

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VIZUALIZACE VÝŠKOVÉHO PROFILU A INFORMACÍ O PROJETÉ TRASE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

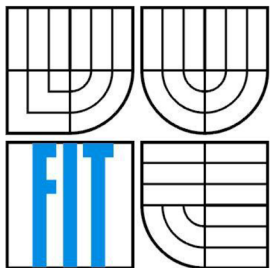
AUTHOR

MICHAL KOLČÁREK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VIZUALIZACE VÝŠKOVÉHO PROFILU A INFORMACÍ O PROJETÉ TRASE

VIZUALIZATION OF ROUTE INFORMATION AND ELEVATION PROFILE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Michal Kolčárek

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Radek Kubíček

BRNO 2011

Abstrakt

Cílem této práce bylo vytvořit komplexní program pro správu geografických dat získaných pomocí GPS přijímače s grafickým uživatelským rozhraním. Výsledná aplikace umožňuje zobrazení údajů o trasách, jejich zobrazení v mapě a vykreslování grafů. Důraz byl kladen hlavně na uživatelskou přívětivost a dobrou použitelnost i pro začínající uživatele. Byl použit programovací jazyk Java za dodržení objektově orientovaného paradigmatu.

Abstract

An ambition of this thesis was to create a complex program for managing geographic data from GPS receiver with graphic user interface. The final application is able to show an information about tracks, to draw tracks into map and to create graphs. Main challenge was to create a user friendly application, which is easy to manage even by beginner users. In this application was used a Java language using objected-oriented paradigm.

Klíčová slova

GPS, trasa, výškový profil, souřadnice, geografická data, Java, HTML

Keywords

GPS, route, elevation profile, coordinates, geographic data, Java, HTML

Citace

Michal Kolčárek: Vizualizace výškového profilu a informací o projeté trase, bakalářská práce, Brno, FIT VUT v Brně, 2011

Vizualizace výškového profilu a informací o projeté trase

Prohlášení

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

.....
Michal Kolčárek
12. 05. 2011

Poděkování

Rád bych poděkoval Ing. Radku Kubičkovi za vedení bakalářské práce, za odbornou pomoc a cenné rady při její tvorbě. Také bych rád poděkoval své rodině, přítelkyni a kamarádům za podporu a pomoc při studiu.

© Michal Kolčárek, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod.....	3
2 GPS a systémy pro určení polohy	4
2.1 Historie a současný stav.....	4
2.2 Fyzická realizace GPS	5
2.3 Určování polohy pomocí GPS	5
2.4 Přesnost určení polohy a omezení funkčnosti GPS	6
2.5 Určení nadmořské výšky	6
2.6 Souřadnicový systém pro GPS	7
2.7 Výpočet vzdálenosti z GPS souřadnic	7
3 Formáty souborů pro uložení dat z GPS	9
3.1 Formát GPX.....	9
3.2 Formát KML.....	10
4 Technologie použité při práci	11
4.1 Programovací jazyk Java	11
4.2 Sestavovací systém Apache Ant.....	11
4.3 Knihovna JXMapKit.....	12
4.4 Technologie XML	13
4.5 Technologie HTML a CSS	13
5 Návrh aplikace	14
5.1 Návrh grafického uživatelského rozhraní	14
5.2 Ovládání mapy	16
5.3 Validace bodů	17
5.4 Možnost práce s vybranými body na trase.....	18
5.5 Zobrazení informací o trase	18
5.6 Zpracování grafu.....	18
5.7 Uložení dat.....	19
5.7.1 Databáze tras.....	19
5.7.2 Uložení jednotlivých tras	20
5.8 Tvorba webových stránek s informacemi o trasách.....	21
6 Architektura aplikace	22
6.1 Balíček <code>track</code>	22
6.1.1 Třída <code>Point4D</code>	23
6.1.2 Třída <code>TrackInfo</code>	23

6.1.3	Třída GpsStorage.....	23
6.2	Balíček files.....	23
6.3	Balíček util.....	24
6.4	Balíček windows.....	25
6.4.1	Třída AllTrakcsDB.....	25
6.4.2	Třída Graph.....	26
6.4.3	Třída GraphCreator.....	26
6.4.4	Třída LineDefinition.....	26
6.4.5	Třída Main.....	27
6.4.6	Třída PanelPointTable.....	27
6.5	Implementace aplikace.....	28
6.5.1	Vlákna.....	28
6.5.2	Automatická validace bodů.....	29
6.5.3	Vykreslení trasy do mapy.....	29
7	Závěr.....	30
8	Přílohy.....	32
8.1	Obsah přiloženého CD.....	32
8.2	Příručka pro používání programu.....	32
8.2.1	Ovládání grafu.....	32

1 Úvod

Úkolem této práce bylo vytvořit komplexní, ale zároveň přehledný program na správu geografických dat. Takovýchto produktů existuje několik, ale bývají dosti složité, nepřehledné a mají proto omezené využití hlavně u začátečníků, kteří se ve většině případů orientují buď v problematice GPS anebo ve výpočetní technice. Také jde často o firemní záležitosti, takovýto program lze pak použít pouze s GPS přijímačem dané značky, nebo využívá mapové podklady, které nejsou volně šiřitelné, a tedy uživatel může narazit na problém s autorskými právy. Proto byl ve vytvořené aplikaci kladen důraz na volně šiřitelné zdroje a mapové podklady. Rovněž se aplikace nesoustředí na jeden konkrétní formát pro uložení GPS dat, ale používá jich vícero. Nejdůležitější formát pro uložení dat z GPS navíc není proprietární a nepodléhá tedy žádné konkrétní společnosti.

GPS je technologií, která je dnes velmi populární a rozšířená. Prozatím jde o jediný globálně použitelný produkt tohoto typu na světě, proto se mnohem více, než pro původní vojenské účely, používá pro navigaci vozidel na cestách a osob v terénu. V kombinaci s jinou technologií lze tyto objekty rovněž sledovat a vyhledávat. Vždyť velké množství mobilních telefonů současnosti technologií GPS disponuje. Proto existuje také mnoho zajímavých aplikací pro různé mobilní platformy, využívající funkce GPS. Nemluvě o osobních automobilech, které jsou často už od výroby vybaveny obrazovkou pro zobrazení pozice vozu a směru do cíle cesty.

Výsledkem této práce nemá být aplikace pro navigaci do cíle nebo pro zobrazení aktuální pozice v terénu, ale program pro zpětnou analýzu absolvované trasy. To je vhodné pro zpětný přehled o navštívených místech. Dalším nezanedbatelným využitím je analýza informací z hlediska srovnávání výkonů. Když například podnikneme dvakrát stejnou cestu, pomůže nám tato aplikace srovnat naše rychlosti během obou cest. Můžeme tak vidět, zdali naše výkonnost klesá či stoupá.

V této práci bude zmíněna nejdříve teorie technologie globální navigace pomocí družic, hlavně pak technologie GPS. Bude krátce popsána historie systémů pro určování polohy, popis toho, jak celý systém funguje, jak se na základě dat ze satelitů určuje poloha a výška a jaká jsou omezení.

V další kapitole bude popsáno několik formátů pro uložení geografických dat.

Další významnou kapitolou budou technologie použité při vývoji aplikace. Zde bude zmíněn programovací jazyk Java, utilita Apache Ant, technologie XML, HTML a CSS. Řeč bude také o knihovně jXMapKit.

V kapitole číslo pět bude rozebrán návrh aplikace, tedy konkrétně postup při tvorbě GUI, ovládání aplikace a její významné součásti a jejich zpracování. Nakonec bude zmínka o uložení dat a exportu informací z aplikace.

Šestá kapitola se věnuje architektuře aplikace. Zde budou rozebrány jednotlivé balíčky, které aplikace obsahuje, diagramy tříd a jednotlivé metody.

V sedmé kapitole budou rozebrány podrobněji některé implementační detaily a zajímavá řešení.

Závěrečná kapitola zhodnocuje výsledek práce, důležité aspekty tvorby aplikace a z nich plynoucí poučení a navrhuje budoucí možnosti pro úpravu aplikace a její rozšíření.

V příloze pak můžeme najít obsah příloženého CD a také stručný popis ovládání aplikace.

2 GPS a systémy pro určení polohy

Zkratka GPS znamená Global Positioning System, tedy celosvětový systém pro určení polohy [5]. Tato poloha se zjišťuje na základě systému družic ve vesmíru, komunikujících s GPS přijímačem. GPS je momentálně celosvětově nejrozšířenější systém tohoto druhu, není však jediný. Původně technologie pro určování polohy na základě signálu z vesmíru sloužily výhradně pro vojenské účely a jejich původ pochází z dob studené války. Hlavní mocnosti, které stály za vznikem těchto systémů byly samozřejmě USA a Rusko.

2.1 Historie a současný stav

GPS je projektem Amerického Ministerstva Obrany (Department of Defence), který je označován jako NAVSTAR GPS. Tento systém je v provozu od roku 1994, v částečném provozu je pak od roku 1993. GPS je nástupcem systému Tranzit, který byl v provozu od 60. let 20. století. Tranzit měl ale oproti GPS mnohem horší přesnost až v řádech stovek metrů. V roce 1996 byl projekt Tranzit definitivně ukončen a plně nahrazen systémem GPS. 1. května 2000 byla funkce GPS zpřístupněna také veřejnosti a postupem času se tento systém stal nejrozšířenějším v civilním sektoru na celém světě [2].

GPS se ovšem nejspíše nikdy neodpoutá od své vojenské minulosti. Přesnost určení polohy u civilních přijímačů je nižší, než u těch pro vojenské účely. Navíc USA stále disponuje možností systém pro civilní sektor jednoduše vypnout tak, že jejich vojenské přijímače zůstanou v provozu. Zároveň také může ovlivnit přesnost uživatelského určení polohy. Na druhou stranu vláda USA financuje celosvětový provoz GPS systému částkami v řádech stovek milionů USD ročně a umožňuje každému obyvateli kdekoli na světě zdarma používat tuto technologii, jejíž vývoj a vybudování si vyžádaly obrovské finanční prostředky.

Systémem podobným GPS disponuje také Rusko. Jmenuje se GLONASS a jeho vývoj byl zahájen v roce 1970. Koncem 90. let byl tento systém v úpadku, ale nyní se usiluje o obnovení jeho plné funkčnosti. Tento systém ale není v soukromém sektoru příliš využíván, hlavně proto, že momentálně nefunguje naplno a také proto, že přijímače signálu pro tento systém nejsou tak dostupné a nejsou kompatibilní s přijímači signálu GPS [2].

O vývoj vlastního systému usiluje také Evropská Unie. Jmenuje se Galileo a jde o první produkt tohoto typu, který nevzniká pro vojenské účely. Tento systém by měly být schopny využívat i přijímače signálu GPS. Systém by měl mít 5 úrovní přesnosti. Ta nejméně přesná třída, která bude ovšem přesnější než GPS, bude k dispozici zdarma. Dále se počítá s další třídou pro soukromý sektor, s ještě vyšší přesností, ale její použití bude zpoplatněno. Zbývající tři třídy nebudou dostupné pro civilní obyvatelstvo a budou se využívat například pro řízení letového provozu, pro záchranné služby nebo pro armády a bezpečnostní složky států. O tomto systému se mluví již od roku 1999, jeho zprovoznění je určeno na rok 2014 a neustále je datum zprovoznění odsouváno, protože momentálně mají státy v EU finanční problémy a celá Evropská Unie prochází krizí. Cena celého projektu je v současné době odhadována na 5,3 miliard Euro. Pro Českou republiku je významné, že v roce 2010 byla za centrum celého projektu zvolena Praha. Systém by měl mít 30 družic a zřejmě nebude pouze evropský. Součástí programu je totiž například také Jižní Korea a Maroko. Po přechodnou dobu spolupracovala na vývoji také Čína, ale v roce 2006 z programu vystoupila a začala vyvíjet vlastní program jménem Beidou navigation system [12].

2.2 Fyzická realizace GPS

Systém GPS je rozdělen na tři segmenty, z nichž každý má jiný úkol. Velmi často jsou takto realizovány i jiné polohovací systémy.

Uživatelský segment je tvořen přístroji, které používají uživatelé. Jde o jediný segment, se kterým fyzicky přijdou do styku.

Kosmický segment je tvořen soustavou umělých družic obíhajících Zemi po přesně specifikovaných drahách. Družice mají několik typů oběžných drah (nízké, střední, vysoké, geostacionární, kruhové a eliptické), dále mají specifickou výšku a sklon. Družice jsou v různém počtu rozmístěny na různých oběžných drahách. Každá družice má své číslo a vysílá jiný kód. Uživatelský přijímač je pak schopen ji při měření polohy správně identifikovat a na základě signálů z několika družic určit svou polohu. Systém má při plném provozu 24 družic v 6 orbitálních drahách s inklinací 55 stupňů, které jsou ve výšce 20200 km a obíhají zemi s periodou 12 hodin.

Řídící segment je tvořen pozemními stanicemi. Tyto stanice sledují družice, vyhodnocují jejich chování, určují parametry oběžných drah, vyhodnocují korekční parametry, vysílají aktualizované údaje na družice a podobně. Pro tento úkol existují 3 typy pozemních stanic. Jde o monitorovací stanice, které jsou rozmístěny tak, aby mohly sledovat co největší počet družic po co nejdélší možnou dobu, monitorují jejich signály a ty pak posílají do řídicích stanic. Pak jsou zde dvě hlavní řídicí stanice, které analyzují signály z monitorovacích stanic a modelují chování kosmického segmentu, určují korekční parametry a ty pak posílají komunikačním stanicím, které je předávají družicím. Komunikační stanice slouží pro nastavování parametrů družic, hlavně korekčních parametrů atomových hodin, nacházejících se na družicích, protože základem přesného určení polohy je přesný čas. Tyto stanice umožňují také snížit přesnost určování polohy pro civilní sektor, pokud jde o vojenský systém pro určení polohy jako v případě GPS [2].

2.3 Určování polohy pomocí GPS

Přijímač určuje svou polohu na základě své vzdálenosti k několika družicím navigačního systému. Systém pracuje v pasivním módu, to znamená, že přijímač signál pouze přijímá, ale nevysílá. To hlavně kvůli energetické náročnosti při přenosu signálu na velkou vzdálenost a velikosti antén, které by k takovému přenosu byly potřeba. Dalším důvodem pro pasivní vysílače je původ technologie, protože původně vznikla pro vojenské účely a případné vysílání signálu by umožňovalo odchytení signálu a tedy prozrazení.

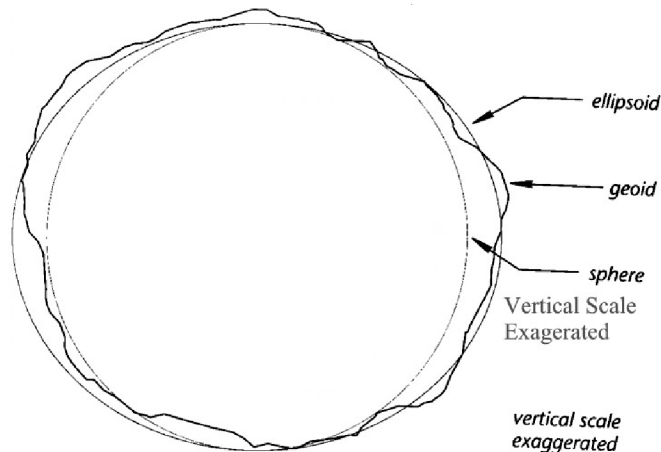
Naprosto klíčovým aspektem pro určování polohy je přesný údaj o čase. Proto jsou družice vybaveny atomovými hodinami. Atomové hodiny jsou oproti jakýmkoli jiným hodinám velmi přesné, k měření času používají rezonanční frekvence atomů. Jejich přesnost je tak velká, že k odchylce 1 sekundy dojde za 150 milionů let. Pro úspěšné určení polohy potřebuje přijímač přijímat signál vždy nejméně ze 4 družic. Signál první družice je použit pro synchronizaci času přijímače s časem GPS a zbývající 3 družice jsou pak použity pro samostatné dálkoměrné měření. Vzdálenost přijímače od družice se vypočítá právě na délce časového intervalu mezi odesláním signálu z družice a jeho příjmem na přijímači. Po krátkou dobu je přijímač schopen pracovat i za dostupnosti pouhých 3 družic. Po krátkou dobu proto, že přesnost hodin v přijímači je o několik řádů nižší než přesnost atomových hodin na družicích, proto se při dlouhodobé dostupnosti pouhých 3 družic zvětšuje chyba výpočtu polohy, protože i velmi malá odchylka času má za následek snížení přesnosti vzdálenosti od družice a tím pádem chybu při výpočtu polohy [2].

2.4 Přesnost určení polohy a omezení funkčnosti GPS

GPS má teoreticky globální dostupnost, ale reálná přesnost a dostupnost GPS závisí na mnoha faktorech. Vzhledem ke vzdálenosti mezi družicí a přijímačem signálu je signál, který doputuje k přijímači, poměrně slabý. Důsledkem tohoto faktu je, že přijímače nejsou většinou schopny signál zachytit v budovách, podzemních objektech a jiných místech bez přímého výhledu na oblohu. Vliv na sílu signálu má také okolní vegetace a ráz krajiny. Problematické jsou zejména kopce, kaňony, skály, městská zástavba a jiné konstrukce. Tyto faktory mají velký vliv na schopnost a přesnost určení polohy. Platí, že čím více máme aktuálně dostupných družic, tím přesnější údaj o poloze dostaneme. Je také důležité, aby družice, ze kterých přijímáme signál, byly co nejdále od sebe a rozloženy po celé ploše, jinak může dojít k dalším výchyilkám. Přijímač pak vytvoří z dostupných družic kombinace pro výpočet polohy a výsledek průměruje. K ohodnocení přesnosti se používá parametr DOP (*Dilution of Precision*). Na základě jeho hodnoty je možné předpovědět přesnost určení polohy. Některé objekty, zejména kovové střechy, způsobují odrazy a násobné šíření signálů. Přijímač pak není schopen poznat, o jaký signál se jedná a bude vnášet chybu do určení polohy. Velký vliv na přesnost má i typ přijímače a použitá metoda měření. Určení polohy pomocí signálu totiž není jednoznačné a neprobíhá jen jedním daným způsobem. Použitá metoda velmi záleží na kvalitě přijímače a také na oblasti jeho použití. Například u geodetických přístrojů je nutné mnohem přesnější určení polohy, než u přijímačů pro navigaci v automobilech. Běžně dosahovaná přesnost u současných turistických a navigačních přijímačů je v rozmezí 5 až 15 metrů. Některé přijímače jsou vybaveny i možností přijímání korekcí signálů GPS ve formě podpůrných systémů označovaných jako WAAS. Tyto korekce jsou vysílány z geostacionárních družic, které jsou umístěny nad rovníkem a pomáhají přijímači zpřesnit polohu a výšku. Další omezení u GPS je to, že pokud se přijímač nepohybuje, není schopen určit svou orientaci a tedy směr pohybu. To je celkem logické, a proto jsou některé přijímače vybaveny navíc elektromagnetickým kompasem [2].

2.5 Určení nadmořské výšky

Určení výšky pomocí GPS není tak jednoduché, jak by se mohlo zdát. Nadmořská výška přepočtená ze souřadnic je vztažena k matematické reprezentaci Země, neboli k ploše tzv. referenčního elipsoidu. Pomocí GPS určujeme tedy výšku nad tímto elipsoidem, nikoliv skutečnou nadmořskou výšku. Pro praktické aplikace nás ale zajímá spíše nadmořská výška nad geoidem, který věrněji reprezentuje tvar zemského povrchu. Geoid se více či méně odlišuje od referenčního elipsoidu a proto je nutné pro přesnější výpočet výšky zjistit výškový rozdíl mezi geoidem a elipsoidem. Pro ilustraci viz obrázek 1. Tento rozdíl je buď známý, nebo lze provést kalibraci u bodu s přesně udanou nadmořskou výškou. Pro Českou republiku se tento rozdíl pohybuje v rozmezí 42,5 a 47 metrů. Některé přijímače samy provádějí tuto korekci v závislosti na aktuální poloze, proto nelze absolutně určit přesnost přijímačů GPS při určování nadmořské výšky, většinou se však pohybuje v rozmezí mezi 15 a 40 metry. Přijímače vybavené barometrickým výškoměrem mají přesnost samozřejmě vyšší, udává se jeden až dva metry.



Obrázek 1: Rozdíl mezi geoidem a elipsoidem [11].

2.6 Souřadnicový systém pro GPS

Pro GPS se používá souřadnicový systém WGS 84 (World Geodetic System 1984). Tento standard byl vydán Ministerstvem obrany USA a definuje souřadnicový systém, referenční geoid a elipsoid pro geodézii a navigaci. Byl vytvořen na základě měření polohového systému TRANZIT. Jde o pravotočivou kartézskou soustavu souřadnic se středem v těžišti Země. Systém se hlavně díky GPS celosvětově rozšířil, zde ovšem nastává problém. Některé státy, například Česká republika, používají jiné souřadné systémy. Ty pak mohou být odlišné od systému WGS 84 a mezi jednotlivými souřadnicovými systémy se musí provádět složité přepočty. V České republice je závazné tzv. Křovákovo zobrazení, které vzniklo po 1. světové válce tak, aby co nejvíce vyhovovalo tehdejšími potřebám naší republiky, ale nyní bohužel dochází k odlišnostem oproti WGS 84.

WGS 84 vyniká svou globální přesností oproti starším systémům jako WGS 72, WGS 66 a WGS 60. Na druhou stranu se již začínají prosazovat novější a přesnější modely jako EGM 96.

Parametry definující referenční elipsoid WGS 84 jsou:

- délka hlavní poloosy: $a = 6\,378\,137\text{m}$.
- převrácená hodnota zploštění: $1/f = 298,257223563$.
- úhlová rychlost Země: $\omega = 7\,292\,115 \cdot 10^{-2}$.
- součin hmotnosti Země a gravitační konstanty: $GM = 39\,860\,000,9 \cdot 10^8 \text{ m}^3/\text{s}^2$.

Z těchto definujících parametrů je také možno vypočítat například velikost vedlejší poloosy elipsoidu, $b = 6\,356\,752,3142 \text{ m}$.

2.7 Výpočet vzdálenosti z GPS souřadnic

V souvislosti s modelem WGS 84 bylo třeba najít způsob, jak vypočítat vzdálenost mezi dvěma body. Pro tento model existuje velmi přesný výpočet pomocí Vincentiho rovnic [3]. Tento postup je ovšem časově a výpočetně velmi náročný, protože obsahuje několik počátečních rovnic a následnou iteraci pro co nepřesnější výpočet. Tím jsme schopni se teoreticky dostat na přesnost v řádu desetin milimetrů i na velké vzdálenosti. To je pro potřeby této práce zbytečné, protože přesnost GPS takovýchto hodnot v žádném případě nedosahuje, rovněž by byla přesnost výpočtu omezena přesností

uložení čísel v plovoucí řádové čárce v počítači. Navíc se dá očekávat, že každá trasa bude mít mnoho bodů a bylo by potřeba počítat vzdálenost mezi každými dvěma sousedními body a tento výpočet by trval velmi dlouho. Proto bylo nutné najít jinou alternativu. Možných řešení se nabízí hned několik.

Základním zjednodušením je nepoužívat výpočet vzdálenosti na elipsoidu, ale na kouli a na ploše. Vzhledem k tomu, že se dá očekávat, že body na trase z GPS budou blízko u sebe, nevnese toto omezení do výpočtu vzdálenosti velkou chybu. Jako nejvhodnější řešení se jeví rozložit si úsek mezi body na dva kruhové oblouky, které na sebe budou kolmé. Jedna z nich bude pro zeměpisnou šířku a druhá pro zeměpisnou délku. Poté se vypočítá vzdálenost mezi šířkami a délkami obou bodů na základě vzorce pro obvod kruhové části kruhového oblouku [1]:

$$d = \frac{2 \cdot \alpha \cdot \pi \cdot r}{360^\circ} \quad (2.1)$$

kde α je rozdíl úhlů ve stupních mezi dvěma body a r je poloměr kružnice. Poté provedeme další zjednodušení, a sice že úsek mezi dvěma sousedními body budeme považovat za přímku. Potom vzdálenost mezi nimi vypočteme na základě Pythagorovy věty [1]:

$$l = \sqrt{(\text{rozdíl délek})^2 + (\text{rozdíl šířek})^2} \quad (2.2)$$

Nyní se ale nabízí další otázka, jaký použít poloměr Země. Určeny jsou tři – poloměr na rovníku (6.378,135 km), poloměr na pólu (6.356,75 km) a střední poloměr (6.372,7955 km).

První možností je vypočítat na základě znalosti délky hlavní a vedlejší poloosy v elipsoidu z definice modelu WGS 84 přesný průměr Země v každém bodě a pro výpočet vzdálenost mezi dvěma body použít aritmetický průměr hodnot poloměrů v těchto dvou bodech. Čili bychom postupovali podle následujících vzorců [1]

$$r_1 = \sqrt{a * \cos \alpha_1^2 + b * \sin \alpha_1^2} \quad (2.3)$$

$$r_2 = \sqrt{a * \cos \alpha_2^2 + b * \sin \alpha_2^2} \quad (2.4)$$

$$r = \frac{r_1 + r_2}{2}, \quad (2.5)$$

kde α_1 je zeměpisná šířka prvního bodu, α_2 je zeměpisná šířka druhého bodu, a je délka hlavní poloosy elipsy a b je délka vedlejší poloosy elipsy.

Tento postup je bohužel také dosti náročný, protože se neustále počítají nové poloměry. Další možností by bylo například provést výpočet průměrného poloměru Země na základě jen prvního a posledního bodu v celé trase. Nejjednodušší je samozřejmě použití středního poloměru Země, protože pokud nebude trasa dlouhá tisíce kilometrů, chyba bude mít přijatelnou velikost a tento postup je také výpočetně nejméně náročný.

3 Formáty souborů pro uložení dat z GPS

Pro uložení informací z GPS existuje vícero různých formátů. Tato data se ukládají nejčastěji do souborů se strukturou XML dokumentu (formáty GPX a KML). Existuje mnoho programů pro konverzi mezi jednotlivými formáty, proto je plně postačující podporovat pouze dva formáty, a to GPX a KML.

3.1 Formát GPX

Asi nejrozšířenějším formátem pro uložení dat z GPS je formát GPX [6] (*GPS eXchange*). Výhodou tohoto formátu je, že je otevřený a za jeho použití se nemusí platit. Je schopen pojmut důležité informace o trase, hlavně souřadnice, nadmořskou výšku a čas. Umožňuje tak jednotlivým uživatelům GPS zařízení sdílet mezi sebou informace o svých trasách. Formát je velmi rozšířený, proto se používá například pro zobrazení v satelitních mapách nebo pro geotagging fotografií. Tento formát obsahuje jednotlivé body na trase, které mohou být sdružovány do množin označovaných jako *track* nebo *route*. Teoreticky toto rozdělení je podle toho, zdali jde o skutečnou trasu, kterou uživatel absolvoval (*track*) nebo o trasu, o jejímž absolvování uvažuje do budoucna (*route*). *Route* navíc neobsahuje údaje o čase. GPX formát má více možných atributů ke každému bodu, povinné jsou zeměpisná šířka a délka, mezi ty volitelné patří hlavně nadmořská výška a čas. Formát umožňuje také přiřazení jména ke každému bodu na trase.

Jak bylo uvedeno výše, soubor ve formátu GPX je vlastně XML soubor. Proto na začátku každého GPX souboru je standardní XML hlavička, za ní následuje GPX hlavička, která udává verzi GPX, software, kterým byl dokument vytvořen a podobně. Za hlavičkou pak následuje jméno celého záznamu. Poté začíná buď element `trk` pro typ *track* nebo `rte` pro *route*. U `trk` následuje jméno tracku a pak jednotlivé segmenty v elementu `trkseg`. V elementu `trkseg` jsou jednotlivé elementy `trkpt`, což už jsou jednotlivé body, které byly zaznamenány. Element `trkpt` obsahuje jako atributy zeměpisnou šířku a délku. Uvnitř tohoto elementu jsou volitelné elementy každého bodu. Z těch je pro účely navržené aplikace důležitá hlavně nadmořská výška.

Zde je příklad uložení jednoho bodu v souboru GPX:

```
<trkpt lat="49.66719437" lon="18.31266022" >  
  <ele>52.308</ele>  
  <time>2011-03-12T09:27:22Z</time>  
  <name>trkpt-2011-03-12T09:27:22Z</name>  
</trkpt>
```

Výpis 1: Demonstrace uložení bodu ve formátu GPX.

3.2 Formát KML

Dalším rozšířeným formátem pro uložení geografických dat je formát KML (*Keyhole Markup Language*). Jak lze zjistit [4], jde opět o formát souboru XML. KML je formát neodmyslitelně spjatý se společností Google a je známý hlavně díky aplikaci Google Earth. Jeho původ je ale u společnosti Keyhole, která tento formát používala ve své aplikaci Earth Viewer. V roce 2004 tuto společnost koupila firma Google a přejala tento produkt. Formát KML v sobě ukládá množství geografických dat, umožňuje například vizualizaci objektů, která je využita v aplikaci Google SketchUp, umožňující export 3D objektů do aplikace Google Earth.

Samotný formát KML je pak méně přehledný než formát GPX, hlavně kvůli definici mnoha dat, sloužících pro zobrazení trasy v Google Earth. Stejně jako u formátu GPX začíná KML klasičkou XML hlavičkou, za kterou následuje KML hlavička a jméno celého záznamu. Pak následuje velké množství dat pro zobrazení v Google Earth. Umožňují například definici barvy čáry, přidání různých ikoněk, zoom na místo startu, sklon úhlu pohledu a podobně. Souřadnice jednotlivých bodů jsou pak uloženy na dvou místech. Nejprve v elementu `LineString` v sekci `coordinates`, a to jako seznam trojic zeměpisná šířka, délka a nadmořská výška. Znovu jsou pak souřadnice uvedeny v elementech `Placemark`, kde je možnost dodat další informace o bodu, hlavně textový popis a čas.

Zde je příklad uložení jednoho bodu v souboru KML. Tento fragment demonstruje složitost uložení bodu:

```
<Placemark>
  <name>TIME: 15:18:48</name>
  <visibility>0</visibility>
  <description>
    <![CDATA[TIME: 05-MAR-11 15:18:48<br>LATITUDE: 47.735954
    N<br>LONGITUDE: 14.912723 E<br>HEIGHT: 1081.101 m]]>
  </description>
  <TimeStamp>
    <when>2011-03-05T15:18:48.000Z</when>
  </TimeStamp>
  <Point>
    <coordinates>14.912723,47.735954,1081.101</coordinates>
  </Point>
</Placemark>
```

Výpis 2: Demonstrace uložení bodu ve formátu KML.

4 Technologie použité při práci

Během implementace bylo využito několika technologií a knihoven. Hlavní z nich je programovací jazyk Java, dále jde o překladový skript ant, technologie XML, HTML, a také o volně šiřitelnou knihovnu JXMapKit.

4.1 Programovací jazyk Java

Jako implementační jazyk byl zvolen programovací jazyk Java, díky osobním zkušenostem s tímto jazykem a také díky jeho objektově orientovaným vlastnostem.

Programovací jazyk Java [8] se poprvé objevil v roce 1995 a vytvořila jej společnost Sun Microsystems, která je nyní součástí společnosti Oracle Corporation. Syntaxe vychází z jazyků C a C++, oproti C++ má však jednodušší objektový model a práce s tímto jazykem probíhá na vyšší úrovni a uživatel se nemusí starat o správu paměti, kterou provádí Garbage Collector. Program v Javě je přeložen do mezijazyka (*bytecode*), který je následně interpretován pomocí Java Virtual Machine. Java Virtual Machine (JVM), běhové prostředí Javy, zaručuje, že se program v Javě interpretuje stejně na všech zařízeních podporujících Javu. Kromě počítačů může jít o mobilní telefony, ale také o množství jiných elektronických zařízení jako jsou pračky a ledničky. JVM se stará o kompletní běh aplikace, pomocí Garbage Collectoru se stará o správu paměti, spravuje vlákna a vytváří a ruší objekty. Java se v dubnu 2011 podle [9] dostala na první místo v žebříčku popularity programovacích jazyků.

Java má velmi rozsáhlé API pokrývající většinu běžných programovacích problémů a tím velmi usnadňuje práci. Během práce byly použity hlavně knihovny pro používání a správu kolekcí (List, Map), pro tvorbu GUI (Swing, AWT) a pro zpracování XML dokumentů (DOM). Použití knihoven pro tvorbu GUI je v Javě poměrně složité a rozsáhlé, proto bylo pro návrh uživatelského rozhraní integrované použito prostředí NetBeans, které je schopno GUI generovat automaticky.

4.2 Sestavovací systém Apache Ant

Pro překládání a spouštění programů v Javě se často používá utilita Ant. Je podobná programu Make, který slouží pro překlad programů v jazyce C. Na rozdíl od Make je překladový skript pro Ant tvořen souborem ve formátu XML. Ant je napsán v Javě, a proto je velmi dobře přenositelný.

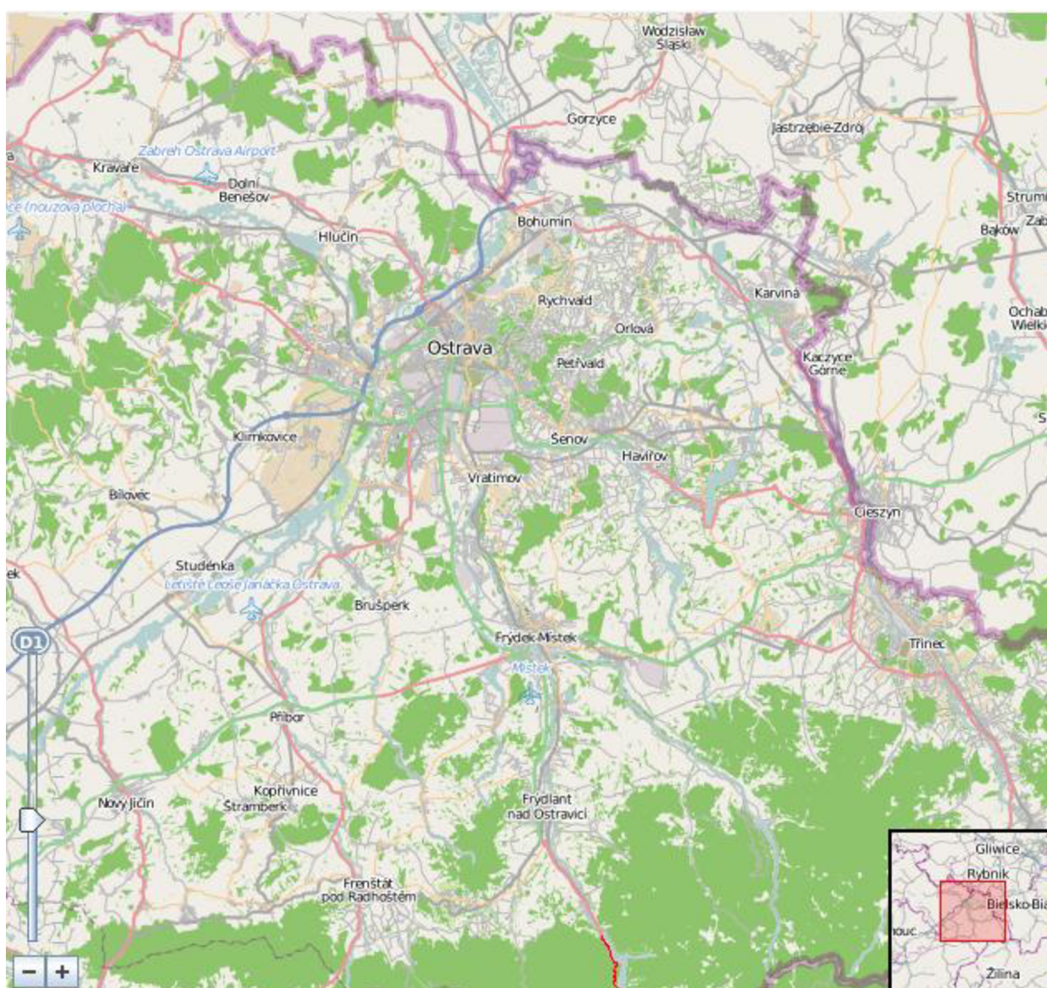
Pro definici překladu je zapotřebí vytvořit soubor `build.xml`. Pomocí XML značek v tomto souboru specifikujeme zdroj a cíl překladu, možnost připojení externích knihoven a podobně. Pomocí nadefinovaných informací ve skriptu můžeme program přeložit, spustit, nebo k němu automaticky vytvořit dokumentaci, takzvaný `javadoc`.

Součástí navrženého řešení je rovněž překladový skript pro Ant. Překlad pak probíhá pomocí následujících příkazů:

- `ant compile` – provede překlad
- `ant run` – spustí přeložený program
- `ant doc` – generuje dokumentaci (`javadoc`) do adresáře „doc“
- `ant clean` – vymaže soubory `class`

4.3 Knihovna JXMapKit

Pro zobrazení mapy byla použita open source externí knihovna JXMapKit [13]. Ta velmi usnadnila práci při implementaci zobrazení mapových podkladů. JXMapKit je zděděna od standardní třídy JPanel, což je jedna ze základních entit ve standardu Swing komponenty Javy. Tato knihovna umožňuje nastavení zdroje mapových dat. V případě vytvořené aplikace byla použita mapa OpenStreetMap, převážně z důvodu její svobodné licence. Součástí instance této třídy je rovněž ovládání mapy, jenž je stejné jako u webových stránek obsahujících mapové podklady. Tedy pro přiblížení mapy slouží rolovací kolečko myši a pro pohyb po mapě stačí uchopit mapu a táhnout myši v požadovaném směru. V panelu JXMapKitu je také zobrazena miniatura mapy pro lepší orientaci a posuvník pro přiblížení mapy. Panel umožňuje pochopitelně také zobrazení bodů a tras do mapy. Obrázek 2 demonstruje vzhled mapového panelu.



Obrázek 2: Vzhled mapového panelu.

4.4 Technologie XML

Tato aplikace by se neobešla bez zpracování XML dokumentů, hlavně proto, že vstupem jsou soubory s geografickými daty ve formátech KML a GPX. Oba tyto typy souborů jsou ve formátu XML, jak již bylo popsáno výše.

XML (*Extended Markup Language*) [7] je značkovací jazyk, který byl vyvinut konsorciem W3C. XML vychází ze starší specifikace SGML (*Standard Generalized Markup Language*). Má velmi širokou aplikaci, umožňuje uložit různé typy dat a primárně slouží pro výměnu dat mezi aplikacemi. Tento jazyk obsahuje tagy (značky), pomocí kterých je určena struktura a hierarchie dat. Jednotlivé značky se uzavírají do špičatých závorek a jsou párové. V případě potřeby mohou mít značky i atributy. Každý prvek tedy musí mít počáteční a koncovou značku. Pokud je toto základní pravidlo porušeno, dojde k chybě při zpracování.

Pro zpracování XML dokumentu se používají dva přístupy. Prvním z nich je jednodušší SAX (*Simple API for XML*), pracující na principu sériového procházení dokumentu. Postupně se načítají jednotlivé úseky a v těch se hledají informace. Výhodou tohoto přístupu je menší paměťová náročnost, protože nenačítá do paměti celý XML dokument, ale na druhou stranu je pak pomalejší vyhledávání, protože je potřeba neustále načítat další části dokumentu. Druhým přístupem je DOM (*Document Object Model*), ten počítá s objektovým přístupem k dokumentu a pracuje tak, že načte celý dokument do paměti a vytvoří jeho objektovou reprezentaci. Jednotlivé elementy jsou při tomto přístupu uloženy do stromové struktury. DOM je paměťově náročnější kvůli nutnosti načíst celý dokument do paměti, ale vyhledávání pak probíhá rychleji. Tento přístup je vhodný v situacích, kdy vyžadujeme náhodný přístup k různým částem dokumentu.

4.5 Technologie HTML a CSS

Dalším značkovacím jazykem použitým v této práci je jazyk HTML (*HyperText Markup Language*) [10]. Ten stejně jako XML patří do rodiny jazyků SGML. Má ovšem jiné využití, protože slouží k tvorbě webových stránek. Syntaxe je velmi podobná syntaxi XML, ale u HTML není tak přísná. Technologii HTML vytvořená aplikace využívá pro export informací o trase do uživatelsky snadno přístupného formátu.

Původně HTML sloužilo jak pro definici obsahu stránky, tak pro definici vzhledu. Od tohoto konceptu se však již upouští a dochází k oddělení těchto dvou funkcí. K definici vzhledu se používá CSS, neboli kaskádové styly. Ty se pak starají o celý vzhled dokumentu a v souboru HTML je pak pouze faktický obsah stránky. To umožňuje jednotnou správu vzhledu celého webu. Součástí aplikace je proto také definovaný styl v souboru CSS, který se ve všech vygenerovaných HTML stránkách použije.

5 Návrh aplikace

Na úvod návrhu aplikace bylo nutné zvážit, co všechno by výsledná aplikace měla umět:

- Vykreslení výškového profilu trasy do grafu. Možnost vykreslovat větší škálu informací do grafu, například závislost převýšení na čase, závislost výšky na vzdálenosti a podobně.
- Podpora minimálně formátů GPX a KML.
- Možnost zobrazení trasy do mapy.
- Zjištění různých informací a statistik o trase, například výpočet převýšení, procentuální rozložení stoupání a klesání, průměrné rychlosti při klesání/stoupání a podobně.
- Zobrazení informací o bodech do tabulky a možnost vyloučit některé body z trasy kvůli jejich nesmyslnosti vlivem špatného signálu GPS.
- Převody mezi formáty KML a GPX.
- Možnost zpracovávat pouze určitou část trasy, tedy jakési lokální informace a zvýraznění bodů této části v grafu a v mapě.
- Správu všech uživatelských tras, což umožňuje globální statistiky.
- Export grafů nebo map s trasou do obrázku ve formátu PNG.
- Export údajů a statistik o trase do souboru formátu HTML včetně získaných grafů a map.
- Vytvoření uživatelsky co nejpřehlednější a nejintuitivnější aplikace.

5.1 Návrh grafického uživatelského rozhraní

Velmi důležitou součástí projektu je grafické uživatelské rozhraní. Výsledná aplikace by měla být použitelná pro co nejširší veřejnost, proto musí mít co nejintuitivnější a nejjednodušší ovládání. V průběhu vývoje bylo provedeno nespočet změn ve vzhledu a ovládání aplikace. Nejdůležitějším faktorem byl celkový vzhled aplikace, tedy to, co uživatel uvidí, když program spustí.

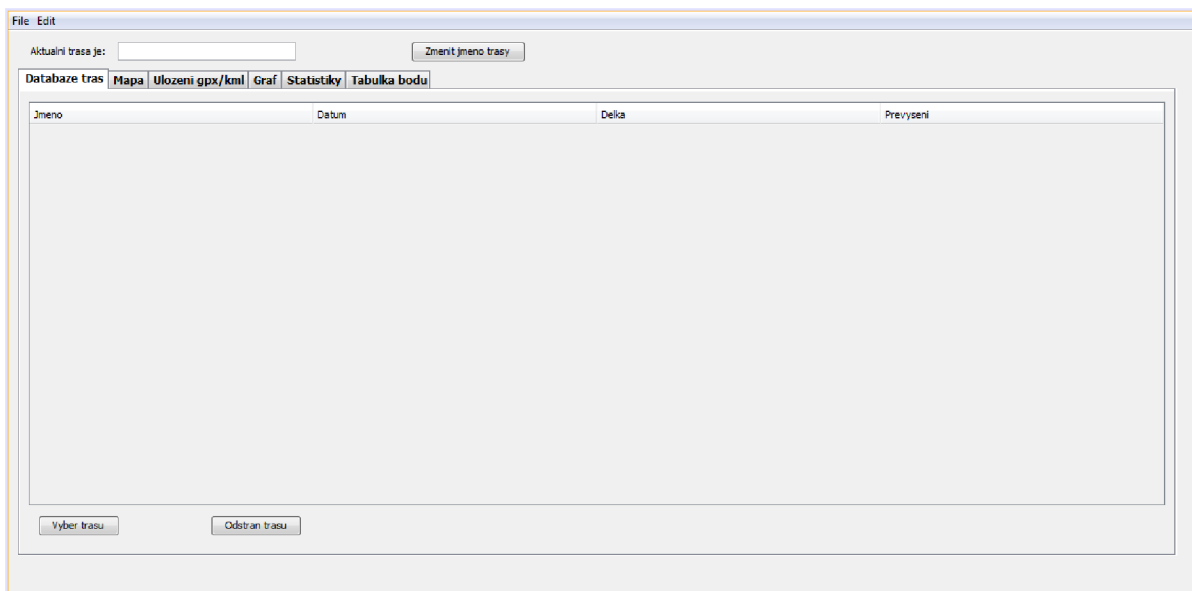
První návrh se velmi brzy ukázal jako naprosto nevhodný. Šlo v podstatě o aplikaci sestávající z velkého množství oken. Každé okno by obsahovalo pouze jeden typ informace, čili v jednom by byla zobrazena mapa, ve druhém grafy, v dalším informace o trase a podobně. Uživatel by si pak při spuštění aplikace mohl zvolit, kterou z těchto informací chce zobrazit. Výsledek by byl ale značně nepřehledný, v oknech by byl zmatek a muselo by se mezi nimi stále přepínat. Navíc by uživatel nebyl schopen pozorovat všechny změny v údajích o trase při úpravě bodů na trase a rovněž by trvalo zbytečně dlouhou dobu, než by se mu zobrazily aspoň nějaké relevantní informace, protože návrh počítal s tím, že nejprve vybere trasu, se kterou chce pracovat, pak vybere typ informací, které chce zobrazit a až v dalším okně by došlo k zobrazení těchto informací. Z implementačního hlediska by pak byla náročná komunikace mezi jednotlivými okny. Proto bylo toto rozvržení aplikace poměrně brzo zavrhnuto a nedošlo ani k jeho implementaci.

Druhý návrh znamenal zlepšení oproti návrhu prvnímu. Došlo ke sjednocení jednotlivých oken do jednoho za pomoci panelového přepínače. Oproti prvnímu konceptu měl tento výhodu hlavně v tom, že uživatel měl jedno okno, kde mohl přepínat mezi jednotlivými panely. Aplikace by byla výrazně přehlednější. Panelů v tomto konceptu mělo být šest:

- Panel s tabulkou všech tras uživatele.
- Panel s mapou.
- Panel pro export tras do formátu GPX a KML.

- Panel pro tvorbu grafu.
- Panel s údaji o trase.
- Panel s tabulkou všech bodů na trase.

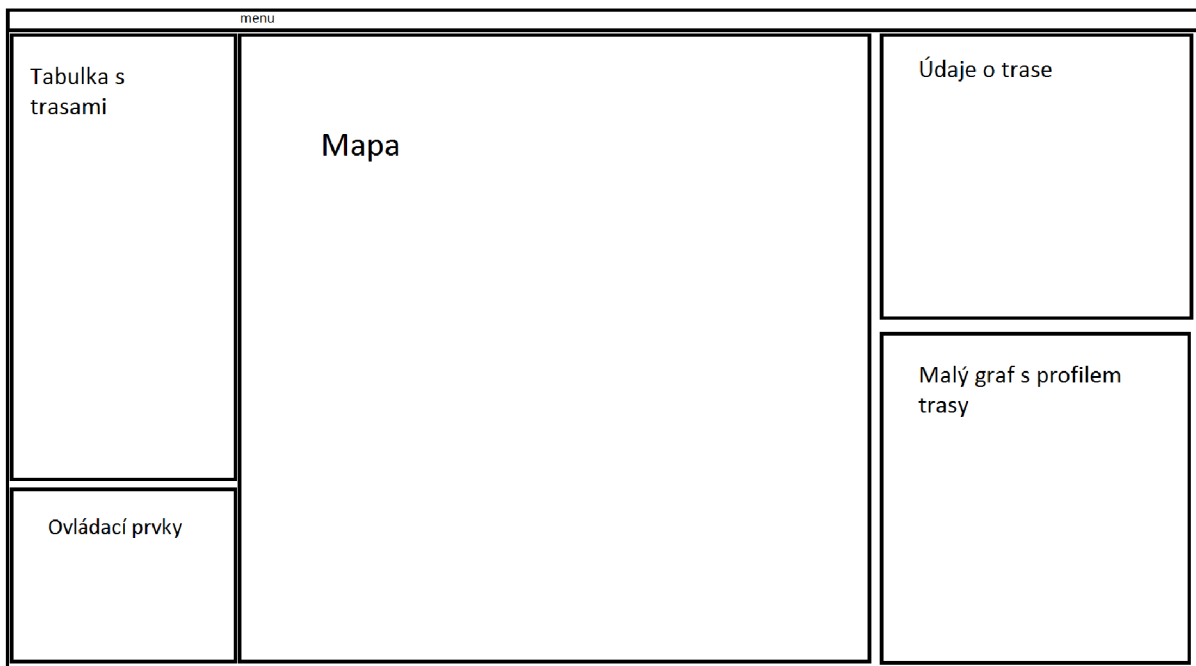
Tento koncept se již dočkal své funkční implementace. Vzhled této implementace je na obrázku 3.



Obrázek 3: Druhý koncept GUI.

Tento koncept byl sice lepší, ale stále měl několik zásadních nedostatků. Šlo hlavně o to, že by uživatel musel neustále přepínat mezi panely a stále by mu nebylo umožněno sledovat vedle sebe více druhů informace a následky své práce s aplikací. Rovněž bylo během práce s aplikací zjištěno, že některé funkce jsou využívány mnohem méně často než jiné, hlavně export do souborů GPX a KML, využívaný velice zřídka a přitom zabírající celý jeden panel. Naopak mapu bylo potřeba vidět téměř neustále. Na druhou stranu se velmi usnadnila komunikace mezi jednotlivými entitami, protože bylo možno vložit všechny komunikující prvky do jedné třídy. Navzdory této výhodě byl celý koncept prohlášen za nevyhovující.

Třetí koncept vznikl na základě pečlivého zvážení kladů a záporů druhého konceptu. Jeho hlavním cílem bylo co nejlépe využít plochu celé obrazovky a zobrazit důležité věci tak, aby je uživatel viděl zároveň. Naopak méně používané věci byly přesunuty do pozadí, aby uživatel nebyl zaskočen přemírou ovládacích prvků a možností, které zřídka kdy využije. To se týkalo například výše zmíněné možnosti exportu tras do formátu GPX a KML přesunuté ze samotného okna do menu. Dále bylo bráno na vědomí, že uživatel potřebuje mít prostor pro údaje o trase a rovněž pro mapu, do které se mu jeho trasa zobrazí. Stejně tak by bylo vhodné hned po spuštění aplikace vidět aspoň malý graf výškového profilu a možnost vyhotovení jiného a většího grafu by bylo vhodné přesunout někam jinam. Také by mělo být pro uživatele jednoduché vracet se k trasám, se kterými pracoval již dříve, ale nepotřebuje o nich nutně mít detailní informace ještě před výběrem dané trasy. Návrh rozvržení tohoto okna je ukázán na obrázku 4. Z návrhu je vidět, že hlavní prostor dostala mapa, protože se jedná o jednu z nejdůležitějších součástí. Do sekce „Ovládací prvky“ by měla být vložena možnost zobrazit souhrnné údaje o všech trasách, zobrazení tabulky s body, tvorba nové trasy ze zvýrazněných bodů a generování webové stránky s informacemi o trase.



Obrázek 4: Třetí koncept GUI.

Menu aplikace není v tomto návrhu příliš rozsáhlé a obsahuje pouze položky pro přidání nové trasy, export trasy do souborů GPX a KML a pro ukončení aplikace.

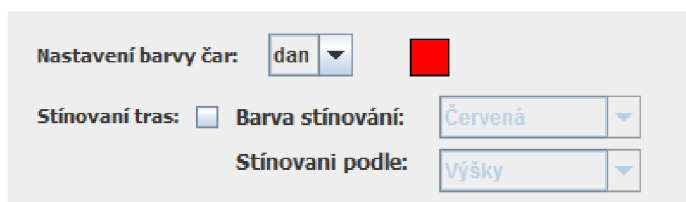
Součástí tohoto návrhu je rovněž snaha o intuitivní ovládání. Oproti druhému konceptu došlo k omezení počtu zbytečných tlačítek a políček. To se týká hlavně údržby tabulky se staršími trasami. V původní koncepci bylo třeba klepnout na různá tlačítka při volbě trasy, při jejím mazání a přejmenování. Nový koncept by měl umožnit volbu trasy označením daného řádku v tabulce a stiskem klávesy Enter. Naopak mazání má proběhnout po stisku klávesy Delete. Přejmenování pak změnou jména trasy přímo v tabulce a stiskem klávesy Enter. Pro usnadnění ovládání jsou pak použity klávesové zkratky pro ovládání menu.

5.2 Ovládání mapy

Ovládání mapy se rovněž snaží být co nejintuitivnější. Tedy přiblížení mapy by se mělo ovládat pomocí rolovacího kolečka myši a pohyb po mapě by měl reagovat na uchopení a tažení za pomoci levého tlačítka myši tak, jako na webových mapových portálech. V mapě samozřejmě dojde k zobrazení aktuálně vybrané trasy.

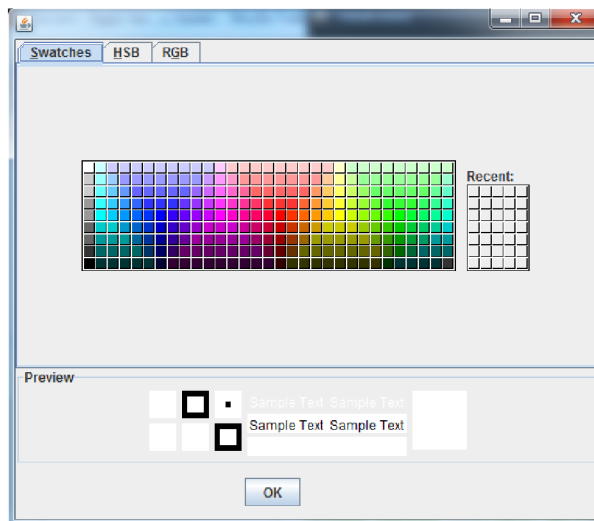
Vhodné bylo také naimplementovat funkci umožňující stínování trasy v mapě na základě rychlosti nebo výšky. Čím je daná veličina větší, tím tmavší je barva daného úseku trasy. Uživatel má možnost vybrat si z několika připravených barev.

Prostředky pro nastavení barvy a stínování trasy se nacházejí v panelu pod mapou. Obrázek 5 demonstruje vzhled tohoto panelu.



Obrázek 5: Panel pro ovládání mapy.

Na obrázku vidíme, že pokud je stínování trasy vypnuto, možnost volby stínování je neaktivní. Po zatržení tlačítka „Stínování tras“ se teprve aktivuje možnost volby parametrů stínování. Přepínání mezi jednotlivými hodnotami není potřeba nijak potvrdit, po každé volbě se provede aktualizace mapy. V případě, že je možnost stínování tras deaktivována, je pro vykreslení trasy v mapě použita barva, která je na pozadí čtverečku s barvou (v tomto případě červená). Uživatel má samozřejmě možnost vybrat si barvu trasy. Po kliknutí na onen barevný čtvereček se zobrazí okno pro výběr barvy. Vzhled tohoto okna ukazuje obrázek 6.



Obrázek 6: Volba barvy trasy.

Volba barvy trasy ovlivní také barvu čáry v grafu dané trasy. Při aktivaci stínování trasy se použije pochopitelně barva, která je zvolena v sekci „Barva stínování“.

5.3 Validace bodů

Vlivem různých chyb při příjmu signálu pomocí GPS přijímače může dojít k situaci, kdy se mezi body z trasy dostanou i body, které jsou na první pohled chybné. Může jít například o skokovou změnu převýšení nebo o prudkou změnu rychlosti pohybu v důsledku nepřiměřeně velké uražené vzdálenosti a podobně. V aplikaci by měla být možnost se s touto skutečností vypořádat.

Byly zvoleny dvě možnosti určení chybných bodů, a sice uživatelskou a automatickou. Automatická by měla odstranit co největší počet bodů, které jsou na první pohled špatné, ale zároveň by neměla být příliš přísná, aby nedošlo k chybnému odstranění bodů, které jsou správné. Uživatelská validace je určena spíše pro odstranění menšího počtu bodů, které automatická validace neodhalila. V případě, že dojde k označení korektních bodů za chybné, měl by mít uživatel možnost verdikt zpětně změnit, proto by při označení bodů za chybné nemělo dojít k jejich trvalému odstranění. Validace bodů by měla probíhat u tabulky se všemi body na trase, kde bude mít uživatel možnost buď automatické, nebo manuální validace. Automatická validace bude vyvolána uživatelem, nebude probíhat automaticky při načtení trasy.

Aby uživatel mohl pozorovat účinky validace na celou trasu, byla vložena tabulka s body pod panel se zobrazenou mapou, její přítomnost lze ovlivnit zatržením tlačítka, které je v sekci „Ovládací prvky“ (viz obrázek 4). Pokud je tlačítko zatrženo, je tabulka s body zobrazena, pokud ne, je zobrazeno ovládání mapy (viz sekce Ovládání mapy).

5.4 Možnost práce s vybranými body na trase

Do aplikace byla rovněž zahrnuta možnost práce s vybranými body či úseky na trase. To vede hlavně k zobrazení informací, týkajících se pouze vybraných bodů. Tyto body se rovněž označí křížkem v mapě a dojde ke zvýraznění i ve všech grafech, které jsou k dané trase zobrazeny.

Výběr bodů může proběhnout dvěma způsoby. První z nich je výběr bodů v tabulce s body a druhou možností je výběr úseku z grafu. Výběr v grafu probíhá dvojitým kliknutím prostředního tlačítka myši nad kterýmkoli grafem, tím se vybere úsek mezi těmito dvěma kliknutími.

Ze zvýrazněných bodů lze vytvořit novou trasu, která bude nezávislá na ostatních, čili se trasa samostatně zobrazí v mapě a dojde přidání této trasy do tabulky všech tras, což umožní například export do souborů KML nebo GPX bez závislosti na původní trase. Z takto vytvořené trasy lze vybírat další úseky a tvořit z nich opět menší a menší trasy.

5.5 Zobrazení informací o trase

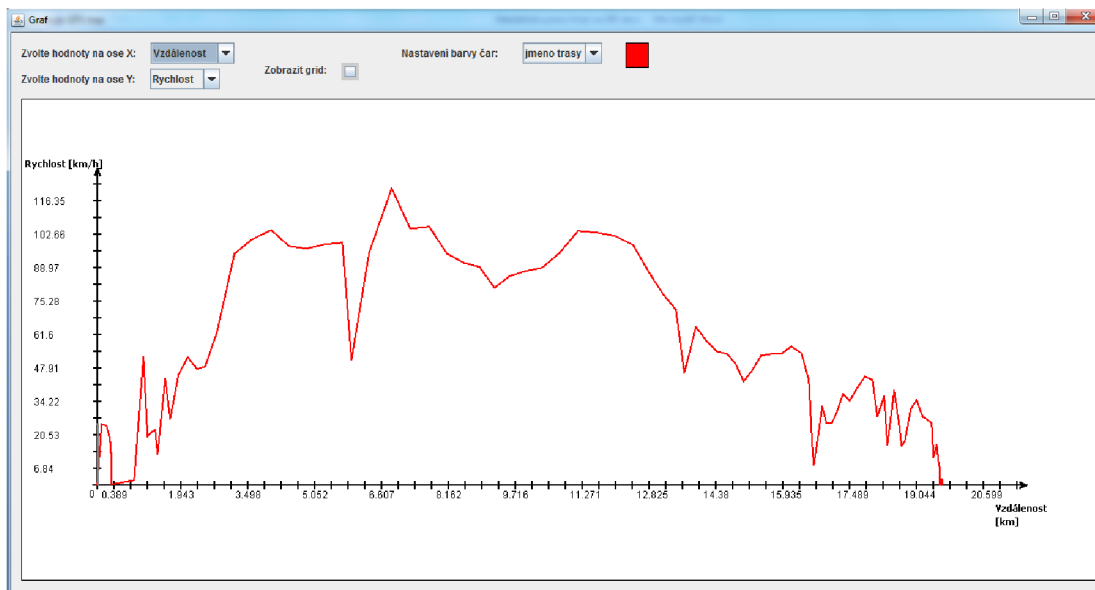
Uživatel má možnost zjistit množství údajů o celé trase a rovněž o zvýrazněném úseku. Tyto informace jsou zobrazeny přehledně do tabulky v pravém horním rohu aplikace. Je zde samozřejmě uvedeno jméno trasy, její délka, převýšení, čas a rychlost a pak také méně důležité údaje, které jsou však rovněž užitečné. Jedná se konkrétně o tyto:

- Délka klesání a stoupání.
- Průměrná rychlost ve stoupání a klesání.
- Podíl stoupání oproti celkové délce trasy.
- Maximální a minimální výška.
- Maximální a minimální rychlost.

Při výběru zvýraznění bodů na trase se do tabulky přidá další sloupec, kam se zobrazí lokální údaje, které jsou platné pro zvýrazněný úsek. Uživatel má tak možnost porovnat informace o svém zobrazeném úseku s informacemi o zbytku trasy.

5.6 Zpracování grafu

Při spuštění aplikace se uživateli zobrazí miniatura grafu demonstrující převýšení na trase v závislosti na uražené vzdálenosti. Pro přístup k dalším možnostem grafu slouží příslušné tlačítko pod miniaturou grafu. Okno grafu je ukázáno na obrázku 7.



Obrázek 7: Okno grafu.

Z obrázku je vidět, že uživatel může zvolit hodnoty na obou osách. Na osu x může nanášet vzdálenost, čas nebo pořadové číslo aktuálního bodu. Na osu y pak lze vykreslovat rychlost nebo převýšení. Dále si může zobrazit mřížku pro lepší orientaci v celém grafu. Pro určení hodnoty konkrétního bodu slouží pomocné čáry, které snadno určí hodnotu na ose X i ose Y. Tyto čáry se zobrazí při přejetí ukazatele myši přes graf. Samozřejmě je možné změnit barvu čáry v grafu. Způsob výběru barvy je stejný jako u mapy. Barva se po výběru použije rovněž v miniatuře grafu a v mapě. Jedna trasa má tedy pouze jednu barvu pro všechny zobrazení, což usnadňuje orientaci pro uživatele, který si tak nemusí pamatovat množství barev u jedné trasy.

5.7 Uložení dat

Veškerá relevantní data jsou uložena ve formátu XML, hlavně z důvodu přenositelnosti. Další způsob by teoreticky byl možný za pomoci některého databázového systému, ale to by vyžadovalo další programové vybavení na straně uživatele, čímž by vznikla závislost na konkrétním databázovém systému a velmi by tak utrpěla přenositelnost celé aplikace. Vzhledem k relativně malému počtu údajů je řešení za pomoci XML souborů vhodnější.

5.7.1 Databáze tras

Program interně používá dva typy XML souborů. Jeden soubor slouží k uložení databáze tras. V tomto souboru jsou uloženy základní informace o všech trasách, jejich jméno, délka, datum, převýšení, čas, jméno souboru s celou trasou a barva zobrazení trasy v mapě a grafu. Tento typ souboru je použit při spuštění programu, slouží pro načtení základního přehledu o trasách. Rovněž jsou v něm uloženy informace o trase, které se mohou častěji měnit, hlavně jméno trasy a barva zobrazení. Nové generování tohoto souboru je totiž poměrně jednoduché, protože soubor tohoto typu je mnohem kratší než soubor druhého typu. Při výběru trasy je z tohoto souboru získáno jméno souboru a celá trasa je pak načtena ze souboru s tímto jménem.

Struktura pak vypadá takto:

```
<tracks>
  <track actual="yes">
    <name></name>
    <date></date>
    <time></time>
    <length></length>
    <ele></ele>
    <filename></filename>
    <color></color>
    <htmlFilename></htmlFilename>
  </track>
</tracks>
```

Výpis 3: Demonstrace uložení trasy.

5.7.2 Uložení jednotlivých tras

Druhý typ interního XML slouží pro uložení jednotlivých tras. K tomuto účelu byl navržen vlastní formát pro uložení dat, hlavně z důvodu potřeby nějakým způsobem sjednotit podporované formáty tak, aby se s nimi poté dalo jednotně pracovat. To je také vhodné pro zachování možnosti přidat další možný formát v budoucnu. Dalším důvodem také bylo vyřazení některých zbytečných dat hlavně ze souboru KML. V tomto souboru jsou uloženy komplexní informace o trase, jak jsou popsány v 5.5. Tyto údaje o trase následují seznam bodů. Uložení bodů má následující strukturu:

```
<points>
  <point valid="true">
    <alt></alt>
    <lon></lon>
    <ele></ele>
    <time> </time>
    <name> </name>
    <actSpeed></actSpeed>
    <actDist></actDist>
  </point>
</points>
```

Výpis 4: Demonstrace uložení bodu na trase.

Vidíme, že u každého bodu je uložena zeměpisná šířka a délka, převýšení, čas, rychlost mezi tímto a předchozím bodem a vzdálenost od začátku trasy. Důležitým faktorem u každého bodu je jeho validita, která je dána atributem u bodu. Pokud je bod nevalidní, atribut `valid` se nastaví na `false`.

Tento soubor se znovu generuje pouze, když dojde k úpravě validity bodů na trase. To je poměrně časově náročná operace, musí totiž dojít k přepočítání všech informací o trase, protože informace nejsou nadále správné. Na druhou stranu tato operace neprobíhá příliš často.

5.8 Tvorba webových stránek s informacemi o trasách

Uživatel má možnost jednoduše exportovat své trasy do HTML dokumentu. Tímto získává možnost přístupu k informacím o jednotlivých trasách bez nutnosti spouštět aplikaci a může se o svou aktivitu pak poměrně snadno například podělit s přáteli. Exportovaná webová stránka pak obsahuje všechny informace o trase, mapu, tabulku bodů a také tři grafy, které jsou pravděpodobně nejdůležitější:

- graf závislosti převýšení na uražené vzdálenosti,
- graf závislosti rychlosti na uražené vzdálenosti,
- graf závislosti převýšení na čase.

Generování webové stránky proběhne po stisku tlačítka „Generovat HTML zprávu“. Při něm se vygenerují soubory s uloženými grafy a mapou a poté proběhne generování samotné HTML stránky. Webové stránky jsou dvojího typu. První se týká všech takto vytvořených tras a obsahuje tabulku se základními údaji o trasách a odkazy na webové stránky pro jednotlivé trasy. Vzhled výsledku je demonstrován na obrázku 8.

Jméno trasy	Datum	Délka	Převýšení
Sobesovice	06.10.2010 12:46:54	13.43 km	473.19 m
Hochkar	03.03.2011 12:39:21	454.81 km	4217.65 m
Ivanovice	13.11.2010 11:08:28	15.56 km	530.61 m
Prasiva	21.11.2010 12:30:35	42.37 km	1048.81 m
Total		526.17 km	6270.26 m

Obrázek 8: Souhrn tras jako webová stránka.

Je důležité, že mapový podklad, který se při této proceduře použije je ten, který má uživatel zobrazen v době, kdy stiskne tlačítko generování HTML zprávy. Tento přístup byl zvolen proto, aby uživatel velikost a obsah této mapy mohl ovlivnit a mapa ve výsledném dokumentu tak bude podle jeho představ.

Je důležité, že výsledné HTML dokumenty splňují požadavky validity podle verze HTML 4.01 Transitional.

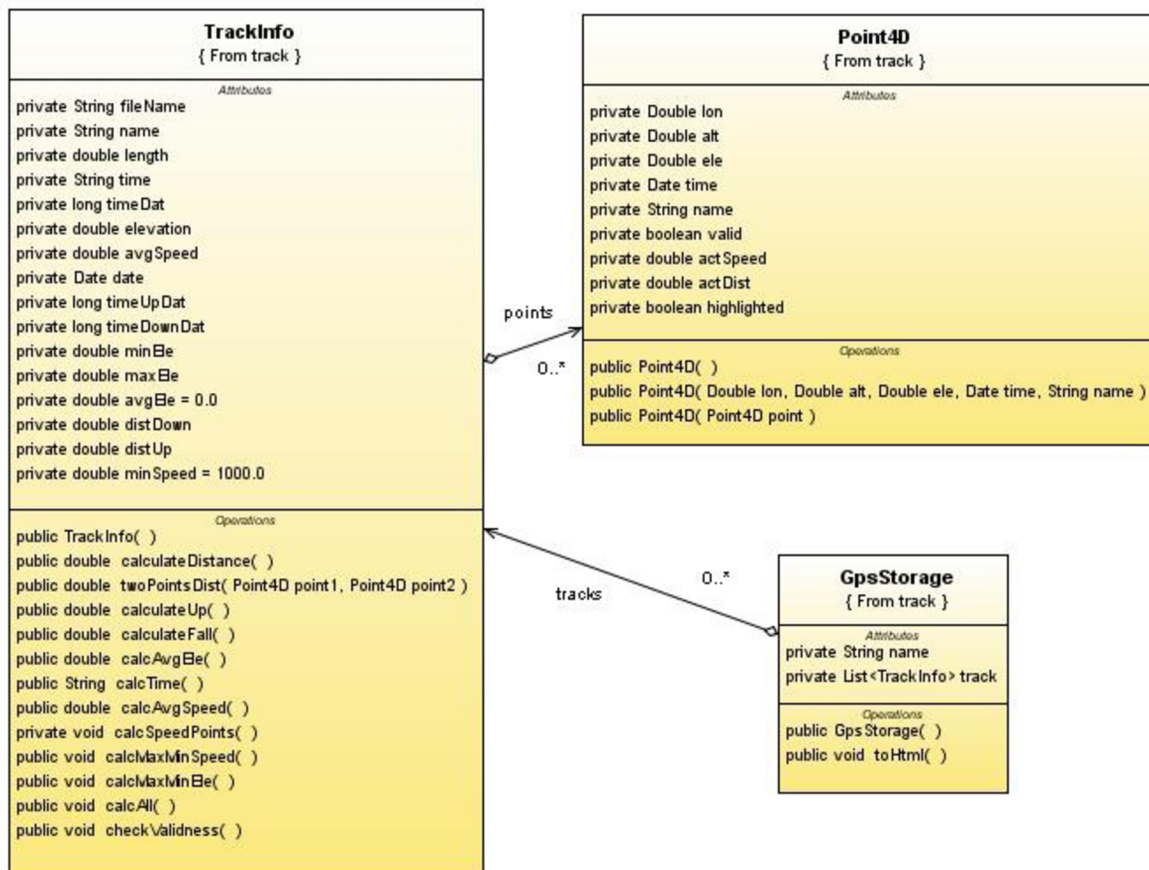
6 Architektura aplikace

Při tvorbě aplikace této velikosti, je velmi důležité dobře promyslet její architekturu. Jde hlavně o rozvržení funkčnosti do balíčků a tříd, pak také musíme zvážit provázání jednotlivých tříd a dědičnost mezi nimi. Pro navrženou aplikaci se jako optimální jeví rozdělení do čtyř balíčků:

- files pro práci se soubory,
- track pro informace o trasách,
- util pro statické metody, které jsou globálně dostupné,
- windows pro okna a celkové GUI.

6.1 Balíček track

Tento balíček je v celé aplikaci stěžejní. Obsahuje možnost uložení jednotlivých tras a bodů na nich a uchovává celkové informace o trasách. Z diagramu na obrázku 9 je patrné, že balíček obsahuje 3 třídy.



Obrázek 9: Diagram tříd balíčku track.

6.1.1 Třída `Point4D`

Tato třída reprezentuje elementární jednotku trasy, přesněji jeden bod. Třída umožňuje uložení významných informací o bodu, a to zeměpisnou šířku, zeměpisnou délku, nadmořskou výšku, jméno, čas záznamu bodu, predikát validity bodu, rychlost pohybu mezi daným bodem a předchozím bodem, vzdálenost od počátku trasy a predikát zvýraznění bodu.

Prvních pět informací k bodu se načte ze vstupního souboru. Výpočet rychlosti a vzdálenosti proběhne později během načítání celé trasy. Validitu a zvýraznění bodu ovlivňuje uživatel za běhu programu, implicitně je bod validní a bez zvýraznění.

6.1.2 Třída `TrackInfo`

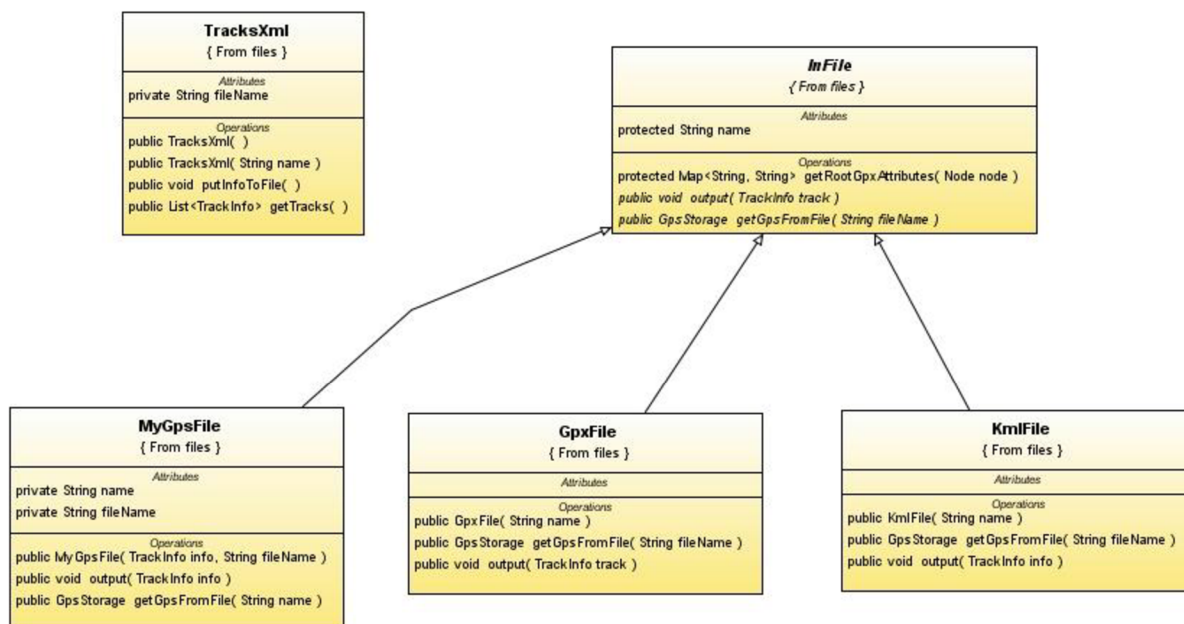
Tato třída reprezentuje jednu celou trasu. Je v ní uložen seznam bodů a také souhrnné údaje o trase (viz sekce 5.5). Mezi jejími metodami jsou pak metody pro provedení výpočtů na trase, tedy metody pro výpočet vzdáleností, rychlostí, převýšení, času a podobně. Dále tato třída také umožňuje uložení jména trasy, jména souboru s informacemi o trase, jména příslušné HTML stránky, pokud byla pro tuto trasu generována, a barvy trasy v grafu a mapě.

6.1.3 Třída `GpsStorage`

Tato třída reprezentuje množinu tras. Používá se při načítání tras ze vstupních souborů, protože kromě tras je vhodné načíst rovněž atributy týkající se trasy v souboru formátů KML nebo GPX. Další funkcí této třídy je export tras do HTML dokumentu, protože třída obsahuje metody pro tvorbu HTML dokumentů pro jednotlivé trasy i pro celkový souhrn všech tras.

6.2 Balíček `files`

Tento balíček obsahuje třídy pro práci s různými typy souborů. Jde konkrétně o soubory KML a GPX a pak také vnitřní typy souborů, jak je uvedeno v sekci 5.7. Diagram tříd k tomuto balíčku ukazuje obrázek 10.



Obrázek 10: Diagram tříd balíčku files.

Z diagramu je patrné, že byla použita dědičnost pro zobecnění práce se všemi soubory, které obsahují informace o trase, tedy soubor GPX (třída `GpxFile`), soubor KML (třída `KmlFile`) a také interní soubor (třída `MyGpsFile`). Všechny tyto třídy totiž dědí od abstraktní třídy `InFile`, obsahující metodu pro extrakci atributů ze souborů a abstraktní metody pro vstup a výstup informací o trase. Tyto metody se pak v jednotlivých zděděných třídách liší podle struktury XML souboru, který je v daném typu obsahu obsažen. Aplikace je navržena tak, aby nebyl problém kdykoli v budoucnu přidat další podporovaný formát souboru bez nutnosti měnit velké úseky kódu. Koncept dědičnosti je zde použit kvůli zobecnění práce s těmito soubory, takže když budeme chtít přidat další typ souboru, stačí přidat třídu, která bude dědit od abstraktní třídy `InFile`.

Samostatnou jednotku pak tvoří třída `TracksXml` starající se o uložení souhrnných informací o všech trasách, její účel je tedy jiný, než u ostatních tříd a proto není zděděna od třídy `InFile`. Tato třída rovněž obsahuje metody pro vstup (`getTracks()`) a výstup (`putInfoToFile()`). Tato třída pracuje vždy se seznamem tras, takže umožňuje vstup i výstup se všemi trasami.

6.3 Balíček util

Tento balíček má v navržené aplikaci speciální roli. Obsahuje pouze jednu třídu `Util`. Tato třída obsahuje pouze statické metody, které jsou použitelné pro různé třídy v celé aplikaci. Metody této třídy se týkají většinou práce s XML soubory, jako je otevření a zpracování do třídy `Document` ze standardní třídy API `Javax.org.w3c.dom.Document`. Jsou zde rovněž metody pro export XML a HTML dokumentů. Další skupina metod slouží pro zpracování datumu. Provádějí převody z řetězců na objekty třídy `Date` a naopak, umožňují výpočet doby trvání a podobně. Metody této třídy ukazuje obrázek 11.

Util
<i>Attributes</i>
<i>Operations</i>
<code>public FileInputStream getFileInputStrem(String fn)</code>
<code>public Document getXMLDocument(String fn)</code>
<code>public Date getDateFromString(String strDate)</code>
<code>public String outTime(Date date)</code>
<code>public String getStringFromXML(Document document)</code>
<code>public String getStringTree(Node node, String str)</code>
<code>private String printElement(Element node, String str)</code>
<code>private String printElement(Text node, String str)</code>
<code>public void displayTree(Document document)</code>
<code>public void printNodeHierarchy(Node node)</code>
<code>public String printNodeType(Node node)</code>
<code>public void printNode(Node node)</code>
<code>public void viewRunningTime(Date startDate, Date endDate)</code>
<code>public void writeXMLIntoFile(Document document)</code>
<code>public String addSpaces(int num)</code>
<code>public String getTimeFromLong(long milsecs)</code>
<code>public double calcSpeed(double length, double time)</code>
<code>public long timeFromString(String time)</code>
<code>public Date dateStringToDate(String strDate)</code>
<code>public double myRound(double rawNum, int num)</code>

Obrázek 11: Metody třídy `Util`.

6.4 Balíček `windows`

Tento balíček pokrývá celou grafiku aplikace. Obsahuje tyto třídy:

- `Alert` – okno upozornění sloužící k hlášení podstatných skutečností uživateli.
- `AllTracksDB` – tabulka obsahující podrobnější informace o starších trasách.
- `ColorChooserMy` – volba barvy (viz obrázek 6).
- `GrFrame` – okno pro velký graf (viz obrázek 7).
- `Graph` – provádí vykreslování grafu.
- `GraphCreator` – pomocná třída pro vykreslování grafu dodávající grafová data.
- `LineDefinition` – definice jednotlivých čar v grafu.
- `Main` – hlavní okno aplikace.
- `MyTableCellRenderer` – pomocná třída pro vybarvování buněk tabulky.
- `PanelPointTable` – panel se zobrazením bodů na trase.

Dále bude proveden podrobnější rozbor některých významných tříd.

6.4.1 Třída `AllTraksDB`

Tato třída tvoří okno s tabulkou všech uložených tras. V této tabulce je uloženo vždy jméno trasy, její délka, převýšení a datum. Pod tabulkou je zobrazen součet převýšení a součet vzdáleností těchto tras.

Výsledek demonstruje obrázek 12. Kromě grafiky je tato třída velmi jednoduchá. Obsahuje jen metodu pro přidání seznamu tras, které se zobrazí v tabulce.

Jméno	Vzdálenost	Převýšení	Datum
#00266# 2010-10-07T06:48:33Z	17.4 km	1091.68 m	Thu Oct 07 06:48:33 CEST 2010
#00833# 2011-02-12T12:53:34Z	6.38 km	244.99 m	Sat Feb 12 12:53:34 CET 2011
#00001# 2010-10-14T06:53:28Z	29.53 km	822.98 m	Thu Oct 14 06:53:28 CEST 2010
dan	19.49 km	451.78 m	Fri Oct 08 08:03:28 CEST 2010
#00001# 2011-01-25T12:52:21Z	26.66 km	1073.47 m	Tue Jan 25 12:52:21 CET 2011
čtvrtek králík	46.94 km	1064.29 m	
Celkem:			
Vzdálenost: 146.39 km			
Převýšení: 4749.19 m			

Obrázek 12: Okno databáze tras.

6.4.2 Třída Graph

Třída `Graph` je pro celou práci velmi podstatná. Provádí vykreslování grafů, což znamená souřadnicový systém, popisky os a případné pomocné čáry. Toho je docíleno zděděním standardní třídy `JPanel`. Díky tomu získáme přístup k instanci třídy `Graphics`, která nám umožňuje vykreslování základních grafických entit jako čar, kruhů, obdélníků a rovněž řetězců. Metody pro vykreslování si samy poradí s rozlišením grafu na základě maximálních a minimálních hodnot, které do grafu vykreslí. Obsahuje dvě důležité metody `makeGraph()`, která vykreslí souřadný systém a metodu `addLine()`, která přidá jednu úsečku do grafu. Pak jsou zde také pomocné metody. Například `addHighlight()` přidáváající zvýraznění bodu a `addCross()` pro přidání pomocných čar k bodu při přejetí myši. Naopak zde existuje také metoda pro zjištění hodnoty v grafu, když známe souřadnice myši.

Zajímavou vlastností vytvořeného grafu je to, že do jednoho obrázku lze vykreslit údaje o více trasách. Uživatel tak získává možnost srovnávat více svých tras. Meze grafu se při této proceduře vypočítají z tras s největšími extrémy. Jednotlivé trasy se vykreslují různými barvami podle toho, jaká barva je pro konkrétní trasu zvolena.

6.4.3 Třída GraphCreator

Tato třída řídí přidávání čar do grafu. Třída `Graph` totiž přijímá úsečky definované počátečními a koncovými hodnotami. Zjišťuje také maxima a minima v požadovaných veličinách, aby mohl mít graf co nejlepší rozvržení. Provádí veškerou komunikaci s instancí třídy `Graph` a deleguje na ni klíčové požadavky. Výstupem této třídy není nic grafického, pouze řídí grafiku jiné třídy.

6.4.4 Třída LineDefinition

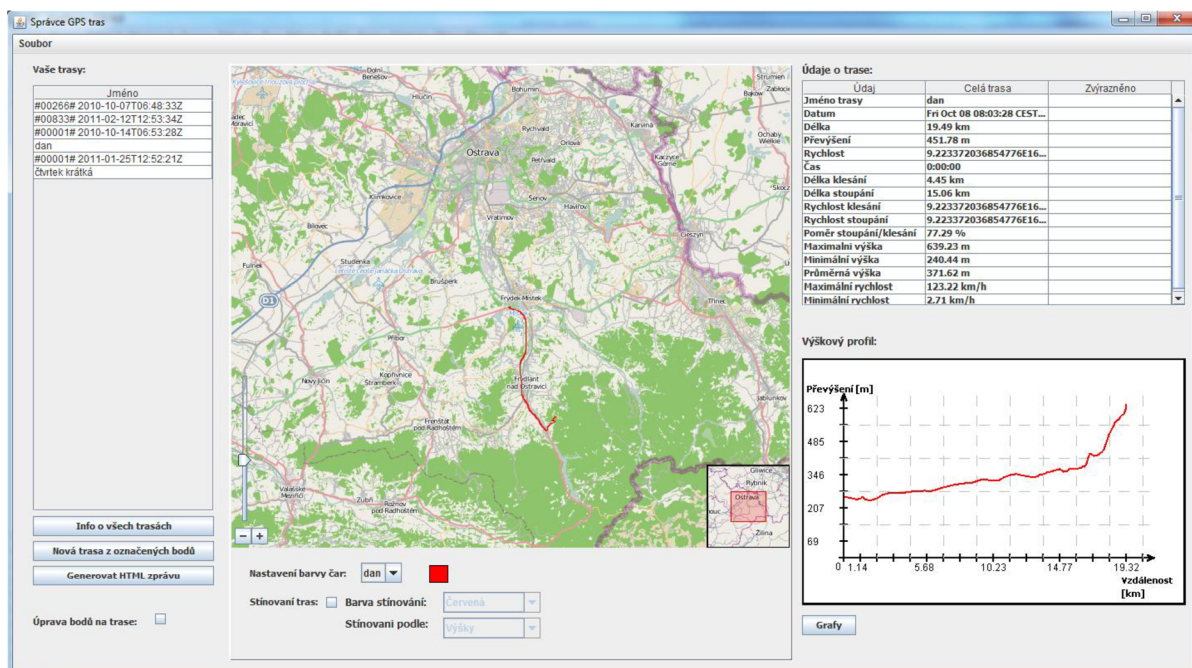
Třída `LineDefinition` je další pomocnou třídou ke grafu. Nemá opět vlastní GUI, ale slouží pouze pro uložení jednotlivých úseček reprezentujících všechny funkční čáry v grafu. Do instance této třídy jsou uloženy souřadnice počátečního bodu, souřadnice koncového bodu a dále barva, kterou se daná úsečka vykreslí. Počáteční a koncový bod jsou objekty třídy `java.awt.Point` a barva je objekt třídy `java.awt.Color`. Jednotlivé úsečky jsou pak uloženy do seznamu a následně vykresleny.

6.4.5 Třída Main

Jak už samotný název napovídá, tato třída je hlavní třídou celého programu a při spuštění se jako první vytvoří její instance. Tato třída vykresluje hlavní okno celé aplikace, reaguje na uživatelskou činnost a vytváří další objekty podle uživatelského chování a řídí běh celé aplikace. Pokud je okno vykreslené třídou `Main` zavřeno, dojde k ukončení celé aplikace.

Tato třída po vykreslení hlavního okna načte databázi starších tras a zobrazí mapu a informace k trase, se kterou se naposledy pracovalo, tedy zobrazí její mapu, údaje o trase a miniaturu grafu převýšení.

Výsledek je ukázán na obrázku 13.



Obrázek 13: Výsledný vzhled aplikace.

Tato třída obsahuje, na rozdíl od ostatních, klasické menu v horní liště. To neobsahuje mnoho položek, aby uživatel nebyl zmaten a rychle se zorientoval. Pro rychlejší přístup k funkcím je každé volbě přiřazena klávesová zkratka. Položky menu jsou tyto:

- Přidat trasu – přidá novou trasu ve formátu GPX nebo KML do databáze tras. Klávesová zkratka `Ctrl+N`.
- Export do souboru GPX – klávesová zkratka `Ctrl+G`.
- Export do souboru KML – klávesová zkratka `Ctrl+K`.
- Konec aplikace – klávesová zkratka `Ctrl+Q`.

6.4.6 Třída PanelPointTable

Poslední třída z tohoto balíčku slouží pro zobrazení a úpravu bodů na trase. Volba zobrazení této tabulky se skrývá pod zatržítkem „Úprava bodů na trase“ a tabulka se zobrazí místo panelu pro volby mapy. To umožní uživateli sledovat změny v tabulce na mapě, v informacích o trase a v grafu. Když uživatel znovu klepne na zatržítko, tento panel zmizí a zobrazí se znovu panel nastavení mapy. Vzhled tabulky pro úpravu bodů je na obrázku 14.

Zeměpisná šířka	Zeměpisná délka	Výška [m]	Čas	Rychlost [km/h]	Validita
49° 40' 40.3"	18° 20' 20.03"	266,65	8:03:28	0	false
49° 40' 40.34"	18° 19' 19.97"	252,61	8:03:43	25,39	true
49° 40' 40.29"	18° 19' 19.94"	254,42	8:03:58	24,79	true
49° 40' 40.3"	18° 19' 19.87"	253,65	8:04:13	19,16	true
49° 40' 40.3"	18° 19' 19.84"	252,77	8:04:28	9,89	true
49° 40' 40.3"	18° 19' 19.84"	251,63	8:04:43	0,45	false

Zobrazení souřadnic:

Počet nevalidních bodů: 17

Obrázek 14: Vzhled tabulky pro úpravu bodů.

Vidíme, že panel obsahuje tabulku se všemi body na trase. U každého bodu jsou jeho souřadnice, tedy zeměpisná šířka a zeměpisná délka, dále jeho výška, čas pořízení bodu, rychlost v daném bodě a nakonec barevné políčko, které určuje, zdali je bod platný či nikoli (viz sekce Validace bodů). Je-li bod validní, je políčko zelené a v opačném případě je červené. Sloupec s políčky platnosti je tak jediným, který uživatel může editovat, při kliknutí myši do buňky s validitou je validita bodu změněna a barva se změní. Uživatelské změny validity se projeví po stisknutí tlačítka „Aktualizovat tabulku“. Tlačítko „Provést automatickou detekci chybných bodů“ provede automatickou validaci bodů. Tlačítko „Reset validity“ nastaví všechny body na trase na validní a tlačítko „Zvýraznit body“ zvýrazní označené body v grafech a mapě. Součástí tohoto panelu je také informace o celkovém počtu nevalidních bodů na trase.

Je možno si změnit typ zobrazení souřadnic. To je buď kompletní, tedy $SS^{\circ} MM' VV, VV''$, ve formátu $SS^{\circ} MM, MMM'$, $SS^{\circ} MM, VV'$, kde S je stupeň, M je minuta a V je vteřina. Poslední možností zobrazení je desetinné číslo.

6.5 Implementace aplikace

V této kapitole bych chtěl zmínit několik zajímavých implementačních řešení, které jsem použil při své práci.

6.5.1 Vlákna

Aplikace používá při jistých příležitostech více vláken, hlavně z důvodu zpracování dlouhých tras. Když byla totiž aplikace původně jednovláknová, trvalo načtení a přidání trasy o délce asi 500 kilometrů kolem 10 minut. Muselo totiž dojít k extrakci všech bodů ze souboru a hlavně se musely nad trasou provést všechny výpočty, které jsou časově poměrně náročné. Po celou dobu načítání vstupního souboru se s aplikací nedalo pracovat, protože nereagovala na vstupní pokyny. Proto bylo zvoleno řešení, které při načítání nového souboru spustí vlákno a po ukončení své činnosti dá uživateli informační hláškou vědět, že je trasa načtena a připravena k práci.

Druhé vlákno bylo zvoleno pro situaci úpravy validity jednotlivých bodů na trase. To opět z důvodu časové náročnosti. Když je totiž validita bodů změněna, stávají se všechny údaje o trase neplatnými a je nutno provést nový výpočet všech informací o trase. Při změně validity je tedy spuštěn tento typ vlákna, který znovu vypočte informace o trase, aktualizuje tabulky s těmito informacemi, graf a mapu.

6.5.2 Automatická validace bodů

Algoritmus validace má velmi jednoduchý princip. Probíhá v cyklu, který takto projde všechny body na trase. Poté proběhne srovnání vzdálenosti, rychlosti a převýšení s průměrnou hodnotou této veličiny v okolních bodech. Počet okolních bodů, které se berou v úvahu je dán konstantou `FW_SIZE`, čili je možno jej bez větších problémů změnit. Po načtení jednotlivých bodů do pole proběhne výpočet průměrné rychlosti, vzdálenosti a nadmořské výšky v okolí daného bodu. Pak se provede porovnání s hodnotami v aktuálním bodě. Míra srovnání je dána konstantami `SP_MULTIPLIER`, `DIST_MULTIPLIER` a `ELE_MULTIPLIER`. Validace tedy probíhá na základě podmínek jako: „pokud není aktuální hodnota veličiny v daném bodě v intervalu (veličina / `X_MULTIPLIER` - veličina * `X_MULTIPLIER`)“, kde `X` je jedna z veličin, je bod označen jako nevalidní. Je tedy vidět, že proces validace není nikterak složitý a přitom dosahuje dobrých výsledků při správném nastavení konstant `FW_SIZE`, `SP_MULTIPLIER`, `DIST_MULTIPLIER` a `ELE_MULTIPLIER`. Toto nastavení bylo provedeno po dlouhém testování různých hodnot a dosahuje rozumných výsledků.

6.5.3 Vykreslení trasy do mapy

Pro vykreslení trasy do mapy bylo nutno přetížit metodu `paintWaypoint()` ve třídě `JXMapKit`. Původní metoda umožňovala pouze zvýraznění bodů v mapě, což bylo nevyhovující. Pro vykreslení bodů do mapy se používá `Set`, tedy neuspořádaná kolekce ze třídy `java.util`. Tato neuspořádanost ale působila problémy, protože spojnice mezi body v mapě se vykreslovaly poměrně náhodně a nikoli mezi dvěma sousedními body, proto bylo zapotřebí jejich uspořádání. To bylo nejprve testováno za pomoci třídy `LinkedHashSet`, ale ani toto řešení nebylo vhodné, protože v určitých situacích stále docházelo k propojování bodů, které spolu nesouvisely. Nakonec byl problém vyřešen tak, že metoda pro vykreslování bodů přistupuje k seznamu bodů v aktuálně zvolené trase a podle tohoto seznamu bodů určuje posloupnost propojení bodů v mapě. Zároveň je zde omezena maximální možná vzdálenost mezi sousedními body, aby nedošlo k propojení bodů z více spolu nesouvisajících tras, které jsou aktuálně vykresleny v mapě.

7 Závěr

Cílem této práce bylo vytvoření komplexního a zároveň jednoduchého programu pro správu geografických dat získaných pomocí GPS, který by dobře sloužil všem cestovatelům. Umožňuje jim vytvořit si databázi svých absolvovaných výletů, analyzovat své výsledky a srovnávat jednotlivé trasy. Program je jednoduchý a jeho použití je snadné i pro člověka, který v informační technice není příliš zblhlý. K jednoduchosti přispívá intuitivní ovládání a rozvržení ovládacích prvků v aplikaci. Díky uložení dat ve formátu XML nepotřebuje uživatel kromě běhového prostředí Javy žádný další nástroj, což rovněž přispívá k dobré použitelnosti a jednoduchosti aplikace. Myslím, že můj prvotní cíl, tedy použitelnost programu širokou veřejností jsem splnil.

V průběhu tvorby aplikace mě napadlo několik možných rozšíření do budoucna. Na jednu stranu jde o rozšíření drobná, jako je například možnost volby více mapových podkladů a více formátů vstupních souborů, či možnost exportu dat do jiných formátů než HTML, například do formátu PDF (*Portable Document Format*). Na stranu druhou mě napadla také rozšíření, která by celou aplikaci posunula do trochu jiné sféry. Tady připadá v úvahu možnost přidání záznamu o srdeční činnosti v průběhu trasy. Takto by vznikl vhodný tréninkový program pro všechny sportovce, kteří by se tak mohli poučit ze svých chyb při tréninku a mohli by se jich příště vyvarovat.

Celá aplikace vznikala tak, aby bylo možné ji do budoucna co nejjednodušeji rozšiřovat díky použití objektově orientovaného jazyka Java. Kód je vhodně hierarchicky členěn do balíčků a tříd a tím umožňuje snadnou aktualizaci. K celému kódu je též dostupný `javadoc`, takže je poměrně jednoduché se v něm orientovat.

Díky této práci jsem získal lepší orientaci v možnostech a vlastnostech jazyka Java. Zdokonalil jsem se hlavně v tvorbě grafického uživatelského rozhraní v tomto jazyce a v obsluze uživatelských vstupů. Novinkou pro mě byla též samostatná práce na takto velké aplikaci, protože zdrojový kód má celkem kolem šesti tisíc řádků. Kvůli tomu byl nutný opravdu dobrý a podrobný návrh, který mi práci usnadnil. Jako nadšený sportovec a cestovatel budu tuto aplikaci používat a snad i vylepšovat. Skutečnost, že patřím do skupiny lidí, pro které je tato aplikace určena, mi velmi pomohla při její tvorbě.

Citace

- [1] BARTSCH, H. J.: *Matematické vzorce*. Mladá fronta, Praha, 1996. ISBN 80-204-0607-8.
- [2] HOJGR R., STANKOVIČ J. *GPS - Praktická uživatelská příručka*. Computer Press, Brno, 2007. ISBN: 978-80-251-1734-7.
- [3] JURKINA M. I., PICK M.: *Numerické výpočty ve světovém geodetickém referenčním systému 1984 (WGS84)*. Vojenský geografický obzor, 2006, č. 1, Příloha 2, Geofyzikální ústav ČAV, Praha.
- [4] [WWW.GOOGLE.COM](http://code.google.com/intl/cs/apis/kml/documentation/kmlreference.html): *KML Reference* [online] 2011. [cit. 2011-23-01]. Dostupné na URL: <<http://code.google.com/intl/cs/apis/kml/documentation/kmlreference.html>>.
- [5] Rapant, P.: *Družicové polohové systémy*. VŠB-TU Ostrava, 2002. ISBN 80-248-0124-8.
- [6] [WWW.TOPOGRAFIX.COM](http://www.topografix.com/gpx.asp): *GPX: the GPS Exchange Format* [online]. Poslední modifikace: 19. července 2007. [cit. 2011-23-01]. Dostupné na URL: <<http://www.topografix.com/gpx.asp>>.
- [7] Kosek, J.: *XML pro každého*, Grada publishing, Praha, 2000, ISBN 80-7169-860-1.
- [8] Keogh, J.: *Java bez předchozích znalostí*, CP Books a.s, Brno, 2005. ISBN 80-251-0839-2.
- [9] [WWW.TIOBE.COM](http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html): *TIOBE Programming Community Index for April 2011* [online]. 2011. [cit. 2011-11-04]. Dostupné na URL: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>.
- [10] Castro, E.: *HTML, XHTML a CSS*. Computer Press, Brno, 2007. ISBN: 978-80-251-1531-2.
- [11] [INSTRUCT.UWO.CA](http://instruct.uwo.ca/earth-sci/505/utms.htm): *Conversion of Latitude* [online]. [cit. 2011-28-04]. Dostupné na URL: <<http://instruct.uwo.ca/earth-sci/505/utms.htm>>.
- [12] [WIKIPEDIA.ORG](http://en.wikipedia.org/wiki/Galileo_(satellite_navigation)): *Galileo (satellite navigation)* [online]. Poslední modifikace: 30. dubna 2011. [cit. 2011-02-05]. Dostupné na URL: <[http://en.wikipedia.org/wiki/Galileo_\(satellite_navigation\)](http://en.wikipedia.org/wiki/Galileo_(satellite_navigation))>.
- [13] [WWW.JARVANA.COM](http://www.jarvana.com/jarvana/view/org/swinglabs/swingx-ws/1.0/swingx-ws-1.0.jar!/org/jdesktop/swingx/JXMapKit.class?classDetails=ok): *Class Details - org.desktop.swingx.JXMapKit* [online]. 2011. [cit. 2011-11-05]. Dostupné na URL: <<http://www.jarvana.com/jarvana/view/org/swinglabs/swingx-ws/1.0/swingx-ws-1.0.jar!/org/jdesktop/swingx/JXMapKit.class?classDetails=ok>>

8 Přílohy

8.1 Obsah příloženého CD

- Zdrojové kódy aplikace.
- Spustitelný soubor.
- Testovací trasy.
- Externí knihovny potřebné pro běh aplikace.
- Technická zpráva ve formátu pdf.
- Technická zpráva ve formátu doc.
- Soubor s popisem překladu a spuštění aplikace.
- Dokumentace ve formátu javadoc.

8.2 Příručka pro používání programu

Po spuštění aplikace se zobrazí hlavní okno (viz Obrázek 13). Po spuštění se zobrazí informace o poslední trase, se kterou uživatel pracoval. Pro volbu jiné trasy stačí zvolit jednu nebo více tras z tabulky s databází tras a potvrdit stiskem klávesy Enter. Stejným způsobem, ale po stisku klávesy Delete, dojde k odstranění vybraných tras z databáze. Změna jména trasy proběhne při změně názvu v tabulce a potvrzení klávesou Enter. Přidání nové ve formátu KML nebo GPX trasy probíhá za pomoci položky v menu. Vzhled položek v menu ukazuje obrázek 15.

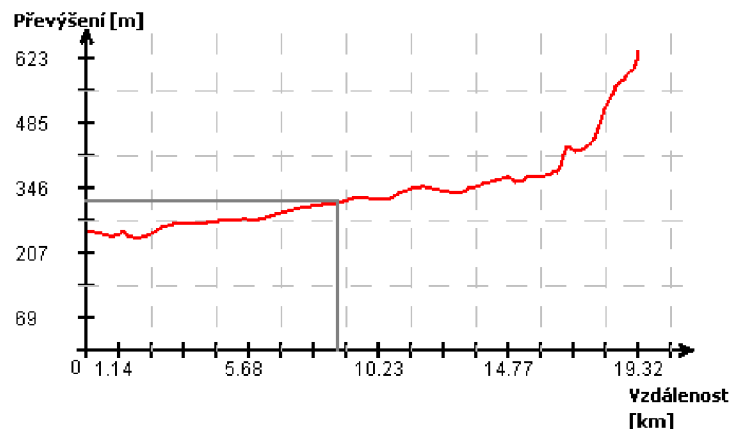


Obrázek 15: Menu aplikace.

Další položky menu slouží pro export trasy do formátu KML a GPX a pro ukončení aplikace.

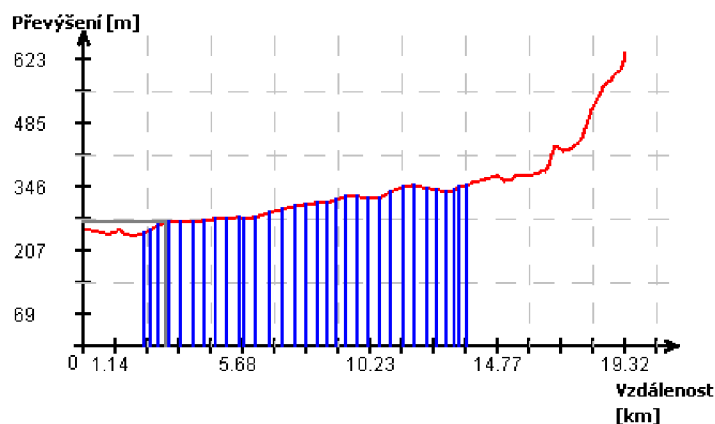
8.2.1 Ovládání grafu

Při přejetí kurzorem myši nad grafem se zobrazí pomocné čáry, které pomohou určit aktuální hodnotu na obou osách, což demonstruje obrázek 16.



Obrázek 16: Pomocné čáry v grafu.

Výběr určité oblasti v grafu proběhne po dvojném stisknutí levého tlačítka myši. Po prvním stisknutí se do grafu přidá modrá čára, která označuje bod, od kterého probíhá výběr. Po druhém stisknutí tlačítka myši se oblast vybere, vyznačí se v grafu a mapě a provedou se výpočty nad novou trasou a zobrazí se do tabulky s údaji o trase. Při dalším klepnutí myši na obrázek se provede výběr jiného úseku. To ukazuje obrázek 17.



Obrázek 17: Demonstrace výběru úseku trasy.

Druhý způsob výběru určitých bodů je pomocí tabulky bodů na trase (viz obrázek 14). Zde stačí označit jeden nebo více bodů a klepnout na tlačítko „Zvýraznit body“. Opět dojde k jejich zvýraznění v grafech a obrázek 18 ukazuje zobrazení zvýrazněných bodů v mapě.



Obrázek 18: Zvýraznění bodů v mapě.