_{levý_horní} + \varepsilon_{lon}$$

Zeměpisná šířka (Latitude) – rozdíl zeměpisné šířky pravého dolního rohu mapového podkladu s levým horním rohem se vydělí velikostí mapového podkladu Y. Tím se získá hodnota zeměpisné šířky jednoho pixelu. Tuto hodnotu je potřeba vynásobit polohou Y, čímž dostaneme výsledek, ke kterému je potřeba ještě přičíst zeměpisnou šířku levého horního rohu mapového podkladu. Hodnota ε_{lat} znamená zpřesnění zeměpisné šířky.

Zeměpisná délka (Longitude) – rozdíl zeměpisné délky pravého dolního rohu mapového podkladu s levým horním rohem se vydělí velikostí mapového podkladu X. Tím se získá hodnota zeměpisné délky jednoho pixelu. Tuto hodnotu je potřeba vynásobit polohou X, čímž dostaneme výsledek, ke kterému ještě přičteme zeměpisnou délku levého horního rohu mapového podkladu. Hodnota ϵ_{lon} znamená zpřesnění zeměpisné délky.

5.5.1.2 Zpřesnění GPS souřadnice na mapě

Hodnoty ϵ_{lat} a ϵ_{lon} zpřesňují výslednou GPS souřadnici. Jedná se o to, že pokud naměřený výsledek dosahuje hodnoty 1.8 a požadovaná hodnota je 1.9, pak se k naměřenému výsledku připočte hodnota 0.1. Hodnoty ϵ_{lat} a ϵ_{lon} samozřejmě nelze zvolit takto snadno, ale princip je zřejmý. Správné hodnoty se určí na základě statistiky naměřených hodnot.

5.5.2 Výpočet polohy na mapě

Ze zadané zeměpisné délky (longitude) a šířky (latitude) bodu na mapě je možné vypočítat polohu tohoto bodu určeného ke zobrazení v aplikaci následujícím způsobem.

Číslo mapového podkladu na němž se bod nachází:

$$Lon_{rozdl} = \frac{Lon_{pravý_dolní} - Lon_{levý_horní}}{\text{počet podkladu } X}$$

$$Lat_{rozdl} = \frac{Lat_{levý_horní} - Lat_{pravý_dolní}}{\text{počet podkladu } Y}$$

$$\text{Číslo}_x = \frac{\text{longitude} - Lon_{levý_horní}}{Lon_{rozdl}}$$

$$\text{Číslo}_y = \frac{Lat_{levý_horní} - \text{latitude}}{Lat_{rozdl}}$$

Číslo mapového podkladu X a Y je celá část výsledků Číslo_x a Číslo_y . Jinak řečeno tyto výsledky se zaokrouhlí směrem dolů.

Výpočet polohy na mapovém podkladu:

$$Poloha_x = \frac{\text{longitude} - Lon_{levý_horní}}{\frac{Lon_{pravý_dolní} - Lon_{levý_horní}}{\text{velikost podkladu } X}} + \gamma_{lon}$$

$$Poloha_y = \frac{Lat_{levý_horní} - \text{latitude}}{\frac{Lat_{levý_horní} - Lat_{pravý_dolní}}{\text{velikost podkladu } Y}} + \gamma_{lat}$$

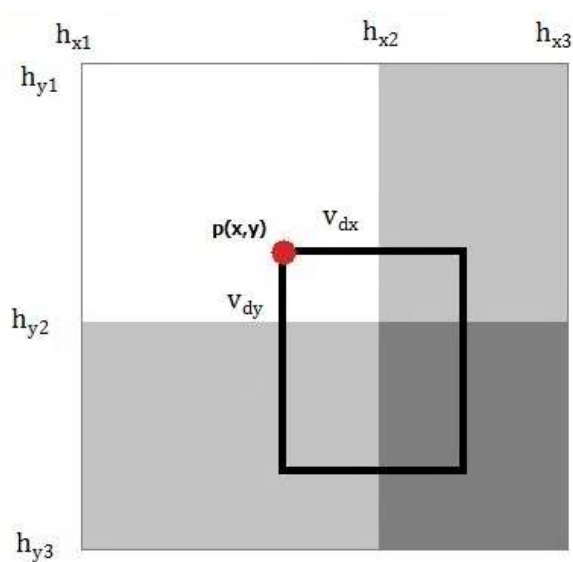
5.6 Zobrazení mapových podkladů

Při zobrazení mapových podkladů mohou nastat tyto tři případy:

- Bude zobrazen pouze jeden mapový podklad.
- Budou zobrazeny dva mapové podklady.
- Budou zobrazeny čtyři mapové podklady.

Otázka zní, jak se pozná, kolik a jaké mapové podklady mají být přidány do kolekce? Tato odpověď je jistě záludná a já jí zodpovím dále v tomto textu.

První věc je určit hranice (Obr. 5.7), podle kterých se zjistí, zda-li je potřeba načítat nebo odebírat mapový podklad z kolekce. Velikost mapového podkladu X a Y je označena jako v_{ox} a v_{oy} . Velikost displeje X a Y je označena jako v_{dx} a v_{dy} . Počátek displeje je označen jako $p(x,y)$. Počátek p se bude měnit posunem po mapě. Hranice ve směru X jsou tři: h_{x1} , h_{x2} a h_{x3} . Ve směru Y jsou také tři: h_{y1} , h_{y2} a h_{y3} .



Obrázek 5.7: Hranice mapového podkladu

Hodnota hranic je následující:

$$h_{x1} = 0$$

$$h_{x2} = v_{ox} - v_{dx}$$

$$h_{x3} = v_{ox}$$

$$h_{y1} = 0$$

$$h_{y2} = v_{oy} - v_{dy}$$

$$h_{y3} = v_{oy}$$

5.6.1 Zobrazení jednoho mapového podkladu

Zobrazení jednoho mapového podkladu nastane tehdy, pokud se počátek displeje $p(x,y)$ nachází v bílé ploše na obrázku 5.7. To znamená, pokud hodnota X počátku p je větší nebo rovna h_{x1} a zároveň menší než hodnota hranice h_{x2} , což je rozdíl velikosti mapového podkladu ve směru X a velikosti displeje ve směru X . Zároveň s tímto musí platit, že hodnota Y počátku p je větší nebo rovna h_{y1} a zároveň menší než hodnota hranice h_{y2} . Formální zápis vypadá následovně:

```
If p(x) >= hx1 and p(x) < hx2 and p(y) >= hy1 and p(y) < hy2
```

```
Then
```

```
    Přidej jeden podklad příslušných parametrů do kolekce a zobraz
```

5.6.2 Zobrazení dvou mapových podkladů

Zobrazení dvou mapových podkladů nastane tehdy, pokud se počátek $p(x,y)$ nachází ve světle šedivých plochách na obrázku 5.7. Pro první plochu to znamená, pokud $p(x)$ je větší nebo rovno h_{x2} a zároveň $p(x)$ je menší než h_{x3} . Dále musí platit, že $p(y)$ je větší nebo rovno h_{y1} a zároveň $p(y)$ je menší než h_{y2} . Pro druhou plochu musí platit, že $p(x)$ je větší nebo rovno h_{x1} a zároveň $p(x)$ je menší než h_{x2} . Musí také platit, že $p(y)$ je větší nebo rovno h_{y2} a zároveň $p(y)$ je menší než h_{y3} . Formální zápis vypadá následovně:

```
If p(x) >= hx2 and p(x) < hx3 and p(y) >= hy1 and p(y) < hy2
```

```
Then
```

```
    Přidej do kolekce dva podklady příslušných parametrů a zobraz
```

```
Else if p(x) >= hx1 and p(x) < hx2 and p(y) >= hy2 and p(y) < hy3
```

```
Then
```

```
    Přidej do kolekce dva podklady příslušných parametrů a zobraz
```

5.6.3 Zobrazení čtyř mapových podkladů

Zobrazení čtyř mapových podkladů nastane tehdy, pokud se počátek $p(x,y)$ nachází v tmavě šedé oblasti na obrázku 5.7. Znamená to, že $p(x)$ je větší nebo rovno h_{x2} a zároveň $p(x)$ je menší než h_{x3} . Současně musí platit, že $p(y)$ je větší nebo rovno h_{y2} a zároveň $p(y)$ je menší než h_{y3} . Formální zápis vypadá takto:

```
If p(x) >= hx2 and p(x) < hx3 and p(y) >= hy2 and p(y) < hy3
```

```
Then
```

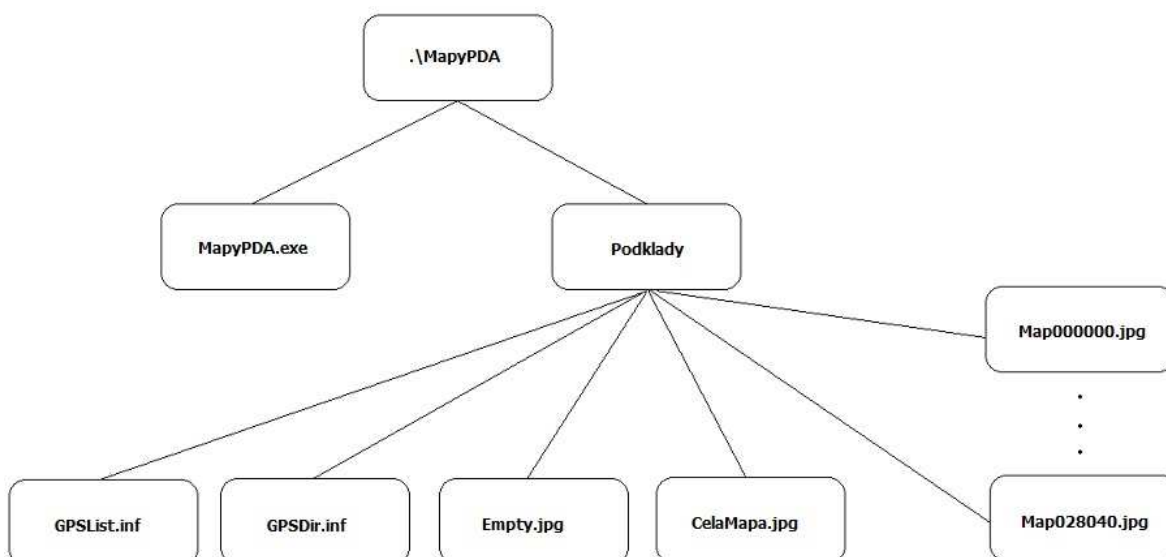
```
    Přidej do kolekce čtyři podklady příslušných parametrů  
    a zobraz
```

Závěrem bych jen podotkl, že to, jaký mapový podklad se má do kolekce uložit a jaký se má z kolekce odebrat, a jak na sebe navazují jejich souřadnice je součástí implementace a popíši to v další kapitole.

5.7 Adresářová struktura

V aplikaci Mapy do PDA je potřeba ukládat mapové podklady a GPS souřadnice. Mapové podklady mají název *Map000000.jpg* až *Map028040.jpg*. GPS souřadnice jsou uloženy do dvou souborů a to *GPSDir.inf*, který v sobě uchovává GPS souřadnice levého horního rohu a pravého dolního rohu celé mapy. Druhým souborem je *GPSList.inf*, který v sobě uchovává GPS souřadnice levého horního rohu a pravého dolního rohu jednotlivých mapových podkladů jak můžete vidět na obrázku 4.2.

Adresářovou strukturu aplikace si můžete prohlédnout na obrázku 5.8. Kořenový adresář aplikace se nazývá *MapyPDA*. Tento adresář obsahuje spouštěcí soubor aplikace *MapyPDA.exe* a adresář *MapyTaborsko*. Adresář *Podklady* obsahuje soubory *GPSList.inf*, *GPSDir.inf* určené k uložení GPS souřadnic. Dále obsahuje soubor *Empty.jpg*, který se zobrazí, pokud posuneme do oblasti mimo mapu a soubor *CelaMapa.jpg*, který se zobrazí v úvodním okně aplikace (Obr. 5.2) pro výběr dané oblasti. Nakonec obsahuje soubory *Map000000.jpg* až *Map028040.jpg*, což jsou mapové podklady určené ke zobrazení v aplikaci *MapyPDA*.



Obrázek 5.8: Adresářová struktura aplikace

6 Implementace řešení

Tato část obsahuje implementaci jednotlivých částí z návrhu řešení. Jedná se o aplikaci jako projekt, dále o podrobný popis jednotlivých tříd a algoritmů.

6.1 Projekt

Tato část popisuje vytvoření projektu MapyPDA v prostředí Visual Studio 2008, kompilaci a nasazení do PDA.

6.1.1 Vytvoření projektu

Ve Visual Studiu 2008 je třeba založit nový projekt výběrem *File -> New -> Project (Ctrl + Shift + N)*. Otevře se dialog, ve kterém je zvolen typ projektu *Smart Device*, což znamená, že se jedná o projekt určený do kapesních zařízení. Dále je třeba vybrat verzi .NET Frameworku 3.5, do kolonky *Name* zadat název projektu a do kolonky *Location* cestu na disk, kam se projekt uloží. Ve druhém dialogovém okně se zvolí cílová platforma, verze .NET Compact Frameworku a šablonu *Device Application*, což je formulářová aplikace. Potvrzením dialogu se vytvoří nový projekt.

6.1.2 Nasazení projektu

Před zkompileváním projektu je potřeba vytvořit v PDA složku s názvem projektu a v ní vytvořit další složku s mapovými podklady. V defaultním nastavení se projekt zkompileje do složky Program Files v PDA, čili je třeba složky vytvořit tam. Poté je nutno připojit PDA pomocí kolébky a *Windows Mobile Device Center* k PC a nakopírovat do složky s mapama mapové podklady. Teď už stačí ve Visual Studiu zkompileovat projekt tímto zvolením možností *Debug -> Start Debugging (F5)*. V dialogovém okně vybrat zařízení, na které se má projekt zkompileovat a stiskem tlačítka *Deploy* potvrdit. Tím se do PDA zkompileje a zavede projekt a může být spuštěn pomocí exe souboru.

6.2 Třídy

Kompletní diagram tříd je k vidění na obrázku 6.1.

6.2.1 Třída Program

Třída `Program` je výchozí bod aplikace.

6.2.1.1 Metody

- `static void Main()`
Vytváří instanci třídy `IntroForm`.

6.2.2 Třída IntroForm

Třída `IntroForm` představuje úvodní formulář.

6.2.2.1 Konstruktor

- `public IntroForm()`
Inicializuje formulář.

6.2.2.2 Vlastnosti

- `public int X`
Vlastnost pro čtení a zápis. Jedná se o pozici X na displeji.
- `public int Y`
Vlastnost pro čtení a zápis. Jedná se o pozici Y na displeji.

6.2.2.3 Metody

- `private void map_ShowMap(object sender, ShowMapEventArgs e)`
Obsluha události ShowMap, vytváří instanci třídy `MapsForm` a zobrazuje dialog.
- `private void menuItem2_Click(object sender, EventArgs e)`
Obsluha události Click, uzavře formulář po vybrání položky *konec* v menu.
- `private void menuItem4_Click(object sender, EventArgs e)`
Obsluha události Click, vypíše informace o aplikaci vybráním položky *o aplikaci* v menu.
- `private void pictureBox1_MouseDown(object sender, MouseEventArgs e)`
Obsluha události MouseDown, uloží aktuální pozici do proměnných `_x` a `_y`.
- `private void pictureBox1_MouseUp(object sender, MouseEventArgs e)`
Obsluha události MouseUp, vytvoří mapu a přiblíží na danou oblast.

6.2.3 Třída MapsForm

Třída `MapsForm` představuje formulář určený ke zobrazení mapových podkladů a k jejich manipulaci.

6.2.3.1 Konstruktor

- `public MapsForm(IEnumerable<MapPicture> collection, IMap map)`
Inicializuje formulář, parametr `collection` je kolekce obrázků určená k počátečnímu zobrazení, parametr `map` představuje mapu, se kterou formulář pracuje.

6.2.3.2 Vlastnosti

- `private bool Distance`
Vlastnost pro čtení a zápis. Určuje, zda jde o režim měření vzdálenosti nebo určování GPS souřadnic.
- `private Distance Dm`
Vlastnost pro čtení a zápis. Slouží k výpočtu vzdálenosti dvou GPS souřadnic.
- `public IMap Map`
Vlastnost pro čtení a zápis. Jedná se o mapu, se kterou formulář pracuje.
- `private int X`
Vlastnost pro čtení a zápis. Pozice X na displeji.

- `private int Y`
Vlastnost pro čtení a zápis. Pozice Y na displeji.

6.2.3.3 Metody

- `private void MapsForm_KeyDown(object sender, KeyEventArgs e)`
Obsluha události `KeyDown`. Jedná se o tlačítka nahoru, dolů, doprava a doleva obsahující funkce pro pohyb po mapě.
- `private void menuItem1_Click(object sender, EventArgs e)`
Obsluha události `Click`. Uzavře formulář vybráním položky `menuItem1`.
- `private void menuItem2_Click(object sender, EventArgs e)`
Obsluha události `Click`. Určuje, zda jde o režim měření vzdálenosti nebo určení GPS souřadnic vybráním položky `menuItem2`.
- `private void pictureBox1_Paint(object sender, PaintEventArgs e)`
Obsluha události `Paint`. Vykresluje mapové podklady.
- `private void pictureBox1_MouseDown(object sender, MouseEventArgs e)`
Obsluha události `MouseDown`. Zjistí, zda jde o režim měření vzdálenosti nebo určení GPS souřadnic a vypíše požadovaný výsledek na displej.
- `private void ShowGps()`
Zobrazí GPS souřadnice na displej prostřednictvím labelů.
- `private void LabelVisible(bool visible)`
Viditelnost labelů určená parametrem `visible`.
- `private void Line(int x1, int x2, int y1, int y2)`
Vykreslí čáru spojující dva body na mapě. Parametry `x1`, `x2`, `y1`, `y2` jsou pozice X a Y prvního a druhého bodu.
- `private void PaintLineTarget()`
Vykreslí zaměřovač sloužící k označení místa na mapě.

6.2.4 Třída Map

Třída představuje mapu, její vlastnosti a metody. Implementuje rozhraní `IMap`.

6.2.4.1 Konstruktor

- `public Map(Point pictureSize, Point displaySize, PictureBox rendering)`
Konstruktor vytváří mapu. Parametr `pictureSize` je velikost jednoho mapového podkladu, parametr `displaySize` je velikost displeje určeného ke zobrazení a parametr `rendering` je místo, kam se mapa vykreslí.

6.2.4.2 Vlastnosti

- `private Point PictureSize`
Vlastnost pro čtení a zápis. Velikost mapového podkladu.
- `private Point DisplaySize`
Vlastnost pro čtení a zápis. Velikost displeje.
- `public IEnumerable<MapPicture> Collection`

Vlastnost pro čtení a zápis. Kolekce obsahuje mapové podklady určené k vykreslení.

- `public int Shift`
Vlastnost pro čtení a zápis. Velikost posunu po mapě.
- `public PictureBox Rendering`
Vlastnost pro čtení a zápis. Místo, kam se má mapa vykreslit.

6.2.4.3 Metody

- `private void SetPositionX(int shift)`
Nastavuje pozici X mapových podkladů. Parametr `shift` je velikost posunu. Kladná hodnota parametru `shift` pozici zvyšuje, záporná snižuje.
- `private void SetPositionY(int shift)`
Nastavuje pozici Y mapových podkladů. Parametr `shift` je velikost posunu. Kladná hodnota parametru `shift` pozici zvyšuje, záporná snižuje.
- `private IEnumerable<MapPicture> SetRight()`
Metoda přidává mapové podklady do kolekce při pohybu vpravo. Vrací kolekci mapových podkladů.
- `private IEnumerable<MapPicture> SetLeft()`
Metoda přidává mapové podklady do kolekce při pohybu vlevo. Vrací kolekci mapových podkladů.
- `private IEnumerable<MapPicture> SetUp()`
Metoda přidává mapové podklady do kolekce při pohybu nahoru. Vrací kolekci mapových podkladů.
- `private IEnumerable<MapPicture> SetDown()`
Metoda přidává mapové podklady do kolekce při pohybu dolů. Vrací kolekci mapových podkladů.
- `public void DisplayLocation(int xArgs, int yArgs)`
Zobrazí konkrétní oblast na mapě. Parametry `xArgs`, `yArgs` jsou pozice ve směru X a Y na displeji.
- `public Gps GetGps(int X, int Y)`
Získá GPS pozici ze zadaných parametrů a tuto pozici vrátí.
- `public void RenderDown()`
Vykreslí mapové podklady při posunu dolů.
- `public void RenderUp()`
Vykreslí mapové podklady při posunu nahoru.
- `public void RenderLeft()`
Vykreslí mapové podklady při posunu doleva.
- `public void RenderRight()`
Vykreslí mapové podklady při posunu doprava.
- `public void RenderPictures()`
Metoda vykresluje aktuální mapové podklady z kolekce.
- `private void OnShowMap(ShowMapEventArgs e)`
Metoda vyvolá událost `ShowMap`. Parametr `e` jsou data nesená událostí.

6.2.4.4 Události

- `public event ShowMapEventHandler ShowMap;`
Událost nastane před zobrazením pozice na mapě.

6.2.5 Třída MapPicture

Třída `MapPicture` představuje konkrétní mapový podklad.

6.2.5.1 Konstruktor

- `public MapPicture(int positionX, int positionY, Point img)`
Konstruktor vytváří konkrétní mapový podklad ze zadaných parametrů. Parametr `positionX` je pozice X horního levého rohu zobrazené části mapového podkladu, parametr `positionY` je pozice Y, parametr `img` je číslo mapového podkladu.

6.2.5.2 Vlastnosti

- `public Gps GpsLeftTop`
Vlastnost pro čtení a zápis. GPS souřadnice levého horního rohu mapového podkladu.
- `public Gps GpsRightBottom`
Vlastnost pro čtení a zápis. GPS souřadnice pravého dolního rohu mapového podkladu.
- `public Point Img`
Vlastnost pro čtení a zápis. Číslo mapového podkladu.
- `public int PositionX`
Vlastnost pro čtení a zápis. Aktuální pozice X levého horního rohu zobrazené části mapového podkladu.
- `public int PositionY`
Vlastnost pro čtení a zápis. Aktuální pozice Y levého horního rohu zobrazené části mapového podkladu.
- `public Image Image`
Vlastnost pro čtení a zápis. Obrázek namapovaný na konkrétní mapový podklad.

6.2.5.3 Metody

- `private void SetGps(int x, int y)`
Metoda nastavuje GPS souřadnice pravého dolního a levého horního rohu mapovému podkladu. Parametry `x` a `y` jsou číslo mapového podkladu.
- `private void SetImage(int x, int y)`
Metoda přiřazuje k mapovému podkladu obrázek. Parametry `x` a `y` jsou číslo mapového podkladu.
- `public static Point GetSize()`
Metoda získá rozměry mapového podkladu připojeného k programu a vrátí jejich hodnotu.

6.2.6 Třída Gps

Třída `Gps` slouží k definování GPS souřadnice.

6.2.6.1 Konstruktor

- `public Gps(double longitude, double latitude, Point img)`

Konstruktor vytváří GPS souřadnici spojenou s mapovým podkladem, na kterém se souřadnice nachází. Parametr `longitude` je zeměpisná délka, parametr `latitude` je zeměpisná šířka a parametr `img` je číslo mapového podkladu.

6.2.6.2 Vlastnosti

- `public Point Img`
Slouží k nastavení a získání čísla mapového podkladu patřící k dané GPS souřadnici.
- `public double Longitude`
Slouží k nastavení a získání zeměpisné délky.
- `public double Latitude`
Slouží k nastavení a získání zeměpisné šířky.

6.2.6.3 Metody

- `public static string LongitudeToDegree(double longitude, double difference)`
Statická metoda, převádí zeměpisnou délku z formátu ve stupních na formát ve stupních, minutách a vteřinách a tento výsledek vrací. Parametr `difference` je odchylka kompenzující nepřesnost GPS.
- `public static string LatitudeToDegree(double latitude, double difference)`
Statická metoda, převádí zeměpisnou šířku z formátu ve stupních na formát ve stupních, minutách a vteřinách a tento výsledek vrací. Parametr `difference` je odchylka kompenzující nepřesnost GPS.

6.2.7 Třída Distance

Třída `Distance` se stará o výpočet vzdálenosti dvou GPS souřadnic.

6.2.7.1 Konstruktor

- `public Distance()`
Konstruktor nastavuje hodnoty na nulu.

6.2.7.2 Vlastnosti

- `public double Lat1`
Vlastnost pro čtení a zápis. Zeměpisná šířka prvního bodu.
- `public double Lat2`
Vlastnost pro čtení a zápis. Zeměpisná šířka druhého bodu.
- `public double Lon1`

Vlastnost pro čtení a zápis. Zeměpisná délka prvního bodu.

- `public double Lon2`

Vlastnost pro čtení a zápis. Zeměpisná délka druhého bodu.

- `public int Counter`

Vlastnost pro čtení a zápis. Počítadlo kolikrát uživatel klikne.

- `public int Xtarget`

Vlastnost pro čtení a zápis. Souřadnice X zaměřovače.

- `public int Ytarget`

Vlastnost pro čtení a zápis. Souřadnice Y zaměřovače.

6.2.7.3 Metody

- `public void SetNull()`

Metoda nastavuje proměnné třídy na nulu.

- `public double DistanceVector()`

Metoda počítá vzdálenost dvou bodů na mapě a tento výsledek vrací.

- `private double DegreeToRadian(double angle)`

Metoda převádí stupně na radiány a tento výsledek vrací. Parametr `angle` je úhel ve stupních.

- `private double RadianToDegree(double angle)`

Metoda převádí stupně na radiány a tento výsledek vrací. Parametr `angle` je úhel v radiánech.

6.2.8 Třída About

Statická třída `About`. Stará se o vypsání informací o aplikaci.

6.2.8.1 Metody

- `public static void AboutApp()`

Statická metoda, vypisuje informace o aplikaci.

6.2.9 Třída Target

Statická třída `Target`. Stará se o vykreslení zaměřovače.

6.2.9.1 Metody

- `public static void PaintTarget(int x, int y, Color color, PictureBox where, Size size)`

Statická metoda, vykresluje zaměřovač, parametry `x` a `y` jsou pozice X a Y, parametr `color` je barva, parametr `where` je kam se má vykreslit a parametr `size` je velikost zaměřovače.

6.2.10 Třída ShowMapEventArgs

Třída `ShowMapEventArgs` představuje data nesená objektem, který událost vyvolal.

6.2.10.1 Konstruktor

- `public ShowMapEventArgs(IEnumerable<MapPicture> collection)`
Konstruktor vytváří instanci třídy, parametr `collection` je kolekce mapových podkladů.

6.2.10.2 Vlastnosti

- `public IEnumerable<MapPicture> Collection`
Vlastnost pro čtení a zápis. Kolekce mapových podkladů.

6.2.11 Rozhraní IMap

Rozhraní `IMap` obsahuje hlavičky základních metod, vlastností a událostí pro práci s mapou.

6.2.11.1 Vlastnosti

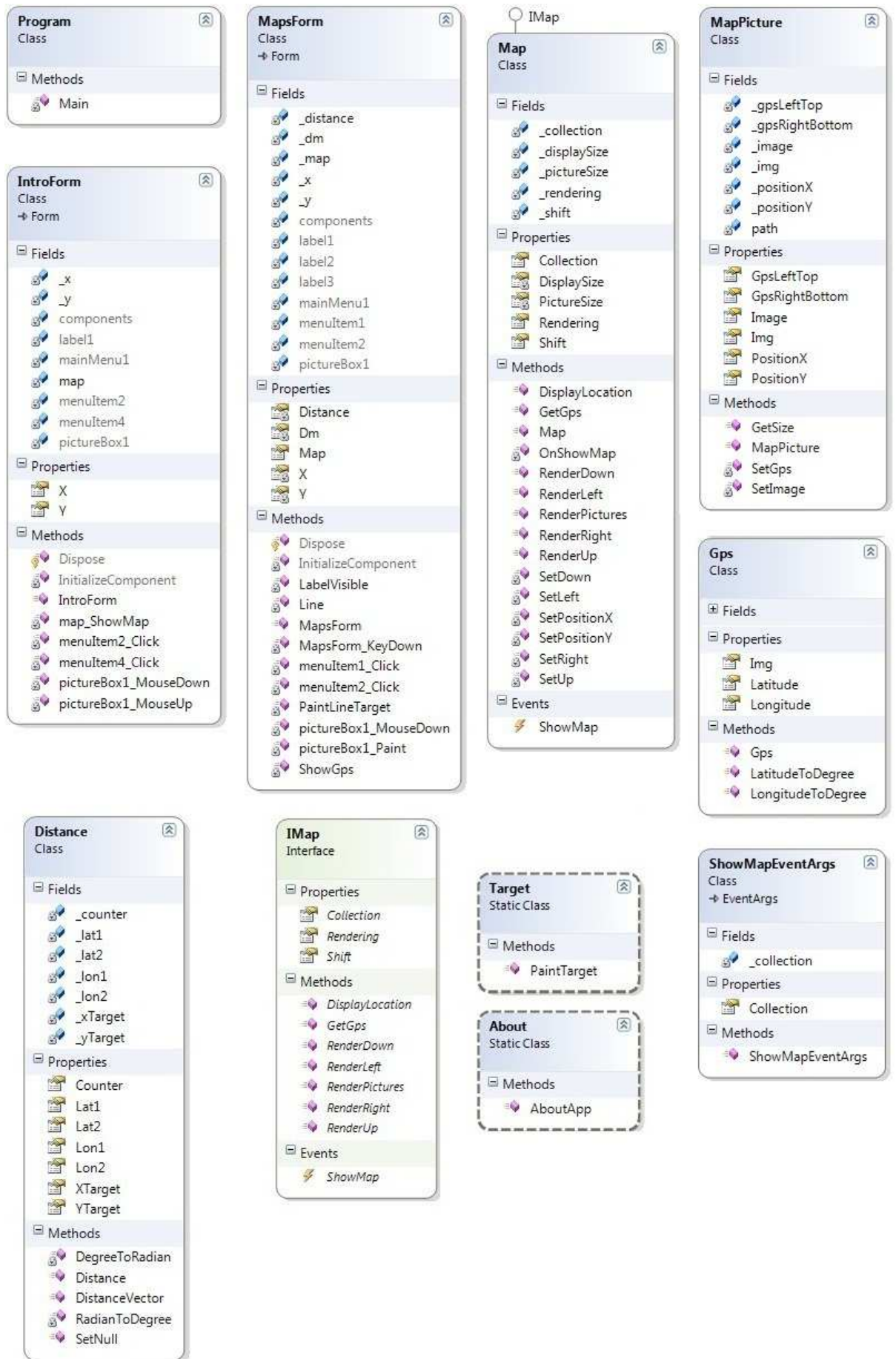
- `PictureBox` Rendering
Místo, kam se má mapa vykreslit.
- `int` Shift
Velikost posunu po mapě.
- `IEnumerable<MapPicture> Collection`
Kolekce mapových podkladů určených k vykreslení.

6.2.11.2 Metody

- `void DisplayLocation(int xArgs, int yArgs);`
Zobrazí konkrétní oblast na mapě. Parametry `xArgs`, `yArgs` jsou pozice ve směru X a Y na displeji.
- `Gps GetGps(int X, int Y);`
Získá GPS pozici ze zadaných parametrů a tuto pozici vrátí.
- `void RenderDown();`
Vykreslí mapové podklady při posunu dolů.
- `void RenderUp();`
Vykreslí mapové podklady při posunu nahoru.
- `void RenderLeft();`
Vykreslí mapové podklady při posunu doleva.
- `void RenderRight();`
Vykreslí mapové podklady při posunu doprava.
- `void RenderPictures();`
Metoda vykreslí mapové podklady.

6.2.11.3 Události

- `event ShowMapEventHandler ShowMap;`
Událost nastane při přiblížení na danou oblast na mapě.



Obrázek 6.1: Diagram tříd

6.3 Zobrazení mapových podkladů

Zobrazení mapových podkladů v aplikaci je navrženo tak, že při stisku tlačítka nahoru se provede metoda `RenderUp()`, při stisku tlačítka dolů metoda `RenderDown()`, při stisku doprava `RenderRight()` a při stisku doleva se provede metoda `RenderLeft()`. O nastavení požadovaných mapových podkladů se starají metody `SetRight()`, `SetLeft()`, `SetUp()` a `SetDown()`, které v sobě implementují potřebnou funkcionalitu (zjištění, jaké mapové podklady se mají přidat do kolekce, nastavení správných parametrů mapových podkladů). Dále bude uveden pouze příklad metod `RenderRight()` a `SetRight()`. Ostatní metody jsou v podstatě stejné, liší se pouze v hodnotách hranic, podle kterých se nastavují požadované mapové podklady.

6.3.1 Metoda `RenderRight()`

```
public void RenderRight()
{
    //globální kolekce mapových podkladů se přepíše novou kolekcí
    //vrácenou metodou SetRight()
    Collection = SetRight();
    //nastaví se pozice mapových podkladů
    SetPositionX(-Shift);
    //vykreslí se mapové podklady z globální kolekce
    RenderPictures();
}
```

Metoda `RenderRight()` se volá při stisku tlačítka doprava. Obsahuje v sobě metodu `SetRight()`, která nastaví kolekci mapových podkladů. Poté metoda `SetPosition(-Shift)` nastaví pozici mapových podkladů o hodnotu `-Shift`. Poslední metoda `RenderPictures()` vykreslí mapové podklady na displej PDA.

6.3.2 Metoda `SetRight()`

```
private IEnumerable<MapPicture> SetRight()
{
    //kolekce mapových podkladů, kterou metoda vrátí
    List<MapPicture> result = new List<MapPicture>();

    //cyklus postupně čte mapové podklady z globální kolekce
    foreach (var o in Collection)
    {
        //zjistí, zdali je pozice X podkladu z cyklu na hranici
        if (o.PositionX == -(PictureSize.X - DisplaySize.X))
        {
            //vytvoří se číslo nového mapového podkladu
            int newX = o.Image.X + 1;
            int newY = o.Image.Y;

            //vytvoří se nový mapový podklad, nastaví se mu správné
            //parametry a přidá se do kolekce
            result.Add(new MapPicture(o.PositionX + PictureSize.X,
            o.PositionY, new Point(newX, newY)));

            //do kolekce se také přidá mapový podklad z cyklu
            result.Add(o);
        }
        //pokud je na druhé hranici, nic se neprovede a podklad bude
        //ignorován
        else if (o.PositionX == -PictureSize.X + Shift)
    }
```

```

        {
            //nic se neprovede
        }
        else
            //nic se neprovede, pouze se do kolekce přidá mapový
            //podklad z cyklu
            result.Add(o);
    }
    //vrací novou kolekci určenou ke zobrazení
    return result;
}

```

V první fázi se vytvoří lokální kolekce, která bude obsahovat nové mapové podklady. Dále se provede cyklus, který postupně projde mapové podklady z globální kolekce mapových podkladů. U těchto mapových podkladů bude testovat, zdali se nějaký z nich nachází na hranici. Pokud ano, tak se buď vytvoří nový mapový podklad, nastaví se mu pozice a spolu s podkladem z cyklu se přidají do lokální kolekce nebo bude podklad ignorován. Pokud není na hranici, tak se pouze do lokální kolekce uloží podklad z cyklu. Nakonec metoda vrátí lokální kolekci.

6.3.3 Metoda SetPositionX(int shift)

```

private void SetPositionX(int shift)
{
    //cyklus postupně prohází kolekci mapových podkladů
    foreach (MapPicture obr in Collection)
    {
        //nastaví se pozice X mapového podkladu z cyklu
        obr.PositionX = obr.PositionX + shift;
    }
}

```

Metoda `SetPositionX(int shift)` nastaví pozici X mapového podkladu z kolekce na hodnotu, která je buď větší (při kladné hodnotě `shift`) a nebo menší (při záporné hodnotě `shift`) než současná hodnota pozice X. Aplikace obsahuje také metodu `SetPositionY(int shift)`, která dělá totéž ve směru osy Y.

6.3.4 Metoda RenderPictures()

```

public void RenderPictures()
{
    //cyklus postupně prohází kolekci mapových podkladů
    foreach (MapPicture obr in Collection)
    {
        try
        {
            //vykresluje mapový podklad z cyklu
            Rendering.CreateGraphics().DrawImage(obr.Image,
            obr.PositionX, obr.PositionY);
        }
        catch (ArgumentNullException) { }
    }
}

```

Metoda `RenderPictures()` prochází kolekci mapových podkladů a postupně každý podklad vykreslí.

6.4 Výpočet GPS souřadnice na mapě

```
public Gps GetGps(int X, int Y)
{
    //GPS souřadnice určená k vrácení
    Gps result = null;
    //zeměpisná šířka
    double latitude = 0;
    //zeměpisná délka
    double longitude = 0;

    //cyklus prochází podklady z globální kolekce
    foreach (var obr in Collection)
    {
        //výpočet skutečné pozice mapového podkladu
        int x = X - obr.PositionX;
        int y = Y - obr.PositionY;

        //zjištění, zda se souřadnice nachází na mapovém podkladu
        //z cyklu
        if (x <= PictureSize.X && y <= PictureSize.Y && x > 0 && y >
0)
        {
            //výpočet zeměpisné šířky
            latitude = (((obr.GpsRightBottom.Latitude -
obr.GpsLeftTop.Latitude) / PictureSize.Y) * y) +
obr.GpsLeftTop.Latitude;

            //výpočet zeměpisné délky
            longitude = (((obr.GpsRightBottom.Longitude -
obr.GpsLeftTop.Longitude) / PictureSize.X) * x) +
obr.GpsLeftTop.Longitude;

            //vytvoření GPS souřadnice
            result = new Gps(longitude, latitude, obr.Img);
        }
    }
    //vrácení GPS souřadnice
    return result;
}
```

Metoda ze zadaných parametrů vrátí GPS souřadnici. Zadané parametry jsou souřadnice X a Y na displeji PDA. V prvním kroku cyklus prochází mapové podklady z kolekce a zjišťuje, jakému mapovému podkladu patří parametry metody. Když tento podklad nalezne, tak vypočte GPS souřadnici podle návrhové části a vrací výsledek.

6.5 Zpřesnění GPS souřadnic na mapě

Zpřesnění GPS souřadnic v aplikaci je implementováno tak, že na základě naměřených GPS souřadnic porovnaných se skutečnými souřadnicemi byly určeny hodnoty ϵ_{lat} a ϵ_{lon} , které se přičtou k výsledkům a tím se výsledky zpřesní. Hodnoty byly zvoleny na základě naměřených GPS souřadnic v aplikaci MapyPDA a na serveru mapy.cz[8]. Z velkého množství naměřených hodnot, které byly zprůměrovány vyšly nejpřesnější výsledky pro $\epsilon_{lat} = 0.6''$ a $\epsilon_{lon} = 0.05''$. Naměřené hodnoty jsou k nahlédnutí v příloze C.

V aplikaci jsou hodnoty ϵ_{lat} a ϵ_{lon} implementovány do metod LongitudeToDegree(double longitude, double difference) a LatitudeToDegree(double latitude, double difference).

Příklad implementace jedné z nich.

```
public static string LatitudeToDegree(double latitude, double difference)
{
    int degrees = (int)latitude;
    double betweenDegMin = (latitude - degrees) * 60;
    int minutes = (int)betweenDegMin;
    double seconds = (betweenDegMin - minutes) * 60;

    return degrees + "° " + minutes + "' " + (seconds +
difference).ToString() + "'N"; //přičtení hodnoty k vteřinám GPS
}
```

6.6 Výpočet vzdálenosti dvou GPS souřadnic

```
public double DistanceVector()
{
    return Math.Acos(
        Math.Cos(DegreeToRadian(Lat1)) *
        Math.Cos(DegreeToRadian(Lon1)) *
        Math.Cos(DegreeToRadian(Lat2)) *
        Math.Cos(DegreeToRadian(Lon2))
        + Math.Cos(DegreeToRadian(Lat1)) *
        Math.Sin(DegreeToRadian(Lon1)) *
        Math.Cos(DegreeToRadian(Lat2)) *
        Math.Sin(DegreeToRadian(Lon2))
        + Math.Sin(DegreeToRadian(Lat1)) *
        Math.Sin(DegreeToRadian(Lat2))
    ) * 6372795;
}
```

Metoda vrací vzdálenost dvou GPS souřadnic v metrech. Konstanta 6372795 je poloměr Země v metrech. Tuto implementaci jsem použil z jazyka PHP a předělal do jazyka C# [20].

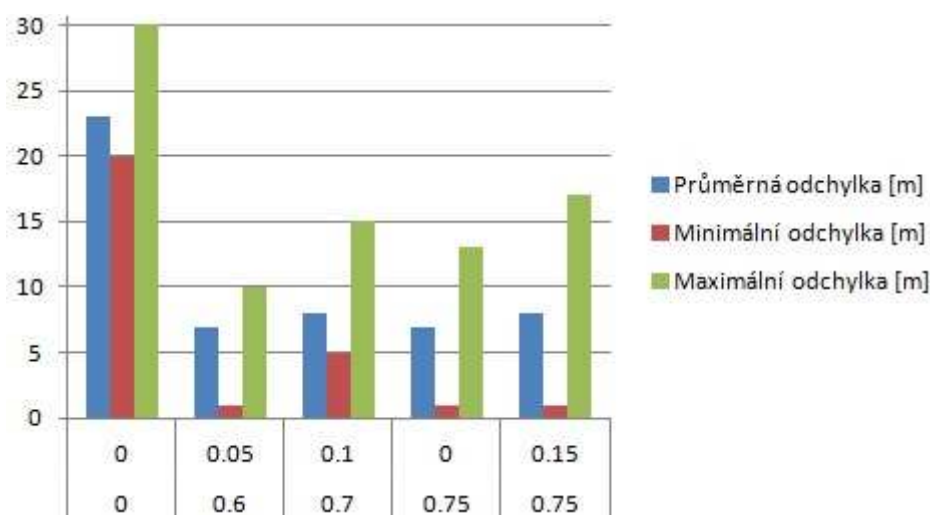
7 Testy přesnosti

Testy přesnosti jsem prováděl měření GPS souřadnice daného bodu v aplikaci MapyPDA a porovnáním[21] s naměřenou hodnotou na serveru mapy.cz[8]. Přesnost jsem testoval pro různé hodnoty ϵ_{lat} a ϵ_{lon} . Naměřené hodnoty jsou k vidění v příloze C a porovnání výsledků v tabulce 7.1 a na obrázku 7.1. Dále jsem vytvořil testovací aplikaci, pomocí níž jsem testoval zobrazení GPS souřadnice v souřadnicovém systému, který jsem navrhl pro aplikaci MapyPDA. V tomto testu jsou výsledky ekvivalentní s výsledky v tabulce 7.1.

Tabulka 7.1: Porovnání naměřených hodnot

ϵ_{lat} [vteřiny]	ϵ_{lon} [vteřiny]	Průměrná odchylka [m]	Minimální odchylka [m]	Maximální odchylka [m]
0	0	23	20	30
0.6	0.05	7	1	10
0.7	0.1	8	5	15
0.75	0	7	1	13
0.75	0.15	8	1	17

Z tabulky a obrázku je patrné, že nejvhodnější nastavení hodnot ϵ_{lat} a ϵ_{lon} je 0.6 respektive 0.05. Nejméně vhodné je podle očekávání nastavení obou hodnot na 0.



Obrázek 7.1: Porovnání naměřených hodnot

8 Závěr

Kapitola obsahuje zhodnocení práce a návrh do budoucna.

8.1 Zhodnocení práce

Zde jsou uvedeny cíle, kterých mělo být dosaženo v této bakalářské práci:

- Aplikace do PDA na platformě Windows Mobile 6
- Implementace a optimalizace map
- Algoritmus zobrazování mapových podkladů
- Návrh souřadnicového systému
- Zpřesnění GPS souřadnic v aplikaci
- DLL knihovna
- Dokumentace

Všech těchto cílů bylo dosaženo, dále uvedu jejich krátké zhodnocení.

8.1.1 Zhodnocení aplikace

Povedlo se vytvořit funkční aplikaci MapyPDA, pomocí které je možné zobrazovat mapové podklady, získávat GPS souřadnice jednotlivých bodů s přesností do 10 metrů a měřit vzdálenost mezi dvěma body na mapě. Do aplikace mohou být přidány nové mapové podklady, stačí pouze ve zdrojovém kódu nastavit potřebné parametry a projekt zkompileovat.

8.1.2 Implementace a optimalizace map

Mapové podklady v PDA jsou uloženy ve složce a jsou pojmenovány dle formátu, kterému rozumí algoritmus určený k jejich načtení do paměti a následnému zobrazení na displej PDA. GPS souřadnice k těmto mapovým podkladům jsou uloženy v textovém souboru. Co se týká optimalizace paměti, mapový podklad má rozměry 640 x 640 pixelů a maximální počet načtených mapových podkladů v paměti jsou čtyři. Při těchto rozměrech a počtu mapových podkladů není s pamětí žádný problém, aplikace byla testovaná při 16ti hodinové zátěži.

8.1.3 Algoritmus zobrazování mapových podkladů

Byl vytvořen algoritmus určený ke zobrazování mapových podkladů. Tento algoritmus je velice obecný a může být součástí i jiných projektů.

8.1.4 Souřadnicový systém

Byl navržen souřadnicový systém pro potřeby aplikace. Tento souřadnicový systém je založený na souřadnicovém systému z .NET Frameworku.

8.1.5 Zpřesnění GPS souřadnic na mapě

GPS souřadnice na mapě v aplikaci se mi podařilo zpřesnit. Docílil jsem toho získáním přesných GPS souřadnic jednotlivých rohů mapových podkladů a dále vytvořením vzorce, na jehož základě jsem implementoval algoritmus, který přesnost GPS souřadnic v aplikaci zvyšuje.

8.1.6 DLL knihovna

Součástí bakalářské práce je také DLL knihovna aplikace, která může být připojena k jiným vyvíjeným programům. Knihovna obsahuje základní rozhraní, pomocí kterého je s ním možno pracovat. Není problém dodělat další rozhraní podle potřeb vývojářů.

8.1.7 Dokumentace

Součástí bakalářské práce je také dokumentace vytvořeného díla. Tuto dokumentaci je možné bezproblémů otevřít na operačních systémech MS Windows.

8.1.8 Získání mapových podkladů

Součástí bakalářské práce je také popis toho, jak získat mapové podklady pomocí softwaru MapGrabber[9] ze serveru mapy.cz[8].

8.2 Návrh pro budoucí řešení

Aplikace MapyPDA tvoří pevný základ pro další rozvoj. Zde je seznam funkcí a úkolů, které by měly být do aplikace dodělány.

- Zpřesnění GPS souřadnic pomocí systému UTM
- Zpřesnění GPS souřadnic pomocí ϵ_{lat} a ϵ_{lon} pro různé oblasti
- Ukládání GPS souřadnic a zobrazení polohy v aplikaci
- Navigace
- Implementace leteckých map
- Platforma Windows Phone 7.x

8.2.1 Zpřesnění GPS souřadnic pomocí systému UTM

Určitě by stálo za pokus vytvořit převodník mezi systémem UTM a GPS. GPS souřadnice v aplikaci pak počítat pomocí souřadnic UTM a výsledek převádět na souřadnice GPS. Implementovat tuto funkcionalitu do aplikace není problém. Jediný problém je vytvořit převodník.

8.2.2 Zpřesnění GPS souřadnic pomocí ϵ_{lat} a ϵ_{lon} pro různé oblasti

V aplikaci je zpřesnění provedeno pomocí hodnot ϵ_{lat} a ϵ_{lon} , které jsou připočteny k GPS poloze. Zajímavý nápad, jak zpřesnit GPS souřadnice v aplikaci, je mapu rozdělit na oblasti s podobnou nepřesností a zvolit pro každou tuto oblast jiné hodnoty ϵ_{lat} a ϵ_{lon} . Toto by neměl být problém, stačí pouze správně určit oblasti.

8.2.3 Ukládání GPS souřadnic

Zvolit typ úložiště a ukládat jednotlivé body s jejich popisem je další zajímavá funkcionality, která by jistě měla být do aplikace přidána. Uživatel by si mohl po spuštění aplikace vybrat místo na mapě, které ho zajímá, nebo si uložit místo nové.

8.2.4 Navigace

Uživatel by si zvolil ze seznamu místo k navigaci, popřípadě by zadal souřadnici a v aplikaci by při pohybu krajinou viděl šipku, která by určovala směr pohybu.

8.2.5 Implementace leteckých map

Implementace leteckých map zvýší uživatelský prožitek. Uživatel by měl možnost zvolit si typ zobrazení výběrem leteckého pohledu nebo standardní mapy.

8.2.6 Platforma Windows Phone 7.x

Platforma Windows Mobile je v poslední době nahrazována novou platformou Windows Phone 7.x. Aplikace by tedy jistě měla být do budoucna předělána na tuto platformu. Ta nabízí zcela nové uživatelské prostředí a ovládání.

Přehled literatury

- [1] KREJCAR, Ondřej. *Platformy operačních systémů a vývoj aplikací na mobilních zařízeních kategorie PDA a Smartphone* [online]. VŠB - TU Ostrava, 2008. 34 s. Dostupné z WWW: <<http://klokan.vsb.cz/vyuka/mgit/V%C3%BDvoj%20aplikac%C3%AD%20na%20PDA%20za%C5%99%C3%ADzen%C3%ADch.docx>>.
- [2] BŘEHOVSKÝ, Martin. *Úvod do geografických informačních systémů* [online]. Plzeň: ZČU. 116 s. Přednáškové texty. Dostupné z WWW: <<http://gis.zcu.cz/studium/ugi/e-skripta/ugi.pdf>>.
- [3] BUREŠ, Pavel. *Začínáme s PDA: co je dobré vědět. Svět hardware* [online]. 11.1.2008 [cit. 2011-10-05]. Dostupný z WWW: <http://www.svethardware.cz/art_doc-C8BD7D7A299365F4C125739E0078E869.html>.
- [4] KRTIČKA, Luděk. *Úvod do kartografie*. 1. vyd. [s.l.] : [s.n.], 2007. 87 s. Dostupné z WWW: <http://www1.osu.cz/~krticka/Krticka_DiV_Kartografie.pdf>. ISBN 978-80-7368-344-3.
- [5] HRDINA, Zdeněk. *Rádiové určování polohy : Družicový systém GPS*. [s.l.] : [s.n.], 1995. 267 s. ISBN 80-01-01386-3.
- [6] RAPANT, P.: *Družicové polohové systémy*. VŠB-TU Ostrava, 2002. 200 str. ISBN 80-248-0124-8. dostupné z WWW: <http://gis.vsb.cz/dokumenty/dns-gps/at_download/file>.
- [7] *První PDA* [online].[cit. 2011-10-05]. První PDA. Dostupné z WWW: <<http://prvnipda.elfineer.cz/>>.
- [8] *Mapy.cz* [online]. 2010 [cit. 2010-11-09]. Mapy.cz. Dostupné z WWW: <<http://mapy.cz/>>.
- [9] PANSHIN, Yuri . *The Code Project* [online]. 2008 [cit. 2011-10-05]. Map Grabber on C#. Dostupné z WWW: <<http://www.codeproject.com/KB/cs/MapGrabber.aspx>>.
- [10] *Microsoft MSDN* [online]. 2010 [cit. 2011-10-05]. MSDN Library. Dostupné z WWW: <<http://msdn.microsoft.com/cs-cz/library/ms123401.aspx>>.

- [11] *Military Map Reading 201* [online]. 2010 [cit. 2011-10-05]. Military Map Reading. Dostupné z WWW: <<http://earth-info.nga.mil/GandG/coordsys/mmr201.pdf>>.
- [12] *UTM*. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-10-05]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/UTM>>.
- [13] *Ortodroma*. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-10-05]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Ortodroma>>.
- [14] *Windows Mobile*. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-10-24]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Windows_Mobile>.
- [15] PROSISE, Jeff. *Programování v Microsoft .NET*. [s.l.] : Computer Press, 2003. 736 s.
- [16] BĚHÁLEK, Marek. *Programovací jazyk c#*. 133 s. Dostupný z WWW: <<http://www.scribd.com/doc/54806242/C-Behalek>>.
- [17] Kuchta, Michal: *Podpora Geocachingu pro kapesní počítače s GPS*, bakalářská práce, Brno, FIT VUT v Brně, 2009
- [18] *Zeměpisná šířka*. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-10-24]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Zem%C4%9Bpisn%C3%A1_%C5%A1%C3%AD%C5%99ka>.
- [19] *Zeměpisná délka*. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2011-10-24]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Zem%C4%9Bpisn%C3%A1_d%C3%A9lka>.
- [20] VRÁNA, Jakub. *PHP triky* [online]. 2007 [cit. 2011-11-03]. Vzdálenost dvou zeměpisných bodů. Dostupné z WWW: <<http://php.vrana.cz/vzdalenost-dvou-zemepisnych-bodu.php>>.
- [21] SCHNEIDER, Adam. GPS Visualizer's coordinate calculators & distance tools [online]. 2011 [cit. 2011-11-09]. GPS Visualizer. Dostupné z WWW: <<http://www.gpsvisualizer.com/calculators>>.

Seznam příloh

Příloha A – Obsah přiloženého CD

Příloha B – Uživatelská příručka

Příloha C – Naměřené hodnoty

A Obsah přiloženého CD

- Adresář MapyPDA s programem
- Adresář s aplikací určenou k testování GPS polohy ze zadaných GPS souřadnic
- Zdrojové kódy k programu MapyPDA
- Dokumentace k programu MapyPDA
- Knihovna DLL
- Text bakalářské práce Mapy do PDA
- README.txt

B Uživatelská příručka

Tato příloha obsahuje návod na obsluhu aplikace Mapy PDA.

B.1 Požadavky před instalací

- PDA s operačním systémem Windows Mobile 6
- 35 MB paměti
- Možnost přenosu dat mezi PDA a PC

B.2 Instalace

Adresář s programem MapyPDA v příloženém CD nakopírujte do kořenového adresáře na kartě Storage Card ve Vašem PDA. Poté spusťte soubor *MapyPDA.exe*.

B.3 Odinstalace

Smažte adresář vytvořený při instalaci.

B.4 Použití programu

Po spuštění programu se objeví úvodní obrazovka, kde si zvolíte dotknutím stylusu místo na mapě, které chcete přiblížit. Dále se spustí hlavní okno aplikace, kde můžete posouvat po mapě stisknutím navigačního tlačítka PDA. V případě, že chcete zjistit GPS souřadnici nějakého místa, stačí se dotknout stylusem obrazovky a na displeji se zobrazí GPS souřadnice. V případě, pokud chcete změřit vzdálenost dvou míst na mapě, stačí v navigační liště vybrat možnost *Měření vzdálenosti* a poté označit místa na mapě dotykem stylusu. Zvolením možnosti *Konec měření* v navigační liště se přepnete zpět do režimu určování GPS souřadnic. Zvolením možnosti *Zpět* v navigační liště se přesunete zpět do hlavního okna, kde můžete aplikaci buď ukončit vybráním položky *Konec* v navigační liště nebo se přiblížit na další oblast na mapě dotykem stylusu.

C Naměřené GPS souřadnice

Tato příloha obsahuje testované GPS souřadnice.

C.1 Porovnání GPS souřadnic

Tabulky obsahují GPS souřadnice v aplikaci MapyPDA, GPS souřadnice na serveru mapy.cz[8] a jejich odchylku v metrech[21].

Tabulka C.1: ϵ_{lat} a $\epsilon_{lon} = 0$ vteřin

GPS souřadnice PDA	GPS souřadnice mapy.cz	Odchylka [m]
49°34'18.9145"N 14°32'8.8201"E	49°34'19.752"N 14°32'9.109"E	27
49°32'1.8153"N 14°47'50.3156"E	49°32'2.464"N 14°47'50.421"E	20
49°24'55.5021"N 14°40'22.3232"E	49°24'56.397"N 14°40'22.325"E	28
49°24'1.1396"N 14°29'24.2708"E	49°24'1.913"N 14°29'24.376"E	24
49°17'59.4464"N 14°27'58.4099"E	49°18'0.056"N 14°27'58.511"E	19
49°15'46.8795"N 14°42'21.0479"E	49°15'47.786"N 14°42'20.997"E	28
49°11'5.3612"N 14°41'53.2628"E	49°11'6.338"N 14°41'53.173"E	30
49°11'28.9736"N 14°51'11.6359"E	49°11'29.395"N 14°51'11.549"E	13
Průměr		23

Tabulka C.2: $\epsilon_{lat} = +0.75$ vteřin a $\epsilon_{lon} = 0$ vteřin

GPS souřadnice PDA	GPS souřadnice mapy.cz	Odchylka [m]
49°34'19.535"N 14°32'8.820"E	49°34'19.769"N 14°32'9.109"E	9
49°32'4.2492"N 14°48'26.7422"E	49°32'4.178"N 14°48'26.759"E	2
49°24'44.9182"N 14°40'19.1465"E	49°24'45.120"N 14°40'19.273"E	7
49°23'44.3393"N 14°29'1.1454"E	49°23'44.244"N 14°29'1.514"E	8
49°17'42.7109"N 14°28'3.4611"E	49°17'42.496"N 14°28'3.652"E	8
49°15'36.9433"N	49°15'36.956"N	8

14°42'21.246"E	14°42'21.655"E	
49°11'4.7574"N 14°41'40.5156"E	49°11'4.912"N 14°41'40.313"E	6
49°11'7.7185"N 14°51'18.5828"E	49°11'7.367"N 14°51'18.407"E	11
Průměr		7

Tabulka C.3: $\epsilon_{lat} = +0.75$ vteřin a $\epsilon_{lon} = 0.15$ vteřin

GPS souřadnice PDA	GPS souřadnice mapy.cz	Odchylka [m]
49°34'22.8379"N 14°31'58.2138"E	49°34'22.866"N 14°31'58.274"E	1
49°32'3.0231"N 14°48'35.1528"E	49°32'2.783"N 14°48'35.123"E	7
49°24'44.724"N 14°40'23.1681"E	49°24'44.952"N 14°40'23.071"E	7
49°23'57.2916"N 14°29'10.4264"E	49°23'57.303"N 14°29'10.631"E	4
49°17'38.889"N 14°28'4.2053"E	49°17'38.711"N 14°28'4.311"E	6
49°15'49.8963"N 14°42'55.9384"E	49°15'50.156"N 14°42'55.719"E	9
49°11'25.8491"N 14°42'10.4511"E	49°11'26.185"N 14°42'10.271"E	11
49°10'53.6648"N 14°51'17.8435"E	49°10'53.182"N 14°51'17.560"E	16
Průměr		8

Tabulka C.4: $\epsilon_{lat} = +0.7$ vteřin a $\epsilon_{lon} = 0.1$ vteřin

GPS souřadnice PDA	GPS souřadnice mapy.cz	Odchylka [m]
49°34'28.828"N 14°32'7.330"E	49°34'28.828"N 14°32'7.330"E	7
49°31'59.4712"N 14°47'49.0223"E	49°31'59.353"N 14°47'48.966"E	4
49°24'49.8528"N 14°39'31.5951"E	49°24'50.131"N 14°39'31.643"E	9
49°23'48.6283"N 14°29'24.2716"E	49°23'48.621"N 14°29'24.514"E	5
49°17'33.0762"N 14°28'18.5168"E	49°17'32.964"N 14°28'18.731"E	6
49°15'59.9498"N 14°42'31.8853"E	49°16'0.152"N 14°42'31.934"E	6
49°11'12.1985"N 14°41'55.4765"E	49°11'12.530"N 14°41'55.557"E	10

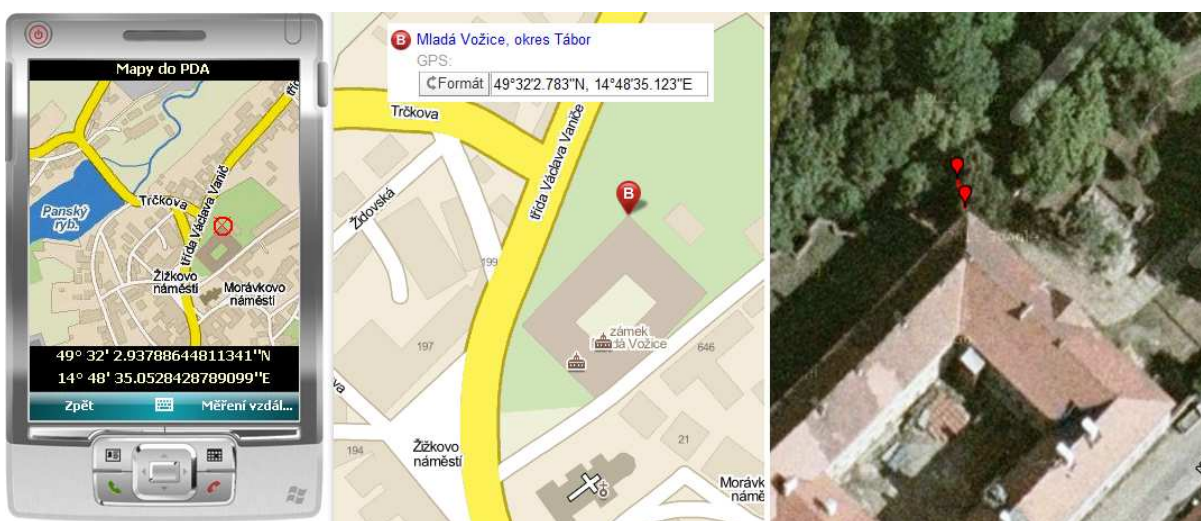
49°10'60.4482"N 14°51'4.0582"E	49°11'0.014"N 14°51'3.806"E	14
Průměr		8

Tabulka C.5: $\epsilon_{lat} = +0.6$ vteřin a $\epsilon_{lon} = 0.05$ vteřin

GPS souřadnice PDA	GPS souřadnice mapy.cz	Odchylka [m]
49°34'29.2936"N 14°31'55.5243"E	49°34'29.558"N 14°31'55.596"E	8
49°32'2.9378"N 14°48'35.0528"E	49°32'2.783"N 14°48'35.123"E	5
49°24'45.220"N 14°40'37.660"E	49°24'45.187"N 14°40'37.636"E	1
49°23'43.2826"N 14°29'0.1036"E	49°23'43.366"N 14°29'0.556"E	9
49°17'43.4028"N 14°28'7.4728"E	49°17'43.421"N 14°28'7.652"E	4
49°15'18.2056"N 14°43'9.3981"E	49°15'18.536"N 14°43'9.431"E	10
49°10'43.3343"N 14°41'44.5182"E	49°10'43.583"N 14°41'44.593"E	8
49°11'9.1801"N 14°51'18.7316"E	49°11'8.986"N 14°51'18.502"E	8
Průměr		7

C.2 Vizualizace naměřených hodnot

V první oblasti je vidět GPS poloha v aplikaci MapyPDA, v druhé oblasti stejná GPS poloha na serveru mapy.cz[8] a ve třetí oblasti jejich rozdíl[21].



Obrázek C.1: Porovnání polohy