

Slovenská Technická Univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

FIIT-13428-20863

Bc. František Kostroš

Program pre spracovanie NMEA viet

Diplomová práca

Študijný program: Počítačové a komunikačné systémy a siete
Študijný odbor: 9.2.4 Počítačové inžinierstvo
Miesto vypracovania: Ústav počítačových systémov a sietí, FIIT STU Bratislava
Vedúci práce: Ing. Peter Trúchly, PhD.

máj, 2010

Čestné prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne, na základe získaných vedomostí a s použitím uvedenej literatúry.

Bratislava, máj 2010

.....
Bc. František Kostroš

PodĎakovanie

Ďakujem môjmu odbornému vedúcemu, Ing. Petrovi Trúchlemu, PhD., za odborné vedenie, trpezlivosť pri konzultáciách, cenné rady a usmernenia a v neposlednom rade aj požičanie GPS zariadenia. Toto všetko výraznou mierou prispelo k skvalitneniu tejto práce aj výsledného softvérového produktu.

Anotácia

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Študijný program: Počítačové a komunikačné systémy a siete

Autor: Bc. František Kostroš

Diplomová práca: Program pre spracovanie NMEA viet

Vedúci diplomovej práce: Ing. Peter Trúchly, PhD.

máj, 2010

Štandard NMEA definuje elektrické rozhranie a údajový protokol na komunikáciu elektronických zariadení používaných v námorníctve. Údaje prenášané pomocou tohto protokolu môžu obsahovať informácie o pozícií, nadmorskej výške, rýchlosti a iných faktoroch. Počítače sú v súčasnosti ideálnym nástrojom na zber a spracovanie takýchto údajov.

V prvej časti predkladanej práce analyzujeme samotný štandard NMEA (kapitola 2.2), ako aj existujúce softvérové prostriedky určené na analýzu viet tohto štandardu na počítači (2.3 až 2.5). V druhej časti potom špecifikujeme požiadavky na túto aplikáciu (3.1) a vytvárame architektonický návrh (3.2), ktorý následne implementujeme (3.3). Implementovaný program dokáže analyzovať vety vysielané GPS zariadeniami typu GGA, GSA, GSV, RMC a GLL. Získané údaje vie zobrazíť vo forme textov, grafov, štatistík a trasy v mape.

Annotation

Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Degree course: Computer and Communication Systems and Networks

Author: Bc. František Kostroš

Master's Thesis: Program for NMEA sentences processing

Supervisor: Ing. Peter Trúchly, PhD.

2010, May

The NMEA standard defines an electrical interface and data protocol for communication between marine electronic devices. Data transmitted over this protocol can include information on position, altitude, speed and other variables. Computers are ideal tool for collecting and processing these data.

The first part of this thesis contains analysis of the NMEA standard (chapter 2.2) and analysis of existing NMEA-compatible software tools (2.3 to 2.5). The second part is then about requirements specification (3.1) and design of application architecture (3.2), which we implement afterwards (3.3). Implemented program can analyze sentences transmitted by GPS devices - specifically sentences of type GGA, GSA, GSV, RMC and GLL. Gathered data are displayed in text form, graphics, statistics and as a route in map.

Obsah

1	Úvod	1
2	Analýza	2
2.1	System GPS	2
2.2	NMEA protokol	3
2.3	Analýza existujúcich programov	8
2.3.1	GPSView	8
2.3.2	VisualGPS	9
2.3.3	NMEA data to Google Map	10
2.3.4	Ďalšie programy	11
2.4	Analýza existujúcich knižníc	12
2.4.1	NMEA Library	12
2.4.2	NMEAP	12
2.4.3	NMEA:: Library	13
2.4.4	HS GPS DLL Library	13
2.4.5	NMEA Java APIs	13
2.5	Zhrnutie analýzy	14
3	Opis riešenia	15
3.1	Špecifikácia požiadaviek	15
3.1.1	Ciele	15
3.1.2	Funkcie	15
3.2	Návrh	16
3.2.1	Vstupná vrstva	17
3.2.2	Logická vrstva	17
3.2.3	Prezentačná vrstva	18
3.3	Implementácia	21
3.3.1	Výber prostriedkov	22
3.3.2	Vstupná vrstva	23
3.3.3	Logická vrstva	24
3.3.4	Prezentačná vrstva	25
3.3.5	Pomocník	33

3.4	Testovanie	33
3.4.1	Analýza viet zo súboru	33
3.4.2	Analýza viet zo zariadenia	34
3.4.3	Simulácia	34
3.4.4	Výsledky testovania	34
4	Záver	35
	Literatúra	37
A	Technická dokumentácia	39
A.1	Hlavné okno	40
A.1.1	Hlavná trieda	40
A.1.2	Vnorené triedy	40
A.1.3	Metódy	42
A.2	Logická vrstva <i>NMEAParser</i>	44
A.2.1	Metóda <i>parse()</i>	44
A.2.2	Metóda <i>analyze()</i>	46
A.3	Pomocné triedy	47
A.3.1	<i>NMEADData</i>	47
A.3.2	<i>SatInfo</i>	47
A.3.3	<i>SerialPortTools</i>	48
A.3.4	<i>SharedData</i>	48
A.3.5	<i>ActualData</i>	48
A.3.6	<i>Tools</i>	48
A.4	GUI triedy	49
A.4.1	<i>MDIDesktopPane</i>	49
A.4.2	<i>InternalFrameProperty</i> a <i>InternalFrameState</i>	49
A.4.3	<i>WindowMenu</i>	49
A.5	Triedy mapy satelitov	50
A.5.1	<i>NMEAXYDataItem</i>	50
A.5.2	<i>NMEAXYSeries</i>	50
A.5.3	<i>NMEAItemRenderer</i>	50

B	Používateľská príručka	51
B.1	Funkcia programu	52
B.2	Hardvérová konfigurácia	52
B.3	Požiadavky na programové prostriedky	52
B.4	Inštalácia a spustenie	52
B.5	Opis prvkov GUI	53
B.5.1	Hlavná ponuka - menu	53
B.5.2	Plocha	54
B.5.3	Stavová časť	58
B.5.4	Pomocník	58
C	Obsah CD média	60

Zoznam skratiek

CR	Carrier Return
dB	decibely
DGPS	Differential GPS
DOP	Dilution of Precision
FAA	Federal Aviation Administration
GPS	Global Positioning System
HDOP	Horizontal Dilution of Precision
LF	Line Feed
MFC	Microsoft Foundation Class
MVC	Model-View-Controller
NMEA	National Marine Electronic Association
PDOP	Position Dilution of Precision
PPS	Precise Positioning Service
PRN	Pseudorandom Number
RTK	Real-Time Kinetic
SNR	Signal-to-noise Ratio
UTC	Coordinated Universal Time
VDOP	Vertical Dilution of Precision

1 Úvod

Navigácia pomocou satelitov je v súčasnosti značne rozšírená. Medzi najpoužívanéjšie systémy určené na navigáciu patrí americký systém GPS. Primárne bol tento systém určený na vojenské účely, napríklad navigáciu na mori. Jednotlivé elektronické zariadenia, ktoré sa na lodi nachádzajú, teda napríklad sonary, echolokátory, autopilot a GPS prijímače navzájom spolupracujú. Nevyhnutnou podmienkou tejto spolupráce je vzájomná komunikácia. Na zabezpečenie tejto komunikácie vytvorila Národná námorná elektronická asociácia (NMEA - National Marine Electronics Association) štandard NMEA-0183, ktorou túto komunikáciu špecifikuje po elektrickej a údajovej stránke.[6]

Pri analýze tejto problematiky sme sa najprv zamerali na systém GPS, keďže protokol NMEA sa bežne používa hlavne v GPS prijímačoch. Po opísaní všetkých dôležitých stránok tohto systému nasleduje popis protokolu NMEA od najstaršej verzie 0180 cez pôvodnú verziu 0183 až po súčasné varianty NMEA 2000. V ďalšej časti sú popísané existujúce softvérové prostriedky na analýzu viet týchto protokolov, analyzované boli open-source aj proprietárne riešenia. Pod pojmom softvérové prostriedky máme na mysli programy aj knižnice určené na analýzu protokolu NMEA.

2 Analýza

Táto kapitola je venovaná analýze systému GPS, protokolu NMEA a popise existujúcich softvérových prostriedkov, teda programov a knižníc, určených na spracovanie viet tohto protokolu.

2.1 Systém GPS

GPS je skratkou Globálneho pozičného systému (Global Positioning System), ktorý bol vyvinutý americkým ministerstvom obrany v 70. rokoch minulého storočia. Pôvodne bol tento systém vyvíjaný na vojenské účely, neskôr bol však sprístupnený verejnosti a v súčasnosti funguje pre verejnosť aj pre vojenský sektor.

Systém GPS poskytuje kdekoľvek na Zemi informácie o polohe a presnom čase. Služi veľkému počtu používateľov, preto funguje na jednosmernom - pasívnom - princípe. Údaje sú teda prenášané len smerom zo satelitu.

Tento systém tvorí 24 satelitov. Aby bolo zabezpečené celosvetové pokrytie, sú tieto satelity rozmiestnené tak, aby sa v každej zo 6 orbitálnych rovín nachádzali 4 satelity [11]. Týmto je zabezpečené, že z ktoréhokoľvek miesta na Zemi sú viditeľné minimálne 4 satelity, pokiaľ sa berie ohľad na elevačný uhol 10° . Práve viditeľnosť týchto 4 satelitov je nevyhnutne nutná na presné určenie polohy.

Systém GPS určuje polohu pomocou trilaterácie, je teda potrebná viditeľnosť minimálne troch satelitov. Na určenie nadmorskej výšky je potom potrebná aj viditeľnosť štvrtého satelitu [12]. Princíp fungovania je nasledovný: satelit pošle prijímaču signály s obsahom "som satelit X, moja pozícia je Y a táto informácia bola poslaná v čase Z". Každý satelit posiela informácie o svojej polohe aj ostatným satelitom. Prijímač porovná čas v ktorom bol signál prijatý s časom v signále a z ich rozdielu vypočíta vzdialenosť medzi prijímačom a satelitom. Po prijatí signálu z troch satelitov dokáže prijímač vyhodnotiť svoju 2D polohu, v prípade štyroch satelitov 3D polohu. Čas v prijímači nie je presný a aj malá odchýlka vynásobená rýchlosťou svetla dokáže spôsobiť obrovské chyby, preto sa poloha nedá presne

určiť iba z dvoch satelitov. Tri satelity sú absolútne minimum, väčšinou sa poloha určuje pomocou štyroch alebo viacerých satelitov.

GPS prijímač teda dostáva údaje o stave, polohe a presnosti vysielajúceho satelitu a časové údaje o vyslaní daného signálu a ich vzťah k času UTC. Z týchto údajov dokáže GPS prijímač vypočítať polohu, nadmorskú výšku, rýchlosť a ďalšie údaje, ktoré preloží do formátu NMEA viet a môže ich poslať ďalej.

2.2 NMEA protokol

Národná námorná elektronická asociácia vytvorila špecifikáciu, ktorá definuje rozhranie medzi rôznymi námornými elektronickými zariadeniami. Tento štandard je proprietárny, je možné zakúpiť ho priamo od NMEA. Väčšina z tohto štandardu bola analyzovaná reverzným inžinierstvom z verejných zdrojov a je dostupná na Internete.

Existuje niekoľko štandardov NMEA. Najstarším je NMEA-0180, ktorý spolu s nasledujúcim štandardom NMEA-0182 definoval komunikáciu v rámci navigačného systému LORAN-C. Vety v NMEA-0180 a NMEA-0182 sú však odlišné od viet súčasného štandardu NMEA-0183. Tento štandard sa ďalej delí na rôzne verzie, od verziu 1.5 až po najnovšiu verziu 4.0, vo väčšine GPS prijímačov sa však v súčasnosti používa verzia 2.0. Existuje aj nový štandard NMEA 2000, ktorý má v porovnaní s NMEA-0183 definovaný oveľa rýchlejší prenos údajov, údaje sú prenášané v kompaktnej forme a je v ňom špecifikovaná obojsmerná komunikácia viacerých zariadení cez sieť.

Údaje sú v štandarde NMEA-0183 prenášané v čitateľnej ASCII podobe, informačnou jednotkou je veta, prenosová rýchlosť je štandardom určená na 4800 bitov za sekundu. Každá veta sa začína znakom '\$', nasleduje 2-znakový identifikátor odosielateľa a 3-znakový identifikátor typu vety. Za ním sa nachádza niekoľko údajových polí oddelených čiarkami, na konci tohto poľa sa môže nachádzať kontrolný súčet a symboly ukončenia riadku CR/LF. Jedna takáto veta môže obsahovať maximálne 82 znakov [4].

Existuje niekoľko typov NMEA-0183 viet, niektorí výrobcovia si vytvárajú dodatočné vlastné typy. V nasledujúcej časti popíšeme typy viet, ktoré sú vysielané väčšinou GPS prijímačov [5].

GGA - obsahuje hodnoty polohy, času a ďalších údajov

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

\$GPGGA, $\overbrace{hhmmss.ss}^1$, $\overbrace{llll.ll}^2$, \overbrace{a}^3 , $\overbrace{yyyyy.yy}^4$, $\overbrace{a,x}^5$, \overbrace{xx}^6 , $\overbrace{,xx}^7$, $\overbrace{,xx}^8$, $\overbrace{,xx}^9$, $\overbrace{,M}^{10}$, $\overbrace{,xx}^{11}$, $\overbrace{,M}^{12}$, $\overbrace{,xx}^{13}$, $\overbrace{,xxxx}^{14}$ * \overbrace{hh}^{15}

Popis polí:

- 1) Univerzálny čas UTC, formát h - hodiny, m - minúty, s - sekundy
- 2) Zemepisná šírka
- 3) Severne (N) alebo južne (S)
- 4) Zemepisná dĺžka
- 5) Východne (E) alebo západne (W)
- 6) Ukazovateľ kvality GPS, určenie polohy:
 - 0 - je nedostupné
 - 1 - pomocou GPS
 - 2 - pomocou DGPS (Differential GPS)
 - 3 - pomocou PPS (Precise positioning service)
 - 4 - pomocou algoritmu RTK (Real-time Kinetic)
 - 5 - pomocou algoritmu Float RTK
 - 6 - metódou "dead reckoning" (odhadom)
 - 7 - ručným vkladaním
 - 8 - simulačný režim
- 7) Počet viditeľných satelitov, 00 až 12
- 8) Hodnota horizontálneho DOP (Dilution of Precision) v metroch
- 9) Výška antény nad/pod hladinou mora
- 10) Jednotka výšky antény - metre
- 11) Rozdiel medzi WGS-84 elipsoidom a úrovňou hladiny mora v geoide
- 12) Jednotka rozdielu - metre
- 13) Vek údajov z DGPS, čas v sekundách od poslednej aktualizácie SC104 typu
1 alebo 9, nulová hodnota značí, že sa DGPS nepoužíva
- 14) Identifikačné číslo stanice DGPS, 0000 až 1023
- 15) Kontrolný súčet

- 7) Hodnota SNR v dB (00 až 99)
 ... (údaje o ďalších satelitoch, ako v bodoch 4 až 7)
 n) Kontrolný súčet

RMC - odporúčané minimálne navigačné informácie

1 2 3 4 5 6 7 8 9 10 11 12 13
 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
 \$GPRMC,hhmmss.ss,A,lll.ll,a,yyyy.yy,a,x.x,x.x,xxxx,x.x,a,m*hh

Popis polí:

- 1) Univerzálny čas UTC
- 2) Stav A - údaje sú platné, V - údaje sú neplatné
- 3) Zemepisná šírka
- 4) Severne (N) alebo južne (S)
- 5) Zemepisná dĺžka
- 6) Východne (E) alebo západne (W)
- 7) Priemerná rýchlosť v uzloch, tzv. SMG (Speed Made Good)
- 8) Skutočná dráha v stupňoch, tzv. TMG (Track Made Good)
- 9) Dátum, formát ddmmyy kde d - deň, m - mesiac, y - rok
- 10) Magnetická odchýlka v stupňoch
- 11) Východne (E) alebo západne (W)
- 12) Ukazovateľ režimu FAA
- 13) Kontrolný súčet

GLL - zemepisná šírka a dĺžka

\$GPGLL, \overset{1}{lll}.\overset{2}{ll}, \overset{3}{a}, \overset{4}{yyyyy}.\overset{5}{yy}, \overset{6}{a}, \overset{7}{hmmss}.\overset{8}{ss}, \overset{9}{a}, \overset{10}{m} * \overset{11}{hh}

Popis polí:

- 1) Zemepisná šírka
- 2) Severne (N) alebo južne (S)
- 3) Zemepisná dĺžka
- 4) Východne (E) alebo západne (W)
- 5) Univerzálny čas UTC, formát h - hodiny, m - minúty, s - sekundy
- 6) Stav A - údaje sú platné, V - údaje sú neplatné
- 7) Ukazovateľ režimu FAA
- 8) Kontrolný súčet

VTG - údaje o rýchlosti a dráhe

\$GPVTG, \overset{1}{x.x}, \overset{2}{T}, \overset{3}{x.x}, \overset{4}{M}, \overset{5}{x.x}, \overset{6}{N}, \overset{7}{x.x}, \overset{8}{K}, \overset{9}{m}, * \overset{10}{hh}

Popis polí:

- 1) Dráha v stupňoch
- 2) T = skutočná
- 3) Dráha v stupňoch
- 4) M = magnetická
- 5) Priemerná rýchlosť v uzloch
- 6) N = uzly
- 7) Priemerná rýchlosť v km/h
- 8) K = kilometre za hodinu
- 9) Ukazovateľ režimu FAA
- 10) Kontrolný súčet

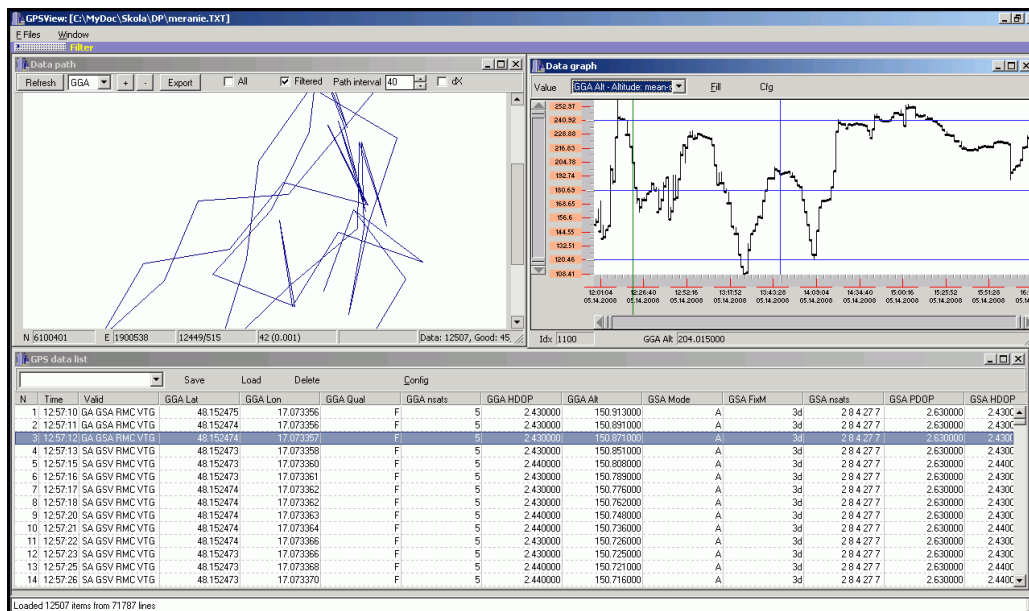
2.3 Analýza existujúcich programov

Na analýzu viet štandardu NMEA existuje niekoľko programov, či už sa jedná o open-source alebo komerčný softvér. Niektoré z týchto programov iba zobrazujú informácie získané spracovaním NMEA viet, iné programy dokážu tieto údaje interpretovať, či už ako krivky v 2D/3D priestore, alebo priamo ako krivky v mapách alebo animácie. Najznámejšie z týchto programov sme popísali v tejto podkapitole.

2.3.1 GPSView

Program GPSView je nástrojom na analýzu NMEA záznamov. Je naprogramovaný v jazyku C++, na spracovanie NMEA viet využíva vlastné procedúry. Program je poskytovaný aj so zdrojovými kódmi, celý projekt je spravovaný pomocou servera Google Code. Testovali sme verziu 140208, keďže však dokumentácia k tomuto programu neexistuje, jeho vlastnosti sme boli schopní overiť len metódou skúšania.

Program dokáže načítať NMEA vety z textového súboru, nedokáže pracovať so



Obr. 1: Okno programu GPSView

živým vstupom. Po načítaní viet dokáže zistené údaje zobrazíť vo forme grafu

(napr. nadmorská výška, počet satelitov, priemerná rýchlosť a pod.), kriviek v 2D pohľade - používajú sa údaje o zemepisnej šírke a dĺžke, alebo vo forme tabuľky, kde sú získané údaje zoradené podľa času prijatia a existuje tu možnosť výberu zobrazovaných údajov a tiež možnosť filtrovania.

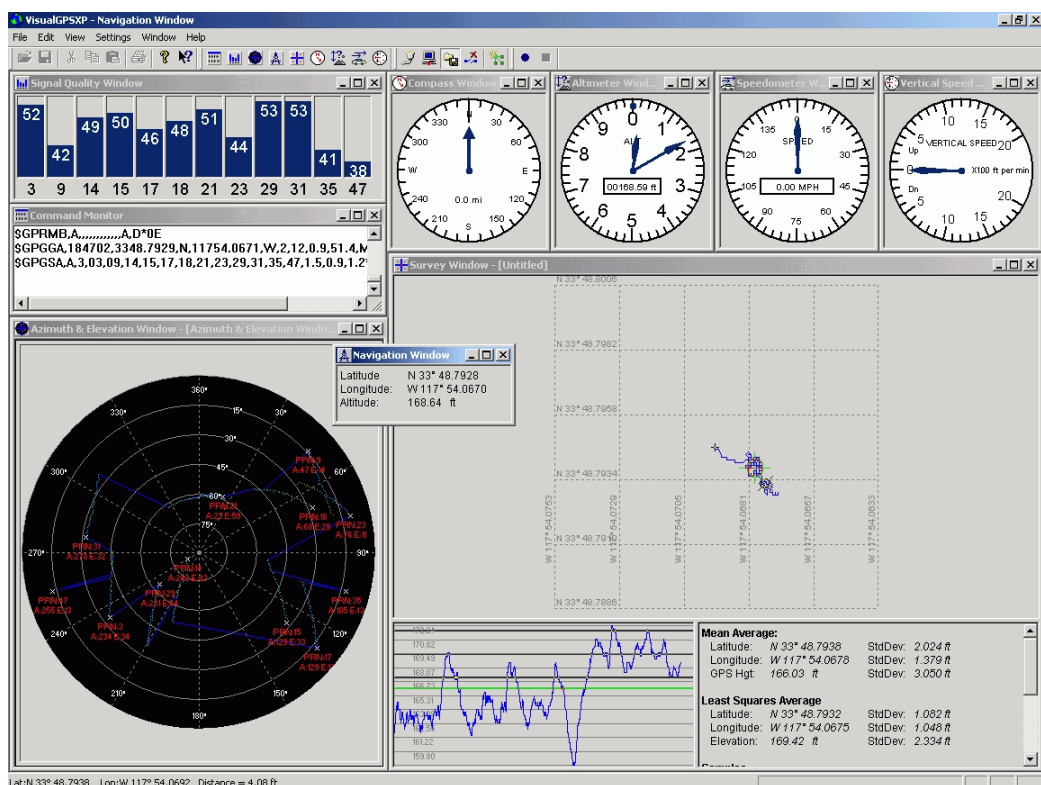
Vzhľadom program pôsobí priemerne, bez používateľskej príručky však vyžaduje značnú dávku trpezlivosti a inteligencie na objavenie všetkých jeho možností. Chýba mu možnosť načítania máp, zobrazené krivky teda nemajú veľkú výpovednú hodnotu.

2.3.2 VisualGPS

Tento program je šírený ako freeware, zdrojové kódy teda k nemu nie sú dostupné. Na stránke výrobcu existuje aj jeho pokročilejší variant s názvom VisualGPS XP. Oba sú naprogramované v jazyku C++ s použitím grafickej knižnice MFC, spracovanie NMEA viet je zrejme zabezpečené vlastnými procedúrami. Testovali sme verziu 3.2 a porovnávali sme ju s pokročilejším variantom XP.

Program na rozdiel od programu GPSView obsahuje dokumentáciu, dokáže pracovať rovnako so živým vstupom aj so súborom. Po otvorení súboru sa údaje zo záznamu prehrávajú v poradí, dá sa zvoliť rýchlosť akou sa majú prehrávať. Program obsahuje niekoľko okien zobrazujúcich rôzne druhy informácií. V okne Survey sa zobrazuje krivka trasy kreslená podľa zemepisnej šírky a dĺžky, v tomto okne sa tiež zobrazuje graf nadmorskej výšky a tabuľka hodnôt jednotlivých údajov. Ďalej je tu okno Signal Quality zobrazujúce kvalitu signálu z jednotlivých viditeľných satelitov. Je tu tiež okno Azimuth & Elevation, ktoré zobrazuje rozmiestnenie satelitov na oblohe. Verzia XP obsahuje tiež kompas, výškomer a ukazovateľ horizontálnej aj vertikálnej rýchlosti.

Vzhľad programu je jednoduchý, ovládanie je intuitívne, používateľ má možnosť nastudovať ho v priloženej príručke. Oproti programu GPSView chýba možnosť vykresliť z načítaných viet grafy, chýba tiež možnosť načítania máp. Ďalšou nevýhodou je iba postupné načítavanie údajov zo záznamu, v prípade zobrazenia grafu celej



Obr. 2: Okno programu VisualGPS XP

cesty je teda potrebné počkať, kým program nespracuje všetky vety v zázname.

2.3.3 NMEA data to Google Map

Ide o webovú aplikáciu naprogramovanú v jazyku Javascript, testovali sme verziu 4.0. Na jej korektný beh je potrebný prístup k sieti Internet. Princíp fungovania je nasledovný [8]:

1. Používateľ vloží do určeného textového poľa NMEA vety a klikne na tlačidlo “Filter data”.
2. Aplikácia odfiltruje z vložených NMEA viet vety typu GPGGA z ktorých zistí čas, zemepisnú šírku a dĺžku.
3. Používateľ klikne na tlačidlo “Generate Map”, následne aplikácia zobrazí mapu podľa zistených koordinátov. Mapy sa zobrazujú pomocou služby Google Maps.

4. Používateľ klikne na tlačidlo “Animate Route”, aplikácia začne animovať priebeh zaznamenatej cesty podľa časových údajov, väčšinou teda každú sekundu jeden krok. Rýchlosť animácie sa dá nastaviť.

Táto aplikácia implementuje funkciu, ktorá všetkým doteraz analyzovaným programom chýbala - zobrazovanie krivky priebehu cesty v mape. Chýbajú jej však všetky ostatné funkcie, takže nezobrazuje nadmorskú výšku, rýchlosť ani prejdenú vzdialenosť. Jednoduchým spôsobom však poukazuje na to, že prepojenie NMEA analyzátoru s mapou nie je problém.

2.3.4 Ďalšie programy

Medzi ďalšie analyzované programy patrí program GPS Express, N2K Analyzer a SA Watch. Tieto programy však nedokážu pracovať s textovým súborom, preto sme ich kvôli absencii skutočného GPS prijímača nemali možnosť prakticky otestovať. Uvádzame teda iba popis funkcií získaný zo stránok programov a z ich používateľských príručiek [1] [7] [18].

GPS Express je program americkej spoločnosti Steelwater Solutions. Dokáže NMEA vety spracovávať aj generovať, zo získaných údajov vie vypočítať štatistiku. Dokáže zobrazovať trasu získanú z NMEA-0183 viet na mape, používa na to proprietárnu technológiu esMapMaster.

Program N2K Analyzer je zameraný na analýzu viet nového štandardu NMEA 2000, vety štandardu NMEA-0183 nepodporuje. Pochádza z dielne americkej firmy Maretron, ktorá zároveň vyrába aj navigačné a iné námorné zariadenia. Program slúži na analýzu a správu zariadení pripojených v NMEA 2000 sieti, po pripojení do siete dokáže zistiť topológiu siete a informácie o jednotlivých zariadeniach v nej. Nedokáže teda priamo spracovávať údaje prenášané zariadeniami, štandard NMEA 2000 je totiž založený na inom princípe ako bol štandard NMEA-0183.

Posledným analyzovaným programom je SA Watch od firmy SRHunting Inc. Program je distribuovaný ako shareware, primárne slúži na zistenie presnosti GPS prijímačov. Túto presnosť zisťuje analýzou NMEA-0183 viet. Získané informácie

dokáže zobrazit v tabulkovej forme, keďže je však primárne určený na určovanie presnosti prijímača, nemá implementované ďalšie funkcie.

2.4 Analýza existujúcich knižníc

Podobne ako programy, existujú aj knižnice určené na uľahčenie spracovania NMEA viet. Tieto knižnice sú väčšinou poskytované v open-source podobe, existujú však aj proprietárne riešenia. Podarilo sa nám nájsť knižnice určené na rozbor NMEA viet v programovacom jazyku C, C++ a Java. V tejto podkapitole ich stručne popíšeme a zhrnieme ich výhody a nevýhody.

2.4.1 NMEA Library

NMEA Library je open-source knižnica pre programovací jazyk C určená na prácu s NMEA protokolom. Dokáže analyzovať NMEA vety, pričom údaje ktoré z nich získa vkladá do údajových štruktúr v jazyku C, čím uľahčuje prácu s nimi. Zároveň dokáže NMEA vety generovať. Podporuje vety typu GPGGA, GPGSA, GPGSV, GPRMC a GPVTG.

2.4.2 NMEAP

NMEAP je knižnica napísaná v jazyku C rozširovaná ako open-source. Rovnako ako predošlá knižnica spracováva NMEAP vety NMEA protokolu a získanými údajmi naplňa štruktúry jazyka C. Rozbor NMEA viet prebieha takto [15]:

1. aplikácia nastaví parser a špecifikuje typy viet, ktoré sa majú analyzovať, rovnako sa špecifikuje kam sa majú ukladať získané údaje.
2. aplikácia prečíta bajty zo vstupného zariadenia a pošle získaný reťazec do funkcie knižnice NMEAP.
3. knižnica NMEAP:
 - pošle prijatý reťazec do lexikálneho skenera, ten rozpozná kompletne a platné vety.
 - pokiaľ je veta rozpoznaná, je poslaná do ďalšieho lexikálneho skenera, ktorý rozdelí túto vetu na jednotlivé časti.

- podľa identifikátora vety sa rozpoznané časti vety uložia do údajových štruktúr, tieto štruktúry obsahujú bežné údajové typy ako integer, float, double a podobne.
- upozorní aplikáciu, že veta bola nájdená a dekodovaná

2.4.3 NMEA:: Library

Táto knižnica je primárne určená pre vnorené systémy založené na obvode Wiring and Arduino [13], je však napísaná v jazyku C++ a jej zdrojové kódy sú voľne dostupné, preto ju možno jednoducho portovať na normálne systémy. Oproti predchádzajúcim spomenutým knižniciam obsahuje aj funkcie na výpočet údajov ako sú rýchlosť, smer cesty, vzdialenosť do určeného bodu na Zemi a podobne.

2.4.4 HS GPS DLL Library

Jediná proprietárna knižnica v našej analýze je od írskej firmy Hillstone Software. Jej cena je 250 EUR, verzia so zdrojovými kódmi sa dá kúpiť za 500 EUR. Táto knižnica je distribuovaná ako dynamická knižnica DLL systému Windows, dokáže dekodovať NMEA vety typu GPGGA, GPRMC a GPVTG na určenie pozície (zemepisnej šírky a dĺžky), súčasnej rýchlosti, nadmorskej výšky a UTC času [2]. Poskytuje tiež rozhranie na prácu so sériovým portom.

2.4.5 NMEA Java APIs

Ako názov napovedá, táto knižnica je naprogramovaná a určená na použitie v programovacom jazyku Java. Je šírená spolu so zdrojovými kódmi a používa architektúru MVC (Model-View-Controller) . Obsahuje teda tri komponenty, ktoré navzájom spolupracujú na spracovaní NMEA viet. Komponent Model má na starosti čítanie údajov zo zdroja. Po úspešnom načítaní určitej časti údajov pošle Model správu o tejto skutočnosti komponentu Controller. Tento komponent získané údaje analyzuje a pokiaľ nájde určité typy viet (typy sa dajú nastaviť), pošle správu komponentu View, ktorý s analyzovanými údajmi môže ďalej pracovať [14].

Možnosti tejto knižnice sú obmedzené iba na analýzu viet typu HDM, DBT a

GLL, vďaka použitej architektúre a dostupnosti zdrojových kódov však nie je problémom naprogramovať analyzačné procedúry pre ďalšie typy viet.

2.5 Zhrnutie analýzy

Analyzovali sme viaceré softvérové prostriedky na spracovanie NMEA viet. Z analýzy môžeme usúdiť, že všetky programy určené na analýzu NMEA viet obsahovali okrem zobrazovania údajov získaných z týchto viet aj nejaké dodatočné funkcie ako napríklad kreslenie hodnôt do grafu, zobrazovanie koordinátov do mapy a podobne. Žiadna z analyzovaných aplikácií však nevysvetľovala čo jednotlivé časti NMEA viet znamenajú, väčšina aplikácií bola určená na použitie v praxi a toto používanie vyžadovalo aspoň základné znalosti protokolu NMEA. Zároveň sa nám nepodarilo nájsť aplikáciu, ktorá by obsahovala všetky funkcie. Ak napríklad aplikácia dokázala kresliť koordináty do mapy, nedokázala kresliť grafy a podobne.

3 Opis riešenia

Obsahom tejto kapitoly je špecifikácia požiadaviek, návrh riešenia a samotná implementácia aplikácie určenej na analýzu NMEA viet. Sú to všetky fázy, ktorými sme pri vytváraní programu prešli. Vstupom aplikácie by mali byť NMEA vety v textovej podobe, výstupom potom hodnoty získané z týchto viet reprezentované rôznymi formami, či už textovými alebo grafickými.

3.1 Špecifikácia požiadaviek

Základnou požiadavkou je schopnosť analýzy NMEA protokolu. Keďže v prípade NMEA sa jedná o textové údaje, je potrebné implementovať aplikačnú logiku, ktorá bude schopná vykonať rozbor jednotlivých viet (parsing) a získané hodnoty poskytnúť prezentačnej vrstve. V prvej fáze projektu teda pôjde o vytvorenie vhodného základu aplikácie, teda vstupnej a logickej vrstvy. V ďalšej fáze pôjde o vytvorenie prezentačnej vrstvy, v ktorej budú získané hodnoty prezentované buď ako text, alebo v grafickej podobe. Programu tiež nesmie chýbať nejaká forma pomocníka, či už na vysvetlenie významu jednotlivých údajov, alebo na prácu s programom samotným.

3.1.1 Ciele

Hlavným cieľom projektu je vytvorenie používateľsky prívetivej aplikácie na analýzu NMEA viet, ktorá bude analyzované vety prezentovať v prehľadnej forme, dokáže týmto hodnotám priradiť určitý praktický význam a bude mať intuitívne ovládanie. Aplikácia by mala byť užitočná rovnako pre ľudí, ktorí už majú skúsenosti s navigačnými a satelitnými technológiami, ako aj pre začiatočníkov v týchto oblastiach. Cieľom je teda vytvoriť aplikáciu s možnosťou využitia v praxi, ktorá bude mať tiež aj výukový charakter.

3.1.2 Funkcie

Aplikácia sa bude skladať z troch vrstiev:

1. vstupná vrstva - získavanie NMEA viet

2. logická vrstva - analýza NMEA viet
3. prezentačná vrstva - prezentácia údajov

Funkciou vstupnej vrstvy bude získavanie NMEA viet, konkrétne by táto vrstva mala zvládnuť načítavanie viet zo súboru a tiež z pripojeného GPS zariadenia. Tieto vety potom poskytne logickej vrstve, ktorá vykoná ich analýzu a získané hodnoty poskytne prezentačnej vrstve. Prezentačná vrstva potom tieto hodnoty prezentuje v rôznych podobách používateľovi.

3.2 Návrh

Naše riešenie by malo do spomenutých troch vrstiev začleniť nasledujúce funkcie:

- vstupná vrstva
 - podpora prijímania údajov z reálneho zariadenia aj zo záznamu (súboru)
 - ukladanie údajov v prípade prijímania zo živého vstupu
- logická vrstva
 - podpora analýzy viet štandardu NMEA-0183, konkrétne typov GGA, GSA, GSV, GLL a RMC
- prezentačná vrstva
 - zobrazovanie údajov ako zemepisná šírka a dĺžka, nadmorská výška a priemerná rýchlosť
 - filtrovanie viet podľa typu a ďalších kritérií
 - rávanie štatistík zo zaznamenaných údajov - dĺžka cesty, časové trvanie cesty, priemerná rýchlosť, prevýšenie a podobne
 - znázorňovanie zaznamenaných údajov vo forme grafu
 - znázornenie trasy zo získaných údajov na mape
 - možnosť simulácie reálneho zariadenia v prípade prijímania údajov zo súboru

3.2.1 Vstupná vrstva

Vstupná vrstva bude obsahovať funkcie na načítavanie NMEA viet zo súboru a čítanie viet zo zariadenia. V oboch prípadoch budú vety ukladané do pamäte,

v prípade čítania viet zo zariadenia môže byť obsah tejto pamäte uložený do voliteľného súboru. Táto pamäť bude tiež zdieľaná s ďalšími dvoma vrstvami.

Funkcie vstupnej vrstvy teda sú:

- práca so súbormi - načítavanie a ukladanie textových údajov
- práca so sériovým portom - nájdenie existujúcich a neobsadených portov, nastavenie protokolu a príjem údajov
- ukladanie získaných údajov do spoločnej pamäte, zapisovanie obsahu tejto pamäte do súboru

3.2.2 Logická vrstva

Logická vrstva bude čítať vety zo zdieľanej pamäte a bude ich analyzovať. Keďže sa jedná o textové reťazce, najvhodnejším spôsobom by bolo využitie tzv. regulárnych výrazov (regex). Pomocou týchto výrazov bude každá veta identifikovaná a následne rozdelená na jednotlivé časti. Tieto časti sa ďalej poskytnú prezentačnej vrstve. Logická vrstva však bude vykonávať aj ďalšie spracovanie týchto častí, konkrétne by sa malo jednať o ich konverziu do čitateľného formátu (napr. v prípade času a dátumu) alebo do formátu, ktorý je bežne používaný v navigáciách (napr. v prípade zemepisnej dĺžky a šírky). Tieto údaje bude logická vrstva poskytovať prezentačnej vrstve vo vhodnej forme.

Funkcie logickej vrstvy:

- získavanie textových údajov zo zdieľanej pamäte
- analýza textových údajov pomocou regulárnych výrazov
- spracovanie analyzovaných údajov - prevod do foriem, ktoré budú ľahko použiteľné v prezentačnej vrstve
- poskytovanie spracovaných údajov vo vhodnej forme prezentačnej vrstve

Spracovanie údajov

Program by mal byť schopný spracovávať 5 typov NMEA viet. Z každej z týchto viet sa dajú získať rôzne údaje, ktoré je potrebné na ďalšiu prácu spracovať. Logická vrstva bude spracovávať tieto údaje:

- veta typu GGA
 - zemepisná šírka a dĺžka - prevod do formátu stupne-minúty-sekundy
 - kvalita signálu - neprijateľná (0) alebo prijateľná (1 a vyššie)
 - počet používaných satelitov (0 až 99)
 - nadmorská výška
- veta typu RMC
 - UTC čas - prevod do formátu HH:MM:SS
 - zemepisná šírka a dĺžka - prevod do formátu stupne-minúty-sekundy
 - dátum - prevod do formátu DD.MM.RRRR
 - platnosť údajov - platné (A) alebo neplatné (V)
- veta typu GSV
 - počet viditeľných satelitov
 - identifikačné číslo každého satelitu - 0 až 99
 - sila signálu pre každý viditeľný satelit - 0 až 99
 - elevácia a azimut každého satelitu - desatinné číslo v stupňoch
- veta typu GSA
 - identifikačné čísla používaných satelitov
- veta typu GLL
 - zemepisná šírka a dĺžka - prevod do formátu stupne-minúty-sekundy
 - UTC čas - prevod do formátu HH:MM:SS
 - platnosť údajov - platné (A) alebo neplatné (V)

3.2.3 Prezentčná vrstva

Prezentčná vrstva bude mať za úlohu prezentovať údaje získané z logickej vrstvy. Základnou formou prezentácie bude výpis jednotlivých prijatých viet v pôvodnom formáte spolu s poradovým číslom a časom prijatia. Na tento zoznam viet bude možné aplikovať filter. Ďalej táto vrstva poskytne napríklad zoznam častí vybranej vety aj s ich popismi. Rovnako tiež poskytne všetky užitočné údaje, ktoré boli z danej vety získané logickou vrstvou. Medzi zaujímavé funkcie tejto vrstvy bude tiež patriť aj kreslenie grafov a kreslenie koordinátou do mapy. Úlohou prezentačnej

vrstvy bude taktiež počítanie štatistík zo spracovaných údajov a ich zobrazovanie. Medzi plánované funkcie patrí aj možnosť simulovať GPS zariadenie časovaným zobrazovaním jednotlivých prijatých viet podľa získaného času.

Funkcie prezentačnej vrstvy:

- poskytnutie grafického rozhrania na ovládanie programu
- zobrazovanie údajov v textovej forme s možnosťou aplikácie filtra
- zobrazovanie údajov v grafickej forme s ďalšími možnosťami
- spracovanie prijatých údajov - počítanie štatistík
- simulácia reálneho zariadenia s možnosťou zrýchliť, spomaliť a pozastaviť jej priebeh

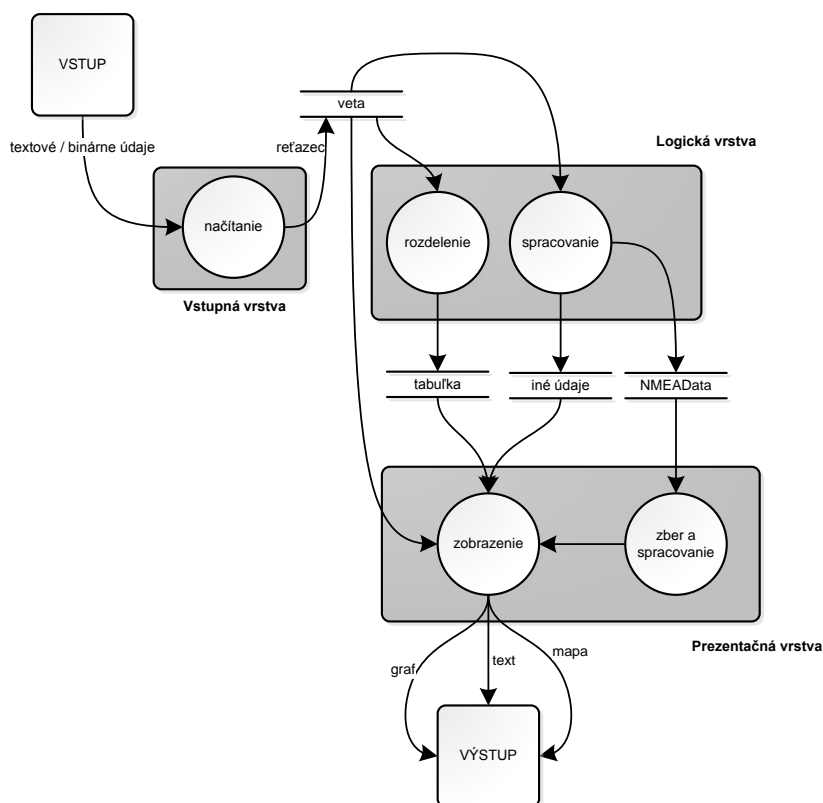
Zobrazovanie údajov

Vzhľadom na množstvo a rozmanitosť údajov, ktoré budú prijímané z logickej vrstvy je potrebné zobrazovať ich v rôznych formách. Keďže má mať program aj výučbový charakter, bolo potrebné niektoré údaje zobrazovať viacerými spôsobmi.

- zemepisná šírka a dĺžka - budú zobrazované v textovej forme a rovnako aj ako bod na mape, body budú pospájané, aby bola viditeľná trasa
- nadmorská výška - v textovej forme, aj vo forme grafu
- sila signálu zo satelitov - vo forme grafu, textové údaje budú dostupné pri výbere jednotlivých viet
- viditeľné satelity - vo forme mapy, rozlíšené budú používané satelity, viditeľné satelity a satelity so slabým signálom
- štatistiky - čas cesty, dĺžka trasy, prevýšenie, priemerná rýchlosť - dostupné v textovej forme
- aktuálne údaje (pri simulácií a čítaní z reálneho zariadenia) - čas, dátum, poloha, výška - v textovej forme

3.3 Implementácia

Na implementáciu aplikácie bol vybraný programovací jazyk Java, najmä kvôli nezávislosti na operačnom systéme, ako aj kvôli existencii podrobnej dokumentácie funkcií na prácu s textovými reťazcami. Ďalším faktorom pri výbere bola schopnosť vytvárať grafické rozhrania. Na to v jazyku Java existuje niekoľko knižníc. Viacero knižníc existuje aj na kreslenie grafov a obrázkov a na sieťovú komunikáciu. Tieto schopnosti budú nevyhnutné na prácu s mapami, pokiaľ by sme napr. chceli využiť možnosti služby Google Maps alebo podobných. Používali sme knižnice Java SDK vo verzii 1.6.0_03, ako vývojové prostredie sme si vybrali program NetBeans vo verzii 6.8.



Obr. 3: Zjednodušený diagram toku údajov

3.3.1 Výber prostriedkov

Pre jazyk Java existuje veľké množstvo programovacích knižníc, ktoré dokážu uľahčiť prácu pri tvorbe grafického rozhrania, alebo implementujú zložité výpočtové algoritmy efektívnejšie. My sme sa v našom programe rozhodli použiť niekoľko takýchto knižníc, ktoré si v tejto podkapitole popíšeme.

Pri implementácii samotnej aplikácie sme postupovali podľa špecifikácie *JSR-296: “Java Application Framework”* [16], ktorá určuje infraštruktúru obvyklú pre väčšinu bežných aplikácií. Zároveň uľahčuje prácu s knižnicou Swing a poskytuje niekoľko vylepšení, ako napríklad vytváranie a spravovanie vláken, ukladanie stavu okna a podobne.

Knižnica **Swing** je v súčasnosti súčasťou základného balíka tried od firmy Sun (JFC - Java Foundation Classes), poskytuje aplikačné rozhranie na prácu s grafickými prvkami ako sú okná, tlačidlá, texty, tabuľky a podobne.

Knižnica **RXTX** [10] je určená na prácu so sériovým portom. Firma Sun, ktorá vyvíja samotný jazyk Java a jej štandardné knižnice skončila s podporou sériového portu pod operačným systémom Windows vo verzii Java 1.2, a v súčasnosti odporúča na prácu so sériovým portom používať práve knižnicu RXTX, pretože tá je aj v súčasnosti aktívne udržiavaná.

Knižnica **JFreeChart** [9] poskytuje prostriedky na vytváranie zložitých grafov jednoducho. Dokáže zobrazovať veľké množstvo rozličných typov grafov, ako sú napríklad grafy čiarové, pruhové, stĺpcové, koláčové, Ganttové grafy, alebo grafy X-Y závislosti.

Knižnicu **SwingX-WS** [3] tvoria triedy určené na interakciu s webovými službami. My z tejto knižnice využijeme triedu na vykresľovanie máp z online služieb ako napr. OpenStreetMaps.

Na implementáciu pomocníka do aplikácie použijeme knižnicu **JavaHelp**. Posky-

tuje prostredie podobné klasickému pomocníkovi zo systému Windows, dokáže tiež zobrazovať pomocníka na základe špecifických podnetov z aplikácie.

Keďže budeme pracovať s veľkým objemom textových údajov, na urýchlenie filtrovania týchto údajov použijeme knižnicu **TrueFilter** od nemeckého autora Christiana Schlichtherle [17]. Pomocou tohto filtra sa dá dosiahnuť filtrovanie dynamických údajov, teda údajov, ktoré sa počas aplikácie filtra môžu meniť.

3.3.2 Vstupná vrstva

V tejto vrstve využívame na prácu so sériovým portom knižnicu RXTX. Pri práci s touto knižnicou sme narazili na niekoľko problémov, ktoré boli spôsobené chybami v implementácii tejto knižnice, na splnenie našich minimálnych požiadaviek však táto knižnica postačuje. Na prácu so súbormi používame bežné metódy jazyka Java, ktoré používajú pri čítaní vyrovnávaciu pamäť.

Vstupná vrstva je rozdelená na 3 časti:

1. trieda čítajúca údaje zo súboru
2. trieda čítajúca údaje zo súboru, určená na simuláciu
3. trieda čítajúca údaje zo sériového portu

Každá z týchto tried je implementovaná ako podtrieda triedy *Task* zo špecifikácie *JSR-296*, pracuje teda v samostatnom vlákne - zvyšok aplikácie sa počas čítania údajov dá naďalej používať. Vlákno komunikuje so zvyškom programu pomocou tzv. zdieľaných premenných, ktorých väčšina je zjednotená v triede *SharedData*.

Obe triedy čítajúce zo súboru sú si dosť podobné, čítajú údaje z textového súboru riadok po riadku, každý riadok je po prečítaní odoslaný do logickej vrstvy na spracovanie. Pri triede určenej na simuláciu je po odoslaní údajov do logickej vrstvy zistený z vety čas, ktorý je uložený, a pri každej ďalšej vete porovnávaný s novým získaným časom. Pokiaľ sa časy nezhodujú, vlákno sa na určitý časový interval pozastaví, čím sa simuluje pozdržanie prijímania, ktoré je bežné pri prijímaní z reálneho zariadenia. Tento časový interval sa dá nastaviť z prezentačnej vrstvy, z ktorej je tiež možné ľubovoľne beh vlákna pozastaviť.

Trieda pracujúca so sériovým portom sa spúšťa z prezentačnej vrstvy po nakonfigurovaní, pracuje podobne ako zvyšné 2 triedy s tým rozdielom, že údaje sú zo sériového portu čítané v binárnej forme, nie textovej. Je teda potrebné ukladať ich priebežne do vyrovnávacej pamäte a postupne ich rozdeľovať na riadky. Trieda posiela prečítané údaje ďalej vždy keď má v pamäti celú vetu.

3.3.3 Logická vrstva

Obsahom tejto vrstvy sú najmä metódy triedy *NMEAParser* určené na analýzu viet a na ich ďalšie spracovanie. Analýza sa vykonáva pomocou regulárnych výrazov. Prijatá trieda je rozdelená na časti pomocou definície, ktorá určuje čo sa má na ktorom mieste vo vete nachádzať. Každá veta má vlastnú definíciu, niektoré ich majú kvôli kompatibilite s rôznymi verziami NMEA štandardu viac (napr. veta typu RMC), niektoré definície je potrebné dynamicky skladať kvôli charakteru analyzovanej vety (veta typu GSV).

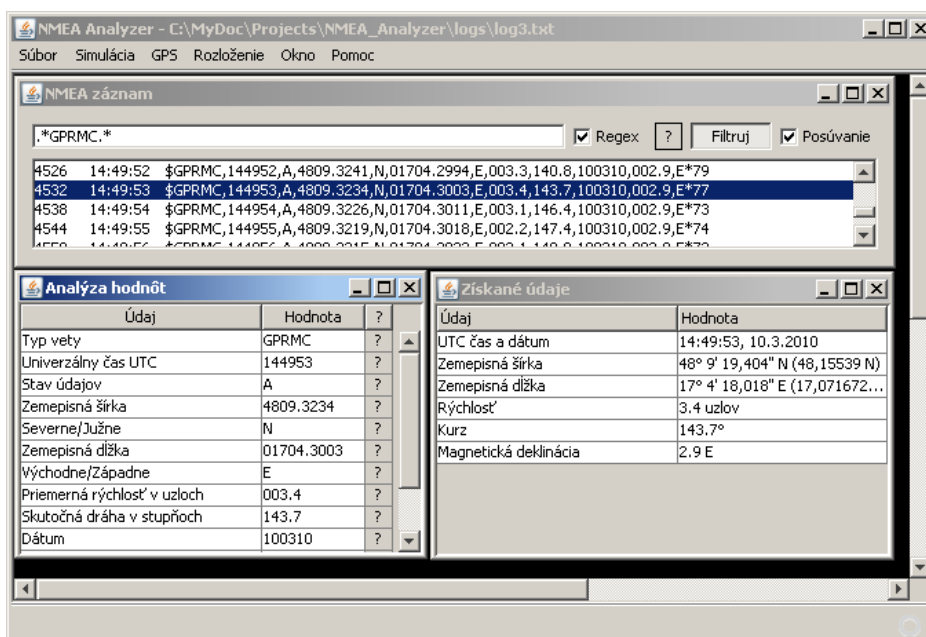
Rozdelená veta je poslaná prezentačnej vrstve v tabuľkovom formáte so stĺpcami ‘‘Popis’’ a ‘‘Údaje’’, a okrem toho je ďalej spracovávaná. Spracované údaje sú buď posielané prezentačnej vrstve v čistej forme (pokiaľ sa už s nimi ďalej nepracuje), alebo sú zabalené do triedy *NMEADData*. Údaje uložené v tejto triede sú určené na ďalšie spracovanie prezentačnou vrstvou, sú v nej preto uložené v transformovanej forme, teda vo forme po prevode na požadovaný tvar. Jedná sa napríklad o čas, dátum, zemepisnú šírku a dĺžku, nadmorskú výšku, zoznam satelitov alebo platnosť údajov. Tieto údaje sú nastavované individuálne pre každú vetu a posielajú sa ďalej prezentačnej vrstve v ktorej sa zbierajú, spracovávajú a zobrazujú.

Transformácia údajov väčšinou spočíva v jednoduchom prevode textových reťazcov na čísla, v niektorých prípadoch sú potrebné ďalšie úpravy. Pri zemepisnej šírke a dĺžke je napríklad potrebný prevod na všeobecne známy systém stupne-minúty-sekundy, pri spracovávaní údajov o satelitoch je zase potrebné vytvoriť ich zoznam a pre každý satelit uložiť údaje o ich elevácií, azimute a sile signálu. Na tento účel

bola vytvorená trieda *SatInfo*, ktorá všetky tieto údaje združuje.

3.3.4 Prezentčná vrstva

Základom prezentačnej vrstvy je grafické rozhranie implementované pomocou grafickej knižnice Swing podľa špecifikácie *JSR-296*. Pri tvorbe grafického rozhrania nám vo významnej miere pomohol nástroj “Swing GUI Builder” obsiahnutý v programe NetBeans. Tento program zároveň poskytoval predlohu na tvorbu aplikácií podľa nami vybranej špecifikácie. Naša aplikácia obsahuje základné okno rozdelené na 3 časti: menu, plocha a stavová časť. Plocha je implementovaná ako trieda *JDesktopPane*, ktorú sme modifikovali (výsledok = trieda *MDIDesktopPane*). V nej sa nachádzajú vnorené okná (trieda *JInternalFrame*) zobrazujúce získané údaje. Tieto okná je možné ľubovoľne po ploche posúvať, maximalizovať, minimalizovať a zatvárať. Grafické rozhranie teda pracuje na princípe pracovnej plochy systému Windows.



Obr. 4: Hlavné okno programu s tromi vnorenými oknami

Program obsahuje hlavné menu a nasledujúce vnorené okná:

- NMEA záznam
- Analýza hodnôt
- Získané údaje
- Nadmorská výška
- Mapa
- Sila signálu satelitov
- Štatistiky a aktuálne údaje
- Ovládanie simulácie
- Pozície satelitov

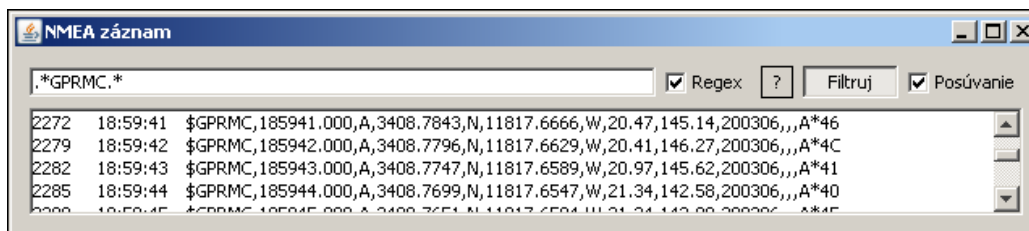
V nasledujúcej časti popíšeme obsah hlavného menu a jednotlivých vnorených okien.

Názvy a funkcie volieb v hlavnom menu sú nasledovné:

- Súbor -> Otvoriť - vytvorí inštanciu triedy zo vstupnej vrstvy slúžiacu na načítanie súboru
- Súbor -> Ukončiť - ukončí beh programu
- Simulácia -> Otvoriť na simuláciu - vytvorí inštanciu triedy slúžiacu na simuláciu
- Simulácia -> Ukončiť simuláciu - zastaví beh vlákna simulácie a odstráni inštanciu triedy
- GPS -> Pripojiť ku GPS - zobrazí dialóg na nastavenie sériového portu, následne vytvorí inštanciu triedy slúžiacu na komunikáciu so sériovým portom
- GPS -> Odpojiť - ukončí komunikáciu so sériovým portom, zruší inštanciu danej triedy
- Rozloženie -> Výchbové/Pracovné/Vlastné - nastaví rozmiestnenie a veľkosti okien podľa vopred nastavených hodnôt, v poslednom prípade (Vlastné) načíta uložené údaje o rozmiestnení zo súboru
- Rozloženie -> Uložiť rozloženie - uloží rozloženie okien a ich veľkosti do súboru
- Okno -> ... - táto položka obsahuje zoznam všetkých dostupných okien v programe, po kliknutí na voľbu sa vybrané okno zobrazí
- Pomoc -> Pomocník - otvorí pomocníka k programu, tá je implementovaná pomocou systému *JavaHelp*.

- Pomoc -> O programe - zobrazí dialógové okno s informáciami o autorovi

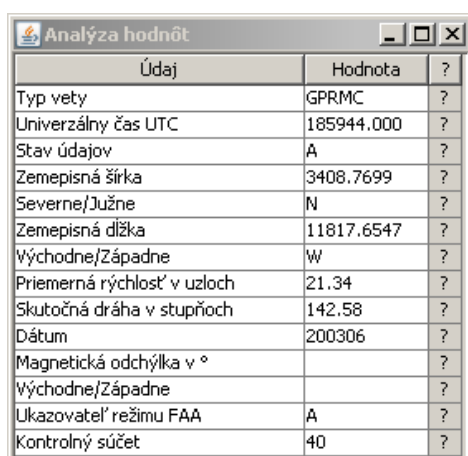
Vnorené okno “NMEA záznam” (obrázok 5) obsahuje grafický prvok *JList*. Tomuto prvku je pridelený údajový model (*DefaultListModel*), do ktorého sú pridávané vety vstupnou vrstvou hneď po prečítaní. Prvok sme pozmenili tak, aby zobrazoval naraz tri stĺpce. V prvom stĺpci je zobrazované poradové číslo vety - to sa počíta vo vstupnej vrstve. V druhom stĺpci je čas získaný z danej vety a v treťom je veta samotná. Čas je získavaný priamo z vety, pokiaľ ide o typ vety ktorá čas neobsahuje, použije sa posledný nastavený. Celý tento zoznam slúži na výber vety na analýzu. Po vybratí vety sa nastaví grafické prvky v ostatných oknách, konkrétne v oknách “Analýza hodnôt”, “Získané údaje” a “Mapa”. Okno obsahuje aj textové pole (*JTextField*) do ktorého môže byť zadaný filter na obmedzenie výpisu. Filtrovanie funguje buď na princípe porovnávania začiatkov viet s obsahom textového poľa, alebo - ak je zaškrtnutá možnosť “Regexp” - sa zadaný výraz berie ako regulárny. Filtrovanie sa vykoná po stlačení tlačidla “Filtruj”, po opätovnom stlačení sa filtrovanie zruší. Okno obsahuje aj možnosť “Posúvanie”, ktoré je užitočné v režime simulácie alebo prijímaní z reálneho zariadenia - indikuje, či sa má zoznam viet posúvať, alebo nie.



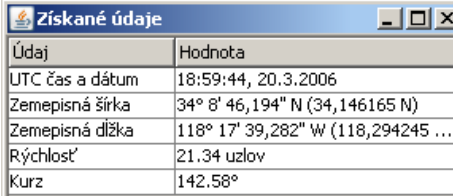
Obr. 5: Vnorené okno “NMEA záznam”

Vnorené okno “Analýza hodnôt” (obrázok 6) obsahuje tabuľku (grafický prvok *JTable*), v ktorom sú (po vybratí vety v okne “NMEA záznam”) vypísané jednotlivé časti vety spolu s ich popisom. Údaje sú do tabuľky pridávané pomocou údajového modelu, obsah toho spravuje logická časť programu. Časti viet sú v základnom tvare bez ďalšieho spracovania, každý riadok tabuľky má v poslednom stĺpci tlačidlo s otáznikom, po jeho stlačení sa otvorí pomocníka k danej časti vety. Toto okno má viac-menej výukový charakter.

Vo vnorenom okne “Získané údaje” (obrázok 6) sú zobrazené údaje spolu s popisom po spracovaní, teda po transformácii na určitý formát. Zobrazenie je riešené opäť pomocou tabuľky, transformácia údajov a vytváranie údajového modelu pre túto tabuľku je vykonávané v logickej vrstve. Používateľ môže porovnať hodnoty pred a po transformácií.

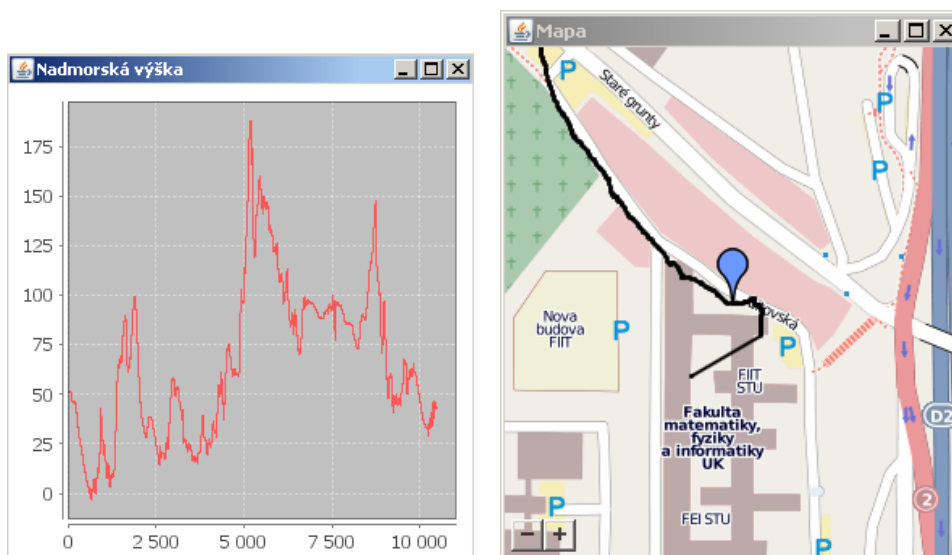


Údaj	Hodnota	?
Typ vety	GPRMC	?
Univerzálny čas UTC	185944.000	?
Stav údajov	A	?
Zemepisná šírka	3408.7699	?
Severne/Južne	N	?
Zemepisná dĺžka	11817.6547	?
Východne/Západne	W	?
Priemerná rýchlosť v uzloch	21.34	?
Skutočná dráha v stupňoch	142.58	?
Dátum	200306	?
Magnetická odchýlka v °		?
Východne/Západne		?
Ukazovateľ režimu FAA	A	?
Kontrolný súčet	40	?



Údaj	Hodnota
UTC čas a dátum	18:59:44, 20.3.2006
Zemepisná šírka	34° 8' 46,194" N (34,146165 N)
Zemepisná dĺžka	118° 17' 39,282" W (118,294245 ...)
Rýchlosť	21.34 uzlov
Kurz	142.58°

Obr. 6: Vnorené okná “Analýza hodnôt” (vľavo) a “Získané údaje” (vpravo)



Obr. 7: Vnorené okná “Nadmorská výška” (vľavo) a “Mapa” (vpravo)

Vnorené okno “Nadmorská výška” (obrázok 7) obsahuje graf nadmorskej

výšky v rozsahu celého načítaného záznamu. Os X tvorí poradové číslo vety v ktorej sa údaj o výške nachádzal, os Y je potom hodnota nadmorskej výšky v metroch nad morom. Tento graf je implementovaný pomocou knižnice *JFreeChart*, konkrétne triedou *JLineChart*. Dá sa približovať a oddaľovať, po ukázaní myšou na čiaru grafu sú zobrazené presné hodnoty súradníc X a Y. Na údaje v tomto okne nemá vplyv výber vety v okne “NMEA záznam”.

Vnorené okno “Mapa” (obrázok 7) obsahuje mapu, do ktorej je kreslená trasa celého záznamu. Mapa je implementovaná pomocou knižnice *SwingX-WS*, bolo však potrebné vytvoriť kresliaci algoritmus, keďže sme potrebovali do samotnej mapy kresliť čiary a vyznačovať body. Mapy sú získavané z Internetu pomocou služby *OpenStreetMap* - získavanie máp pomocou služby *Google Maps* bolo v prvej fáze nášho projektu nelegálne, a aj keď bola táto licenčná politika počas druhej fázy zmenená, nepodarilo sa nám ju včas implementovať. Po výbere vety v okne “NMEA záznam” sa v mape zobrazia koordináty, ktoré vybraná veta obsahuje. Keďže aplikácia neobsahuje databázu máp znamená to, že pokiaľ nie je dostupné pripojenie na Internet, mapy sa nevykresľujú.

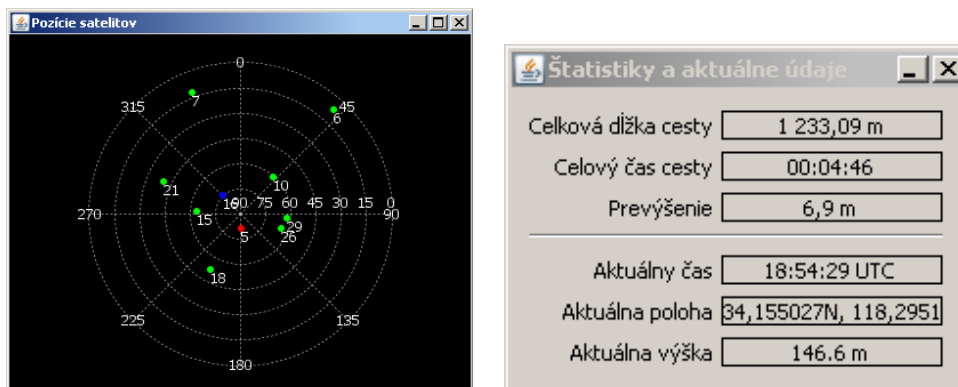
Vnorené okno “Sila signálu satelitov” (obrázok 9) obsahuje stĺpcový graf sily signálu zo satelitov. Graf je implementovaný opäť pomocou knižnice *JFreeChart*, tentoraz triedou *JBarChart*. Údaje o používaných satelitoch a sile ich signálu sú zhromažďované v údajovej štruktúre *ArrayList*, ktorá sa naplňuje údajmi z logickej vrstvy. Obsah tohto okna je využiteľný len pri simulácií a pri čítaní údajov z reálneho zariadenia, keďže inak zber údajov nie je vykonávaný.

Vo vnorenom okne “Štatistiky a aktuálne údaje” (obrázok 8) je 6 textových polí, 3 z nich zobrazujú štatistické údaje ako celkový čas cesty, celková dĺžka trasy a prevýšenie, zvyšné 3 zobrazujú aktuálne údaje pri simulácií alebo čítaní z reálneho zariadenia, konkrétne aktuálny čas, pozíciu a výšku. Celkový čas cesty je rátaný tak, že sa uloží prvý známy čas, ktorý je odpočítaný od každého ďalšieho získaného času. Časy sú prerátané na sekundy, vypočíta sa rozdiel a ten je prerátaný naspäť na hodiny, minúty a sekundy. V tomto výpočte je tiež ošetrený prechod medzi dvoma dňami. Celková dĺžka trasy sa počíta ako Euklidovská vzdialenosť medzi dvoma

bodmi v priestore. Bolo však potrebné prerátať polárne súradnice (zemepisná výška a šírka) do karteziánskych súradníc. Boli použité tieto vzorce:

$$\begin{aligned}
 \text{výška} &= \text{polomer Zeme} + \text{nadmorská výška} \\
 P_{\text{polar}} &= ((\text{zemepisná}) \text{ dĺžka}, (\text{zemepisná}) \text{ šírka}, \text{výška}) \\
 P_{\text{kartez}} &= (x, y, z) \\
 x &= \text{výška} \times \cos(\text{šírka}) \times \sin(\text{dĺžka}) \\
 y &= \text{výška} \times \sin(\text{šírka}) \\
 z &= \text{výška} \times \cos(\text{šírka}) \times \cos(\text{dĺžka}) \\
 \text{vzdialenosť} &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}
 \end{aligned}$$

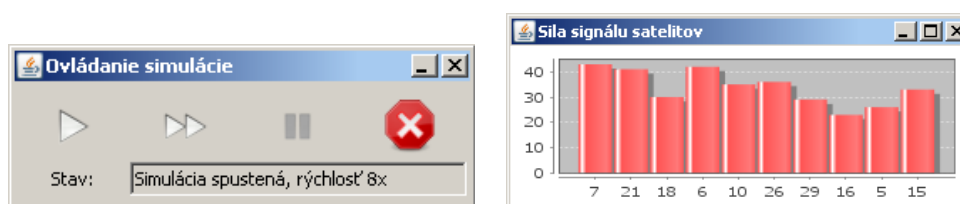
Pomocou nich zisťujeme vzdialenosť medzi každými dvoma bodmi, ktorú potom spočítavame a zobrazujeme ako celkovú dĺžku trasy. Prevýšenie počítame ako rozdiel medzi maximálnou a minimálnou výškou počas celej cesty.



Obr. 8: Vnorené okná “Pozície satelitov” (vľavo) a “Štatistiky a aktuálne údaje” (vpravo)

Ďalšie vnorené okno má názov “Ovládanie simulácie” (obrázok 9). Zobrazuje sa len po spustení simulácie, alebo na vyžiadanie. Obsahuje 4 tlačidlá a textové pole indikujúce aktuálny stav simulácie. Jednotlivé tlačidlá sú sprístupňované po načítaní súboru na simuláciu (“Spustiť”) a po spustení simulácie (“Zvýšiť rýchlosť”, “Pozastaviť” a “Ukončiť simuláciu”). Tlačidlom “Zvýšiť rýchlosť”

sa znižuje časový interval čakania vo vstupnej vrstve, čím sa urýchľuje simulácia. Simulácia môže bežať maximálne 8x rýchlejšie ako je normálny čas, po prekročení tejto hranice sa rýchlosť nastavuje opäť odznova. Aktuálna rýchlosť je zobrazovaná v textovom poli v tomto okne. Po stlačení tlačidla “Ukončiť simuláciu” sa simulácia ukončí - zastaví sa beh daného vlákna a jeho inštancia sa označí na odstránenie - a toto okno sa zavrie.



Obr. 9: Vnorené okná “Ovládanie simulácie” (vľavo) a “Sila signálu satelitov” (vpravo)

Medzi okná zobrazujúce grafy patrí aj okno s názvom “Sila signálu satelitov” (obrázok 9). Tvorí ho stĺpcový graf, v ktorom sú zobrazené hodnoty sily signálu z používaných satelitov. Os X v tomto prípade tvoria identifikačné čísla satelitov, os Y je potom hodnota sily signálu SNR udávaná v dB. Graf je implementovaný ako trieda *JBarChart* z knižnice *JFreeChart*, údaje o hodnotách sily signálov sú získavané z viet typu GSV a sú uložené v inštancii triedy *DefaultCategoryDataset*, do nej sú pridávané v prezentačnej vrstve po spracovaní každej takejto vety.

Posledným vnoreným oknom je okno s názvom “Pozície satelitov” (obrázok 8). Tvorí ho polárny graf z knižnice *JFreeChart* implementovaný triedou *PolarChart*. Na vykreslenie satelitov bolo potrebné túto triedu pozmeniť, vytvorili sme pre ňu vlastný vykresľovač (renderer) a tiež vlastný typ objektov, ktoré sa do tohto grafu dajú pridávať. Tieto úpravy boli potrebné, aby sa satelity zobrazili v grafe po zadaní ich azimutu, elevácie a stavu. Používané satelity sú vykreslené zelenou farbou, viditeľné satelity s dobrým signálom modrou farbou a viditeľné satelity so slabým signálom farbou červenou. Každý bod predstavujúci satelit má vedľa seba udané číslo, ktoré predstavuje identifikačné číslo satelitu (PRN).

3.3.5 Pomocník

Pomocník je v našej aplikácii implementovaný pomocou knižnice *JavaHelp*. Obsahuje informácie o používaní programu ako aj informácie o analyzovaných vetách a ich častiach. Vyvolať sa dá buď z menu výberom Pomoc->Pomocník, alebo kliknutím na príslušné tlačidlo v okne “Analýza hodnôt”. Pomocník je implementovaný ako dokumenty vo formáte HTML, ktoré sú otvárané pomocou triedy *HelpSet*. Táto trieda otvorí XML súbor zjednocujúci všetky HTML dokumenty a podľa požiadavky z programu otvorí danú kapitolu (teda HTML súbor).

3.4 Testovanie

Testovanie implementovaného programu prebiehalo v 3 fázach. Najprv sme otestovali funkčnosť pomocou súboru, ktorý obsahoval záznam z reálneho GPS zariadenia. Následne sme otestovali schopnosť programu spolupracovať s takýmto zariadením - testovali sme analýzu údajov z reálneho zariadenia. Na záver sme otestovali simuláciu prijímania pomocou údajov získaných v predošlom teste.

3.4.1 Analýza viet zo súboru

Pri analýze viet zo súboru sme použili záznamy získané pomocou niekoľkých zariadení, aby sme overili kompatibilitu s rôznymi verziami štandardu NMEA. Konkrétne šlo o zariadenia firiem Garmin a TomTom. Odhalili sme mierne odchýlky v definíciách viet typu GPRMC pri údajoch zo zariadení firmy Garmin, boli sme teda nútení pridať ďalšiu definíciu pre vety tohto typu do logickej vrstvy. Získané údaje sme porovnávali s údajmi získanými z rovnakých viet pomocou iných programov spomenutých v časti “Analýza existujúcich riešení”.

3.4.2 Analýza viet zo zariadenia

Pri tomto testovaní sme použili zariadenie Garmin GPS20x. Analýza prebiehala hladko, získané údaje sme porovnávali s údajmi z ďalšieho GPS zariadenia (mobilného telefónu) a aj skutočnosti (v prípade mapy). Nevýhodou nášho zariadenia bola pomerne pomalá úvodná synchronizácia so satelitmi. Zároveň sa občas prejavili chyby v implementácii knižnice na prácu so sériovým portom.

3.4.3 Simulácia

Testovanie režimu simulácie prebiehalo tak, že sme používali údaje získane počas testovania analýzy viet zo zariadenia. Mohli sme teda porovnať nakoľko sa tieto dva režimy podobajú. Zároveň sme testovali možnosť zrýchliť a pozastaviť simuláciu. Zistili sme, že pri vysokých rýchlostiach simulácie (teda nízkych intervaloch pozastavenia vlákna) sa môžu vyskytnúť konflikty pri zápise do zdieľaných premenných, stabilitu aplikácie však tento fakt neovplyvní a všetky funkcie pracujú normálne ďalej.

3.4.4 Výsledky testovania

Počas testovanie sme odhalili viacero malých aj závažnejších chýb, ktoré sme však dokázali odstrániť. Keďže sme nemali priamy prístup k štandardu NMEA a možné hodnoty jednotlivých viet sme mohli určiť len z pozorovaní a internetových zdrojov, nemôžeme s istotou tvrdiť, že náš program dokáže pracovať so všetkými NMEA zariadeniami. Otestovali sme však zariadenia firiem, ktoré sú v súčasnosti najrozšírenejšie a zabezpečili program tak, aby s nimi dokázal spolupracovať. Ďalším z nedostatkov, ktoré sme nedokázali ovplyvniť je fakt, že knižnica na prácu so sériovým portom v jazyku Java nie je najstabilnejšia a tak sa môžu pri pokuse o komunikáciu s reálnym zariadením vyskytnúť problémy. Celkovo však považujeme funkčnosť aplikácie za dostačujúcu na bežnú analýzu NMEA viet.

4 Záver

V prvej časti projektu (kapitola 1 a 2) sme poskytli stručný úvod do GPS navigácie, uviedli sme všetky dôležité informácie o protokole NMEA a popísali sme základné typy viet tohto protokolu. Následne sme vykonali analýzu softvérových nástrojov a programovacích knižníc určených na prácu s NMEA vetami. Z tejto analýzy sme usúdili, že existuje mnoho rôznych spôsobov ako spracovať NMEA vety, žiadny z analyzovaných programov však nemal implementované všetky. Zároveň mali všetky programy skôr praktické využitie, nešlo teda o programy, ktoré by priblížili protokol NMEA začiatočníkom. Všetky analyzované programy a knižnice nám poskytli inšpiráciu a nápady, ktoré sme sa pokúsili zrealizovať a vylepšiť v našom projekte.

V druhej časti projektu (kapitola 3, podkapitoly 3.1 a 3.2) sme špecifikovali požiadavky na našu aplikáciu a vytvorili architektonický návrh. Hlavným cieľom sa stalo vytvorenie používateľsky prívetivej aplikácie na analýzu NMEA viet, ktorá bude mať tak výučbový ako aj praktický charakter. V návrhu sme určili základné funkcie aplikácie - definovali sme 3 základné vrstvy, ktoré medzi sebou komunikujú a zabezpečujú chod aplikácie. Týmto vrstvám sme následne prideliť konkrétnejšie funkcie a určili sme spôsob, akým budú medzi sebou komunikovať.

V nasledujúcej časti projektu (podkapitola 3.3) sme potom preložili definície zo špecifikačného a návrhového jazyka do jazyka Java, určili sme teda konkrétnejšie podoby jednotlivých vrstiev. Popísali sme tu podrobnejšie funkcie týchto vrstiev z implementačného hľadiska, zároveň sme tiež určili podobu grafického rozhrania programu.

V poslednej časti (podkapitola 3.4) sme implementovaný program otestovali vo všetkých dostupných režimoch a vychytali sme všetky chyby, ktoré sme našli. Zároveň sme zistili, že kvalita údajov poskytovaných naším programom závisí od kvality GPS prijímača, a tento fakt nedokážeme žiadnym spôsobom ovplyvniť. Ďalším zisteným problémom bola aj nedostupnosť rôznych verzií NMEA štandardu, čím sme odchýlky v týchto verziách protokolu mohli zistiť len pozorovaním

správania sa rôznych zariadení.

Na záver môžeme povedať, že sa nám podarilo splniť vytýčené ciele. Implementovaná aplikácia je používateľsky prívetivá, má intuitívne ovládanie a poskytuje užitočné funkcie rovnako začiatočníkom ako aj skúsenejším používateľom. Program dokáže analyzovať základné typy viet vysielané zariadeniami GPS, dokáže z nich získať všetky užitočné údaje a tie dokáže prezentovať vo viacerých formách. Samozrejme existuje viacero možností ako program vylepšiť - napríklad implementáciou podpory máp služby Google Maps, vylepšením simulácie a podobne. Rovnako aj z pohľadu optimalizácie je v programe čo zlepšovať. V rámci daných možností sme však vytvorili v praxi využiteľný program, ktorý obsahuje aj výučbové funkcie.

Literatúra

- [1] *GPS Express* [online]. Steelwater Solutions, 2009. [cit. 13. 5. 2009]. Dostupné z: <http://www.steelwater.com/gpsexpress>.
- [2] *HsGpsDll Library User Manual (v1.1)* [online]. Hillstone Software, 2008. [cit. 14. 5. 2009]. Dostupné z: http://www.hillstone-software.com/HsGpsDllManual_v1_1.pdf.
- [3] *SwingX-WS* [online]. SwingLabs, 2010. [cit. 1. 5. 2010]. Dostupné z: <https://swingx-ws.dev.java.net/>.
- [4] DEPRIEST, D. *NMEA data* [online]. 2008. [cit. 13. 5. 2009]. Dostupné z: <http://www.gpsinformation.org/dale/nmea.htm>.
- [5] DEPRIEST, D. *List of NMEA 0183 sentences* [online]. 2009. [cit. 14. 5. 2009]. Dostupné z: <http://gpsd.berlios.de/NMEA.txt>.
- [6] EL-RABBANY, A. *Introduction to GPS: The Global Positioning System, Second Edition*. Boston : Artech House Publishers, 2006. ISBN 1596930160.
- [7] ELLISON, B. *Maretron's N2K Analyzer, a shy beauty* [online]. 2007. [cit. 13. 5. 2009]. Dostupné z: http://www.panbo.com/archives/2007/09/maretrons_n2k_analyzer_a_shy_beauty.html.
- [8] GIBSON, D. *GPS NMEA data to Google Map converter* [online]. 2006. [cit. 12. 5. 2009]. Dostupné z: <http://www.gonmad.co.uk/nmea.php>.
- [9] GILBERT, D. *JFreeChart* [online]. Object Refinery Limited, 2010. [cit. 5. 5. 2010]. Dostupné z: <http://www.jfree.org/jfreechart/>.
- [10] JARVI, K. *RXTX* [online]. 2010. [cit. 4. 5. 2010]. Dostupné z: <http://rxtx.qbang.org/>.
- [11] KAPLAN, E. -- HEGARTY, C. *Understanding GPS: principles and applications*. Boston : Artech House Publishers, 2006. ISBN 1580538940.

- [12] KÖHNE, A. -- WÖSSNER, M. *Position Determination with GPS* [online]. 2009. [cit. 10. 5. 2009]. Dostupné z: <http://www.kowoma.de/en/gps/positioning.htm>.
- [13] LAMERS, M. *NMEA*:: [online]. 2008. [cit. 13. 5. 2009]. Dostupné z: <http://www.maartenlamers.com/nmea/>.
- [14] LE DIOURIS, O. *Java API for NMEA programming* [online]. 2003. [cit. 14. 5. 2009]. Dostupné z: <http://javanmea.sourceforge.net/>.
- [15] M. HOWARD, D. *NMEAP tutorial and reference* [online]. 2005. [cit. 14. 5. 2009]. Dostupné z: <http://www.dmh2000.com/nmea/tutorial.html>.
- [16] POTOCHKIN, A. *JSR-296* [online]. 2010. [cit. 14. 3. 2010]. Dostupné z: <http://jcp.org/en/jsr/detail?id=296>.
- [17] SCHLICHTHERLE, C. *TrueFilter* [online]. 2010. [cit. 28. 4. 2010]. Dostupné z: <https://truefilter.dev.java.net/>.
- [18] YEAZEL, J. *SA Watch Instructions* [online]. 2001. [cit. 13. 5. 2009]. Dostupné z: <http://gpsinformation.net/main/sawatch.htm>.

A Technická dokumentácia

A.1 Hlavné okno

Hlavné okno je vytvorené v triede *NMEA_Analyzer_App* vytvorením inštancie triedy *NMEA_Analyzer_View*. Trieda *NMEA_Analyzer_App* rozširuje triedu *SingleFrameApplication* zo špecifikácie JSR-296.

A.1.1 Hlavná trieda

Je rozšírením triedy *FrameView* a obsahuje väčšinu funkcionality celej aplikácie. Stretávajú sa tu všetky 3 vrstvy. V konštruktoze triedy sa inicializujú všetky grafické komponenty - vnorené okná, menu aj stavový riadok. Tiež sa tu vytvára monitor úloh (trieda *TaskMonitor*), ktorý spravuje 3 triedy typu *Task* zo vstupnej vrstvy. Trieda obsahuje niekoľko metód, ktoré sú volané po aktivácií prvkov v grafickom rozhraní a tiež niekoľko pomocných metód.

A.1.2 Vnorené triedy

Všetky triedy vnorené v hlavnej triede sú rozšírením triedy *Task* zo špecifikácie JSR-296 a tvoria základ spracovania viet v programe. Obsahujú vstupnú vrstvu a zasahujú aj do prezentačnej vrstvy. Triedu tvorí minimálne 1 metóda: *doInBackground()*, ktorá obsahuje väčšinovú funkčnosť triedy. Voliteľne sa tu môže nachádzať metóda *process()*, ktorá spracováva údaje počas behu metódy *doInBackground()*, alebo metóda *finished()*, ktorá sa vykonáva po skončení hlavnej metódy. V našom prípade sme triedy implementovali nasledovne:

```
1 // trieda na spracovanie suboru
2 private class analyzeFileTask extends Task<Object, Void> {
3     private File file;
4
5     analyzeFileTask(Application app, File file) {
6         super(app);
7         this.file = file; // subor s údajmi
8     }
9
10    protected Object doInBackground() {
11        // nacitanie, spracovanie, nastavenie premennych
12    }
13
14    protected void finished() {
15        // nastavenie grafickych prvkov podla nastavenych premennych
```

```

16  }
17  }
18
19  // trieda na spracovanie suborov na simulaciu
20  private class analyzeFileSimTask extends Task<Object, SharedData> {
21      private File file;
22
23      analyzeFileSimTask(Application app, File file) {
24          super(app);
25          this.file = file; // subor s udajmi na simulaciu
26      }
27
28      protected Object doInBackground() {
29          SharedData shared_data = new SharedData();
30          // nacitanie, spracovanie, nastavenie premennej shared_data;
31          publish(shared_data); // publikovanie premennej
32      }
33
34      protected void process(List<SharedData> list) {
35          // zber premennych, nastavovanie grafickych prvkov
36      }
37
38      protected void finished() {
39          // ukoncenie po odsimulovani celeho suboru
40      }
41  }
42
43  // trieda na citanie udajov zo serioveho portu
44  private class connectGPSTask extends Task<Object, SharedData> {
45      private InputStream istream;
46
47      connectGPSTask(Application app, InputStream istream) {
48          super(app);
49          this.istream = istream; // vstup zo serioveho portu
50      }
51
52      protected Object doInBackground() throws Exception {
53          SharedData shared_data = new SharedData();
54          // nacitanie, spracovanie, nastavenie premennej shared_data;
55          publish(shared_data); // publikovanie premennej
56      }
57
58      protected void process(List<SharedData> list) {
59          // zber premennych, nastavovanie grafickych prvkov
60      }
61  }

```

A.1.3 Metódy

Väčšinu metód v hlavnej triede tvoria metódy volané grafickými prvkami ako sú tlačidla a voľby v menu. Nachádzajú sa tu však aj ďalšie metódy určené napr. na prácu s mapou. Popíšeme si niektoré metódy z oboch kategórií:

Metóda **drawMap()**: vykresľuje trasu do mapy, volá ďalšiu metódu **drawWaypoint()** na vykreslenie orientačného bodu (waypoint) a následne aktualizuje grafický prvok, ktorý túto mapu obsahuje.

```
1 public void drawMap() {
2     Painter<JXMapView> lineOverlay = new Painter<JXMapView>() {
3         public void paint(Graphics2D g, JXMapView map, int w, int h) {
4             ...
5             java.awt.Rectangle rect = jXmkMap.getMainMap().getViewportBounds();
6             // nastavi zaciatok suradnicoveho systemu na dane koordinaty
7             g.translate(-rect.x, -rect.y);
8
9             // tu je nastavenie kreslenia - farba, vyhladzovanie, hrubka ciary
10            ...
11            for (GeoPosition gp : MapPath) { // pre vsetky ulozene body
12                // konverzia zemepisnej sirky a dlzky na suradnice x a y pre grafiku
13                Point2D pt = jXmkMap.getMainMap().getTileFactory().geoToPixel(gp,
14                    zoom);
15                if (lastX != -1 && lastY != -1) {
16                    // kreslenie ciary medzi poslednym a sucasnym bodom
17                    g.drawLine(lastX, lastY, (int) pt.getX(), (int) pt.getY());
18                }
19                lastX = (int) pt.getX();
20                lastY = (int) pt.getY();
21            }
22            ...
23            if (waypoint_gp != null) {
24                drawWaypoint(g, map); // kreslenie orientacneho bodu
25            }
26            g.dispose();
27        }
28    };
29    // tu je aktualizacia grafickyh prvkov
30 }
```

Metóda **jLstLogValueChanged()**: volá sa pri výbere vety v okne “NMEA záznam”, vytvorí inštanciu triedy **NMEAParser**, ktorá vybranú vetu spracuje. Následne podľa získaných údajov nastaví údajové modely tabuľkám v oknách “Analýza

hodnôt” a “Získané údaje” a nastaví orientačný bod, ak vybraná veta obsahovala súradnice.

```
1 private void jLstLogValueChanged(javax.swing.event.ListSelectionEvent evt) {
2     if (jLstLog.getSelectedValue() != null) {
3         // vytvorenie triedy NMEAParser s parametrom vybranej vety
4         NMEAParser parser = new NMEAParser(((String[]) jLstLog.getSelectedValue
5             ())[2]);
6         // naplnenie udajoveho modelu tabulky rozdelenou vetou
7         jTblAnalysis.setModel(parser.getParsedTM());
8         // nastavovanie sirky stlpcov a stlpca s tlacitkom napovedy
9         jTblAnalysis.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
10        DefaultTableColumnModel colModel = (DefaultTableColumnModel)
11            jTblAnalysis.getColumnModel();
12        colModel.getColumnModel(0).setPreferredWidth(175);
13        colModel.getColumnModel(1).setPreferredWidth(75);
14        colModel.getColumnModel(2).setPreferredWidth(20);
15        colModel.getColumnModel(2).setCellRenderer(new TableCellButtonRenderer());
16        colModel.getColumnModel(2).setCellEditor(new TableCellButtonEditor(new
17            JCheckBox()));
18        // naplnenie udajoveho modelu tabulky spracovanou vetou
19        jTblInfo = Tools.autoResizeColWidth(jTblInfo, parser.getAnalyzedTM());
20        NMEAData data = parser.getData(); // získanie spracovaných udajov
21        if ((data.getGeoPos() != null) && data.getGeoPosStatus()) {
22            waypoint_gp = data.getGeoPos();
23            drawMap(); // aktualizacia mapy aj s vybraným bodom (ak nejaký je)
24        }
25    }
26 }
```

Metóda **jBtnFilterActionPerformed()**: vykonáva sa po stlačení tlačidla “Filtruj” v okne “NMEA záznam”, podľa toho či je zaškrtnuté políčko “Regexp” sa vety filtrujú buď podľa začiatku, alebo podľa regulárnych výrazov. Na filtrovanie sme kvôli rýchlosti použili knižnicu TrueFilter.

```
1 private void jBtnFilterActionPerformed(ActionEvent evt) {
2     AbstractButton ab = (AbstractButton) evt.getSource();
3     // ak sa ma filtrovať ...
4     if (ab.isSelected()) {
5         ...
6         FilteredStaticListModel fmodel = new FilteredStaticListModel();
7         fmodel.setFilter(new ListElementFilter() {
8             public boolean accept(Object o) {
9                 String[] str = (String[]) o;
10                if (regex) {
11                    // filtrovanie podľa regulárnych výrazov
12                    return str[2].matches(jTxtFilter.getText()) ? true : false;
13                }
14            }
15        });
16    }
17 }
```

```

13         } else {
14             // filtrovanie podla zaciatku retazca
15             return str[2].startsWith(jTxtFilter.getText()) ? true : false;
16         }
17     }
18 });
19 fmodel.setSource(listModel);
20 jLstLog.setModel(fmodel);
21 } else { // ak sa nema filtrovat, nastav povodny udajovy model
22     jLstLog.setModel(listModel);
23 }
24 }

```

A.2 Logická vrstva *NMEAParser*

Trieda *NMEAParser* tvorí väčšinu logickej vrstvy. Obsahuje 2 hlavné metódy: metódu *parse()* a metódu *analyze()*, ktoré vetu rozdeľujú resp. spracovávajú. Okrem toho trieda obsahuje viacero pomocných metód na transformáciu údajov.

A.2.1 Metóda *parse()*

V tejto metóde sa daný reťazec identifikuje podľa začiatkových písmen. Pokiaľ sa začiatok rovná jednému z piatich typov (GPGGA, GPGSA, GPRMC, GPVTG, GPGSV), podľa identifikovaného typu sa nastaví identifikátor a podľa neho sa prijatý reťazec následne rozdeľuje. Rozdeľovanie funguje na princípe regulárnych výrazov hlavne kvôli tomu, aby sme sa vyhli nestabilite programu v prípade, keď sa pokúsi spracovať vetu, ktorá nie je generovaná podľa štandardu NMEA, ďalším dôvodom je zamedzenie spracovávania nekonzistentných údajov. Pre väčšinu viet sú regulárne výrazy dopredu definované, len pre vety typu GSV je nutné túto definíciu poskladať podľa počtu satelitov, ktoré daná veta obsahuje. Regulárne výrazy sú implementované pomocou triedy *Scanner* a jej metód *findInLine()* a *match()*. Po rozdelení vety sa jej časti pridávajú do údajového modelu spolu s popisom, ktorý je uložený v poli reťazcov. Tento model je potom vrátený metódou.

```

1 public DefaultTableModel parse() {
2     String[] str = new String[32];
3     Scanner s = new Scanner(sentence); // vytvorenie triedy Scanner s vetou
4     MatchResult result = null;
5     int index = -1;

```

```

6 DefaultTableModel tableModel = new DefaultTableModel();
7 ... // tu je priprava udajoveho modelu
8
9 if (sentence.startsWith("$GPGSA")) {
10     index = 0;
11 } else if (sentence.startsWith("$GPRMC")) {
12     index = 1;
13 } else if (sentence.startsWith("$GPVTG")) {
14     index = 2;
15 } else if (sentence.startsWith("$GPGGA")) {
16     index = 3;
17 } else if (sentence.startsWith("$GPGSV")) {
18     index = 10;
19 } else {
20     return tableModel;
21 }
22 // vety GGA, GSA, VTG a RMC
23 if (index != -1 && index != 10) {
24     // porovname vyraz s retazcom
25     s.findInLine(regexes[index]);
26     try {
27         result = s.match(); // ziskame z retazca hodnoty
28     } catch (IllegalStateException e) {
29         ...
30         // ak ide o iny typ vety RMC treba pouzit inu definiciu
31         // tu sa vykona porovnanie retazca s druhou definiciou
32     }
33     // pridavanie popisu a vkladanie do udajoveho modelu
34     str = desc[index].split(",");
35     for (int i = 0; i < str.length; i++) {
36         String[] tmp = { str[i], result.group(i + 1), str[i] };
37         tableModel.addRow(tmp);
38     }
39 } else if (index == 10) { // veta typu GSV
40     // skladanie regularneho vyrazu
41     int count = sentence.replaceAll("[^,]", "").length();
42     String regex = "\\$(\\w+),(\\d*),(\\d*),(\\d*),";
43     String descGSV = "Typ_vety,Pocet_satelitov_v_skupine,Cislo_vety_v_
44         skupine,Pocet_viditelnych_satelitov,";
45     for (int i = 0; i < ((count - 3) / 4); i++) {
46         regex += "(\\d*),(\\d*),(\\d*),(\\d*)";
47         descGSV += "Cislo_satelitu,Elevacia,Azimut,Odstup_signal-sum(SNR),";
48         if (i != ((count - 3) / 4) - 1) {
49             regex += ",";
50         }
51     }
52     regex += "\\*(\\w+)";
53     descGSV += "Kontrolny_sucet";
54     ...
55     // tu prebieha porovnavanie, ziskavanie hodnot a ich pridavanie do

```

```

55     // udajoveho modelu ako pri predchadzajucich typoch viet
56     }
57 }
58 this.hodnoty = result;
59 this.typ_vety = index;
60
61 return tableModel;
62 }

```

A.2.2 Metóda *analyze()*

Tu sa hodnoty získané metódou `parse()` spracovávajú. Spracovanie prebieha podľa typu vety, vykonávajú sa tieto operácie:

- prevod hodnoty DOP na textovú informáciu - metóda `evalDOP()`
- prevod času na čas v zrozumiteľnej forme - metóda `evalUTC()`
- prevod zemepisných koordinátou do textovej podoby - metóda `evalLatLon()`
- prevod dátumu do zrozumiteľnej podoby - metóda `evalDate()`
- prevod hodnoty kvality signálu do textovej podoby - metóda `evalQuality()`

```

1 public DefaultTableModel analyze() {
2     String[] tmp = new String[2];
3     DefaultTableModel tableModel = new DefaultTableModel();
4     ...
5     // tu je priprava udajoveho modelu
6
7     // pokial sa vetu nepodarilo rozdelit vratime prazdny udajovy model
8     if (hodnoty == null) {
9         return tableModel;
10    }
11
12    switch (typ_vety) {
13        case 0:
14            ...
15            // tu je spracovanie udajov z vety GSA
16            break;
17        case 1:
18        case 4:
19            ...
20            // tu je spracovanie udajov z vety typu RMC - obidve definicie
21            break;
22        case 2:
23            ...

```

```
24     // tu je spracovanie udajov z vety typu VTG
25     break;
26     case 3:
27         ...
28     // tu je spracovanie udajov z vety typu GGA
29     break;
30 }
31 return tableModel;
32 }
```

Za zmienku stojí ešte metóda `getData()`, ktorá vykonáva časť z toho čo metóda `analyze()`, údaje sú však pridávané do inštancie triedy `NMEADData`, ktorá je po spracovaní touto metódou vrátená. Týmto spôsobom získava prezentačná vrstva údaje.

A.3 Pomocné triedy

Počas implementácie bolo potrebné implementovať niekoľko pomocných tried, či už určených na komunikáciu medzi vrstvami, zber údajov v prezentačnej vrstve, úpravu grafických prvkov, alebo jednoducho na sprehľadnenie zdrojového kódu.

A.3.1 *NMEADData*

Táto trieda predstavuje údajovú jednotku, používanú pri prenose údajov z logickej vrstvy do vrstvy prezentačnej. Obsahuje údaje o čase, dátume, zemepisných súradniciach, nadmorskej výške, rýchlosti a platnosti údajov. Je v nej tiež zoznam používaných satelitov a zoznam tvorený ďalšou triedou `SatInfo`, ktorý združuje informácie o každom viditeľnom satelite.

A.3.2 *SatInfo*

Trieda `SatInfo` združuje informácie o satelite, teda jeho identifikačné číslo PRN, kvalitu signálu, azimut a eleváciu. Tieto údaje sú združované pre všetky viditeľné satelity v triede `NMEADData`, a sú používané pri vykresľovaní sily signálu z jednotlivých používaných satelitov ako aj pri vykresľovaní pozície jednotlivých satelitov na oblohe.

A.3.3 *SerialPortTools*

Metódy v tejto triede slúžia na výber, konfiguráciu a pripojenie sa na sériový port. Trieda používa metódy knižnice RXTX, po pripojení vráti metóda connect() inštanciu triedy InputStream, s ktorou potom pracuje vstupná vrstva.

A.3.4 *SharedData*

Táto trieda slúži na komunikáciu vlákna vytváraného vo vstupnej vrstve so zvyškom programu. Obsahuje zemepisné údaje a údaje o nadmorskej výške. Čítacie vlákno tieto údaje nastaví a pošle ich pomocnej funkcií, ktorá podľa nich potom nastaví jednotlivé grafické prvky - mapu a graf nadmorskej výšky. Iné údaje nie je potrebné zdieľať takýmto spôsobom, pretože grafické prvky, v ktorých sú tieto údaje využívané majú komunikáciu s ostatnými vláknami vyriešenú zavedením triedy Listener.

A.3.5 *ActualData*

Trieda ActualData obsahuje informácie o aktuálnych údajoch, teda dĺžke trasy, čase cesty a prevýšení. Dané údaje sú spracovávané prezentačnou vrstvou, a sú zobrazované v okne “Štatistiky a aktuálne údaje”.

A.3.6 *Tools*

Obsahom tejto triedy sú všetky zvyšné pomocné triedy, pre ktoré sa neoplatilo vytvárať vlastný súbor. Ide napríklad o modifikáciu vykresľovača (renderera) buniek v tabuľke v okne “Analýza údajov”, ktorá je využívaná pri vyvolávaní pomocníka pre danú časť vety. Ďalej je tu modifikácia zoznamu viet v okne “NMEA záznam”, ktorý sme potrebovali upraviť tak, aby obsahoval 3 stĺpce. Potom sa tu nachádza aj metóda na automatické nastavenie šírky stĺpcov v tabuľke a metóda na prevod stupňov na radiány - tá sa využíva pri kreslení mapy satelitov.

A.4 GUI triedy

Okrem skôr spomínaných tried, ktoré upravovali grafické prvky v aplikácií sme museli vykonať ďalšie, zložitejšie úpravy niektorých komponentov tak, aby vyhovovali našim požiadavkám. V tejto časti si ich popíšeme.

A.4.1 *MDIDesktopPane*

Hlavným dôvodom na modifikáciu triedy DesktopPane bola požiadavka na vytvorenie plochy, ktorá by neobmedzovala veľkosť a počet okien, ktoré sa v nej dajú zobraziť. Pôvodná implementácia totiž pri posune okna mimo plochu nezobrazuje posúvače, ktorými by sa dalo plochu posúvať. Modifikovali sme teda triedu DesktopPane podľa článku uverejneného na stránke JavaWorld, čím sme docielili spomínanú funkcionálnosť.

A.4.2 *InternalFrameProperty* a *InternalFrameState*

Tieto triedy sme potrebovali implementovať na to, aby sa do súboru stavu programu dal ukladať stav vnorených okien, teda ich pozícia a veľkosť. Ukladanie do súboru je riešené implementáciou špecifikácie JSR-296. Tieto vlastnosti sme využili pri rozmiestňovaní vnorených okien po ploche pri voľbe režimu.

A.4.3 *WindowMenu*

V tejto triede implementujeme položku v menu, ktorá obsahuje výpis všetkých vnorených okien, ktoré sa dajú na plochu otvoriť. Trieda bola vytvorená, podobne ako trieda MDIDesktopPane, podľa článku uverejnenom na stránke JavaWorld.

A.5 *Triedy mapy satelitov*

Najväčšie úpravy si vyžadovalo pozmenenie klasického polárneho grafu z knižnice JFreeChart tak, aby dokázal zobrazovať pozície satelitov na oblohe. Dosiahli sme to rozšírením údajových tried XYSeries, XYDataItem a grafickej triedy DefaultPolarItemRenderer.

A.5.1 *NMEAxyDataItem*

V tejto triede implementujeme prvok série, ktorá je pridávaná do grafu. Obsahuje identifikačné číslo satelitu, jeho stav (či je používaný, viditeľný, alebo má slabý signál), eleváciu a azimut. Pomocou tejto triedy sa vytvára trieda série.

A.5.2 *NMEAXYSeries*

Implementuje sa tu sériu prvkov, ktoré sme definovali v triede *NMEAXYDataItem*. Hodnoty z tejto série sú ďalej použité pri vykresľovaní polárneho grafu.

A.5.3 *NMEAItemRenderer*

Samotná kresliaca trieda, v ktorej je definované ako sa majú vykresľovať jednotlivé prvky v grafe. Satelity sú vykreslené ako farebné bodky v grafe, farby bodiek indikuje ich stav, pri bodke je napísané aj číslo satelitu.

B Používateľská príručka

B.1 Funkcia programu

Program ‘‘NMEA Analyzátor’’ slúži na spracovanie viet protokolu NMEA-0183. Vety dokáže spracovávať zo záznamu, aj z reálneho GPS zariadenia, pokiaľ je k počítaču pripájané cez sériový port, alebo obsahuje softvér vytvárajúci virtuálny sériový port. Program dokáže spracované vety zobrazíť v grafickej forme v podobe grafov a mapy, aj vo forme textovej v podobe analyzovaných častí jednotlivých viet, pričom ku každej časti poskytuje aj pomocníka.

B.2 Hardvérová konfigurácia

Na plynulú prácu s programom je odporúčaná nasledovná konfigurácia:

- dvojjadrový procesor s frekvenciou minimálne 1,5 GHz,
- 1 GB pamäte RAM,
- pevný disk s 10 MB pamäte (len aplikácia a knižnice),
- klávesnica a myš,
- internetové pripojenie (na zobrazenie máp)
- sériový port / USB port (na pripojenie GPS zariadenia)

B.3 Požiadavky na programové prostriedky

Na inštaláciu a spustenie programu sú potrebné tieto programové prostriedky:

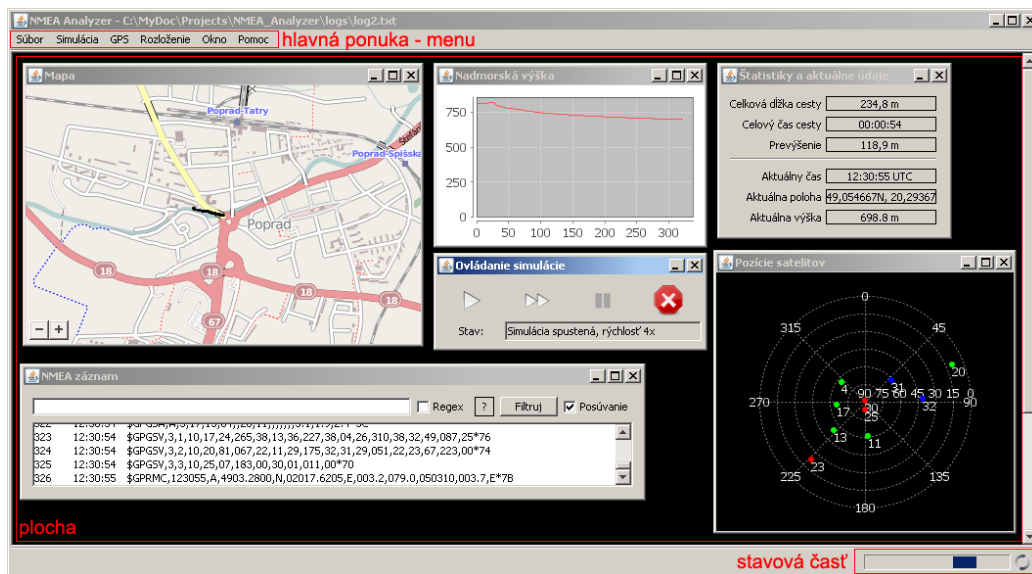
- operačný systém Windows
- nainštalovaný Java Runtime Environment vo verzií minimálne 1.6
- v prípade zariadenia s USB pripojením - program na vytvorenie virtuálneho sériového portu pre dané zariadenie

B.4 Inštalácia a spustenie

Inštalácia spočíva v skopírovaní adresára NMEA_Analyzator so všetkými súbormi, ktoré obsahuje, na pevný disk. Na spustenie programu stačí spustiť súbor NMEA_Analyzator.exe.

B.5 Opis prvkov GUI

Po spustení programu sa zobrazí voľba rozloženia okien - Výučbové, Pracovné, alebo Prázdne. Táto voľba zvyšuje prehľadnosť pri práci s programom. Okná, ktoré vybraný režim neukáže je možné neskôr dodatočne zobraziť a nastaviť. Po výbere rozloženia sa ukáže hlavné okno programu. Tvoria ho 3 časti: hlavná ponuka (menu), plocha a stavová časť.



Obr. 1: Hlavné okno programu

B.5.1 Hlavná ponuka - menu

Hlavná ponuka obsahuje 6 položiek:

- Súbor: voľby Otvoriť - otvorí dialógové okno na výber súboru, Ukončiť - ukončí program
- Simulácia: Otvoriť na simuláciu - otvorí dialógové okno na výber súboru, Ukončiť simuláciu - ukončí simuláciu
- GPS: Pripojiť ku GPS - otvorí dialógové okno na výber sériového portu, Odpojiť - odpojí program od sériového portu
- Rozloženie: Výučbové a Pracovné - nastaví rozloženie a veľkosti okien, Vlastné - načíta uložené rozloženie, Uložiť rozloženie - uloží rozloženie okien do súboru

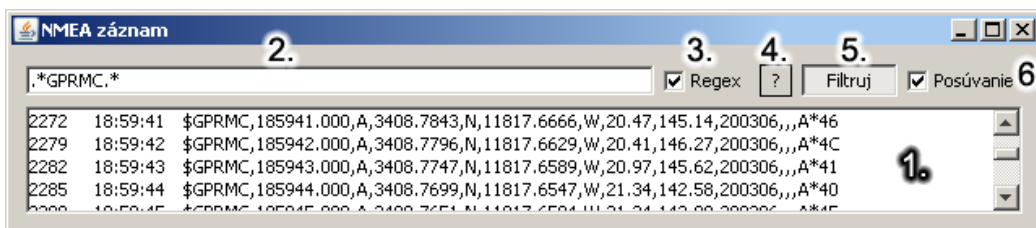
- Okno: obsahuje zoznam okien, po výbere sa dané okno zobrazí
- Pomoc: Pomocník - vyvolá pomocníka k programu, O programe - zobrazí informácie o autorovi

B.5.2 Plocha

Plocha predstavuje miesto, v ktorom sú zobrazované vnorené okná. Tie je možné po ploche posúvať, meniť ich veľkosť, minimalizovať, maximalizovať a zatvárať ich. Celkovo existuje 9 druhov okien, ktoré je možné na ploche zobrazíť.

NMEA záznam

Okno obsahuje zoznam prečítaných NMEA viet spolu s časom a poradovým číslom vety (1.). Ďalej sa tu nachádza textové pole, do ktorého je možné zadať filtrovací výraz (2.). Vety môžu byť filtrované podľa začiatku, alebo, pokiaľ je zaškrtnuté pole "Regex" (3.) je filtrovanie vykonávané podľa zadaného regulárneho výrazu. Pomoc pri zadávaní regulárnych výrazov je možné získať po stlačení tlačidla (4.). Filter sa na zoznam viet aplikuje po stlačení tlačidla "Filtruj" (5.), po opätovnom stlačení sa filtrovanie zruší. Zaškrtnutím poľa "Posúvanie" (6.) je zoznam viet pri režime simulácie a pri práci s reálnym zariadením posúvaný na koniec, inak sa pozícia nemení.



Obr. 2: Okno "NMEA záznam"

Analýza hodnôt

Tabuľka v tomto okne sa zobrazuje po výbere vety zo zoznamu v okne "NMEA záznam". Vybraná veta je v tejto tabuľke rozdelená a ku každej jej časti (2.) je zobrazený popis (1.) a tlačidlo (3.), ktorým sa dá vyvolať pomocník k časti vety.

Údaj	Hodnota	?
Typ vety	GPRMC	?
Univerzálny čas UTC	185944.000	?
Stav údajov	A	?
Zemepisná šírka	3408.7699	?
Severne/Južne	N	?
Zemepisná dĺžka	11817.6547	?
Východne/Západne	W	?
Priemerná rýchlosť v uzloch	21.34	?
Skutočná dráha v stupňoch	142.58	?
Dátum	200306	?
Magnetická odchýlka v °		?
Východne/Západne		?
Ukazovateľ režimu FAA	A	?
Kontrolný súčet	40	?

Obr. 3: Okno "Analýza hodnôt"

Získané údaje

Podobne ako pri predchádzajúcom okne je tu tabuľka, ktorá sa zobrazí po výbere vety zo zoznamu. V tabuľke sú zobrazené údaje z vety po spracovaní (2.) tiež s popisom (1.).

Údaj	Hodnota
UTC čas a dátum	18:59:44, 20.3.2006
Zemepisná šírka	34° 8' 46,194" N (34,146165 N)
Zemepisná dĺžka	118° 17' 39,282" W (118,294245 ...)
Rýchlosť	21.34 uzlov
Kurz	142.58°

Obr. 4: Okno "Získané údaje"

Mapa

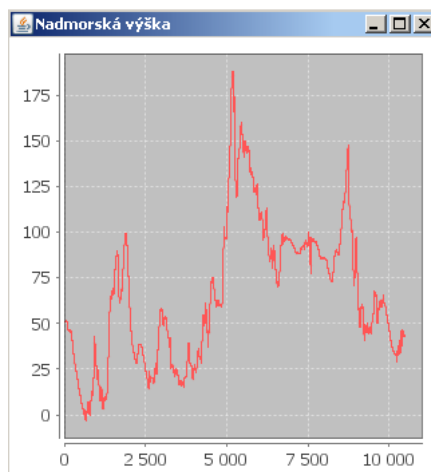
V tomto okne sa nachádza mapa zobrazujúca trasu získanú z údajov zo spracovaných viet (1.) a orientačný bod (2.), ktorý sa zobrazuje po výbere vety zo zoznamu. Vybraná veta však musí obsahovať údaje o zemepisnej šírke a dĺžke. Mapa sa dá posúvať pomocou myši, približovať a oddiaľovať je možné pomocou kolieska na myši, alebo pomocou tlačidiel (3.).



Obr. 5: Okno "Mapa"

Nadmorská výška

V okne sa zobrazuje graf nadmorskej výšky. Os X tvorí poradové číslo vety v zozname, os Y je potom samotná hodnota nadmorskej výšky. Graf je možné pomocou myši približovať a oddiaľovať, po ukázaní myšou na určitú časť grafu sa zobrazia presné hodnoty X a Y.

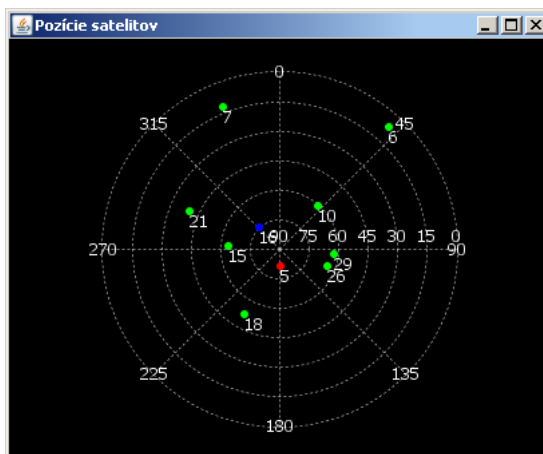


Obr. 6: Okno "Nadmorská výška"

Pozície satelitov

Toto okno zobrazuje pozíciu satelitov. Tá je vyjadrená pomocou azimutu (0° až 359°) a elevácie (0° až 90° , kde 90° je horizont). Satelity sú farebne odlišené, zelená

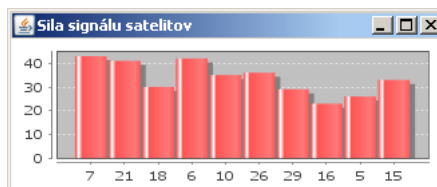
farba indikuje používané satelity, modrá farba viditeľné satelity s dobrým signálom a červená farba viditeľné satelity so slabým signálom. Všetky satelity sú rozlíšené pomocou ich identifikačných čísel PRN.



Obr. 7: Okno "Pozície satelitov"

Sila signálu satelitov

V tomto okne sú zobrazované hodnoty sily signálu pre jednotlivé používané satelity vo forme stĺpcového grafu. Sila signálu môžu nadobúdať hodnoty od 0 po 99 dB. Počet stĺpcov v grafe závisí na počte používaných satelitov. Presná hodnota sily signálu sa zobrazí po ukázaní myšou na vybraný stĺpec. Os X tvoria identifikačné čísla používaných satelitov.

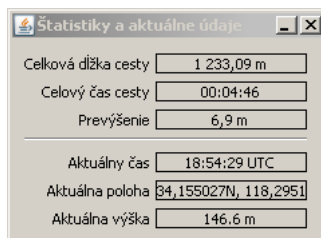


Obr. 8: Okno "Sila signálu satelitov"

Štatistiky a aktuálne údaje

Ako už názov okna napovedá, sú tu zobrazené aktuálne a štatistické údaje. Zo štatistických údajov ide o celkovú dĺžku cesty (v metroch) meranú od prvej vety, celkový čas cesty vo formáte HH:MM:SS a prevýšenie. Aktuálne údaje sú zobra-

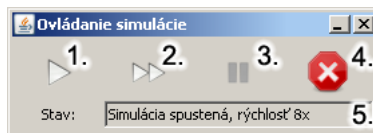
zované len pri simulácií a pri práci s reálnym zariadením. Zahŕňajú aktuálny čas, aktuálnu polohu a výšku.



Obr. 9: Okno “Štatistiky a aktuálne údaje”

Ovládanie simulácie

Toto okno je aktívne len v režime simulácie, obsahuje prvky slúžiace na jej ovládanie. Tlačidlom 1. sa celá simulácia spúšťa, prípadne sa obnovuje ak je pozastavená. Tlačidlom 2. sa zvyšuje rýchlosť simulácie, dá sa nastaviť na hodnoty 1x až 8x skutočnej rýchlosti. Po prekročení maximálnej hodnoty sa rýchlosť opäť nastaví na 1x. Tlačidlom 3. sa simulácia môže pozastaviť, obnoví sa stlačením tlačidla 1. Tlačidlom 5. sa režim simulácie zruší. Textové pole “Stav” (5.) zobrazuje aktuálny stav simulácie a nastavenú rýchlosť.



Obr. 10: Okno “Ovládanie simulácie”

B.5.3 Stavová časť

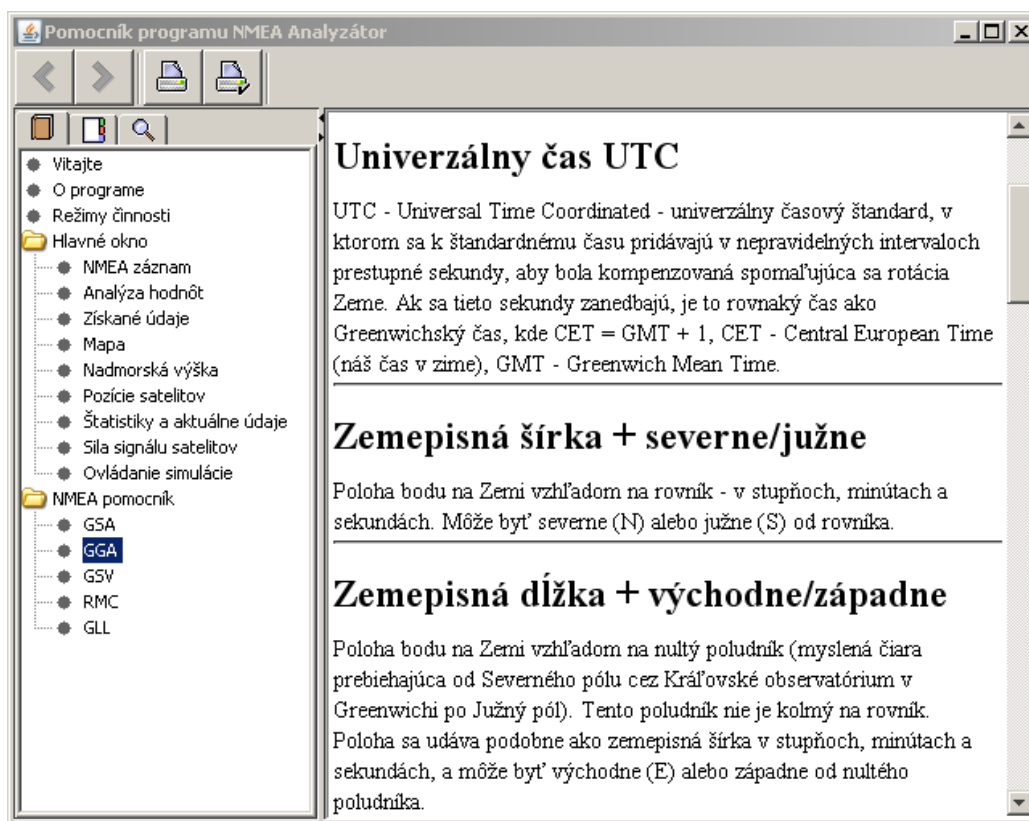
V stavovej časti sa nachádza graf činnosti, ktorý sa zobrazuje ak program vykonáva niečo na pozadí, konkrétne načítavanie, simuláciu alebo prijímanie údajov z GPS zariadenia.

B.5.4 Pomocník

Pomocník sa dá vyvolať z programu tromi spôsobmi:

1. z hlavnej ponuky výberom voľby Pomocník v položke Pomoc - zobrazí sa okno Pomocníka
2. kliknutím na tlačidlo s otáznikom v okne "NMEA záznam" - zobrazí sa Pomocník s informáciami o regulárnych výrazoch
3. pri analýze viet kliknutím na tlačidlo s otáznikom vedľa ľubovoľnej časti vety - zobrazí sa Pomocník s informáciami o danej časti vety

Pomocník sa ovláda primárne myšou, okrem kontextovej pomoci obsahuje aj možnosť vyhľadávania.



Obr. 11: Okno Pomocníka

C Obsah CD média

Na priloženom CD médiu sa nachádza:

- tento dokument
- zdrojový kód programu
- spustiteľná verzia programu
- testovacie súbory s NMEA vetami