

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Vizualizace procesu optimalizace vsázek pro zóny jaderných reaktorů

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 5. května 2015

Josef Zeman

Abstract

This work is targeted at creating and describing a program that visualizes data obtained during the fuel load optimization process for nuclear reactor zones, facilitating thus the determination of limit values. This program is a flexible tool for presenting overviews of data gathered during the fuel load optimization process. It offers an insight into the methods of visualization used in this program, and into the creation and structure of the program itself, which, it is believed, will enable the reader to operate the program.

Abstrakt

Tato práce se zaměřuje na tvorbu a popis programu, který vizualizuje data získaná při procesu optimalizace vsázek pro zóny jaderných reaktorů, a tím usnadňuje nalezení limitních hodnot. Tento program je flexibilní nástroj pro zobrazování přehledů dat získaných při procesu optimalizace vsázek. Po přečtení by čtenář měl mít přehled o způsobech zobrazování dat použitých v tomto programu. Dále by měl čtenář získat povědomí o struktuře a způsobu výroby tohoto programu a měl by být schopen s tímto programem pracovat.

Obsah

1	Úvod	1
2	Návrh programu	2
2.1	Cíl	2
2.2	Požadované vlastnosti	2
3	Návrh řešení	3
3.1	Použité technologie	3
3.1.1	Java	3
3.1.2	AWT	3
3.1.3	Swing	4
3.1.4	JFreeChart	4
3.1.5	XML	4
4	Realizace	6
4.1	Struktura a formát dat	6
4.1.1	Gafuel	6
4.1.2	Current	7
4.1.3	Layout	8
4.1.4	Soubor projektu	9
4.1.5	Soubor relace	9
4.2	Načtení dat	10
4.3	Načtení projektu	12
4.4	Načtení relace	12
4.5	Uložení relace	14
4.6	Lokalizace	15
4.7	Komponenty programu	17
4.7.1	Hlavní okno	17
4.7.2	Výběrové okno lokalizace	18
4.7.3	Výběrové okno proměnné „ <i>SelectionRange</i> “	20
4.7.4	Výběrové okno rozměrů obrázku	21

4.7.5	Dialogové okno načítání a ukládání	21
4.7.6	Menu	22
4.7.7	Graf	26
4.7.8	Seznam bodů	28
4.7.9	Interní plocha	29
4.7.10	Interní okna	31
5	Závěr	36
A	Uživatelská příručka	39
A.1	Návod na spuštění	39
A.2	Načtení dat	39
A.3	Změna nastavení	40
A.4	Ovládání komponentů	41

1 Úvod

Cílem této práce je vytvořit program s grafickým uživatelským rozhraním, usnadňující proces optimalizace vsázek vizualizací zón jaderných reaktorů. Toto rozhraní je rozvrženo přehledně tak, aby vyhovovalo jeho uživatelům, a bylo vytvořeno po konzultaci s jeho uživateli. Přehled zón jaderných reaktorů je vytvořen způsobem, na který jsou cíloví uživatelé zvyklí. Umožňuje v základu pracovat s dvěma nejvíce používanými typy reaktorů. Program podporuje načtení rozložení reaktoru z externího souboru. Díky tomu lze v budoucnu seznam podporovaných typů reaktorů snadno rozšířit o všechny typy, používající programem známé tvary kazet. Vytvoření tohoto programu bylo žádoucí z důvodu absence komplexního a univerzálního nástroje pro vizualizaci tohoto problému, který umožňuje práci s více druhy reaktorů stejným způsobem.

Po přečtení této práce by měl čtenář získat komplexní přehled o ovládání a používání programu a způsobu, jakým byl tento program vytvořen. Tato práce představuje náhled do konstrukce jednotlivých částí tohoto programu a do řešení jednotlivých problémů zpracovávaných při tvorbě tohoto programu.

V první části této práce je popsán návrh řešení a cíle daného programu. Jsou zde popsány použité technologie. Dále následuje popis praktického řešení jednotlivých částí tohoto navrhnutého programu pomocí zvolených popsaných technologií.

2 Návrh programu

2.1 Cíl

Cílem je vyrobit program, který usnadní práci a přehledně zobrazí „zajímavé“ hodnoty, určené k dalšímu zkoumání pomocí výpočtů. Zobrazuje vsázky reaktoru a jejich několik základních vlastností. Při procesu optimalizace vsázek je nutné vyloučit vsázky překračující stanovený limit a naopak je možné některé vsázky využít znovu, pokud pro to splňují, všemi svými vlastnostmi, požadované předpoklady. Právě tyto vsázky by mělo být možné pomocí tohoto programu snadno odhalit.

2.2 Požadované vlastnosti

V programu je možné u všech otevřených oken, zobrazující části reaktoru, zvolit možnost, podle kterých vlastností se mají zobrazené komponenty obarvovat. Program umí načítat i ukládat aktuální rozvržení okna a komponentů do souboru projektu. Zobrazuje data získané rozparsováním datových souborů. Obsahuje přibližitelný graf, zobrazující načtená data, tedy jednotlivé výpočty s různými kombinacemi umístění vsázek. Je možné jakýkoliv bod na grafu označit a otevřít tím okno s podrobnějšími vizualizovanými daty, kterými je právě jeden výpočet s určitým rozložením vsázek. Pro ulehčení označení jednoho bodu v grafu je zde vypsán seznam všech bodů v nejbližším okolí označeného místa v grafu. Až přes tento seznam bodů se tedy ve výsledku označují jednotlivé body a otvírají jejich příslušná detailní zobrazení. Velikost tohoto okolí, které určuje, jaké body se vypíšou v seznamu, je nastavitelné v menu programu. Program podporuje více jazyků a v menu lze změnit lokalizaci. Všechno nastavení v menu se uloží do souboru po uložení relace.

3 Návrh řešení

3.1 Použité technologie

3.1.1 Java

„Java“ je programovací jazyk a výpočetní platforma pro aplikace běžící na většině operačních systémů, včetně operačních systémů na mobilních zařízeních. Dále je možné aplikaci, psanou v programovacím jazyku „Java“, integrovat do webové stránky. Programovací jazyk „Java“ byl pro tento program vybrán právě především z důvodu kompatibility a přenositelnosti na širokou škálu různých zařízení. Výkonnostně tento jazyk mírně zaostává oproti nižším programovacím jazykům, jako je například „ANSI C“, avšak pro potřeby vizualizace v tomto programu je jeho rychlost naprosto dostačující. „Java“ byla vytvořena v roce 1991 malou skupinkou inženýrů společnosti „Sun Microsystems“. Tato skupinka si říkala „Green Team“ a vedl je James Gosling. „Green Team“ demonstroval jejich nový programovací jazyk na ručním interaktivním ovladači. Jejich snažení bylo původně cíleno na trh digitální kabelové televize. V roce 1995 tým oznámil, že internetový prohlížeč „Netscape Navigator“ do sebe začlení „Java“ technologii. Dnes již „Java“ není použita pouze v oblasti internetu, ale stojí za mnoha aplikacemi, navigačními systémy a hrami na spoustách zařízení používaných v každodenním životě. [5, 10, 8].

3.1.2 AWT

„AWT“, celým názvem „Abstract Windowing Toolkit“, je knihovna tříd psaných pro programovací jazyk „Java“. Tato knihovna umožňuje a usnadňuje vytváření prvků grafického uživatelského rozhraní. Práce s ní je snadná a všechny její třídy a metody jsou plně zdokumentovány. Tyto detailně popsané dokumentace jsou volně k dispozici, viz [1]. „AWT“ poskytuje dobře navrhnuté objektové orientované rozhraní pro nízkoúrovňové služby a zdroje. Jelikož je programovací jazyk „Java“ platformově nezávislý musí být nutně i „AWT“ knihovna platformově nezávislá. „AWT“ byl navržen na poskytování běžné sady nástrojů pro návrh grafického uživatelského rozhraní, který funguje na rozmanitých platformách. Prvky uživatelského rozhraní poskytované knihovnou „AWT“ jsou implementovány pomocí nativní sady nástrojů gra-

fického uživatelského rozhraní každé platformy, přičemž zachovávají vzhled každé platformy. Nevýhodou takového přístupu je fakt, že grafické uživatelské rozhraní navržené na jedné platformě, může po zobrazení vypadat odlišně na jiných platformách [3]. Tato knihovna je v programu použita pouze z části a to z důvodu nutnosti použití některých implementací poskytnutých touto knihovnou. Knihovna „AWT“ je v programu použita ve spolupráci s knihovnou „Swing“, která je popsána níže v sekci 3.1.3 s názvem „Swing“.

3.1.3 Swing

„Swing“ je oficiální framework vytvořen pro programovací jazyk „Java“ a je používán pro tvorbu grafického uživatelského rozhraní. Byl vydán společností „Sun Microsystems“. Jedná se o užitečný a schopný nástroj pro práci s grafickými komponenty jakými jdou například tlačítka, tabulky, stromy atd. Knihovna „AWT“, která je popsána výše, je neoddělitelnou součástí knihovny „Swing“, proto jsou při tvorbě tohoto programu obě knihovny použity ve vzájemné spolupráci k dosažení požadovaného vzhledu grafického uživatelského rozhraní [4, 9].

3.1.4 JFreeChart

„JFreeChart“ je knihovna, která je napsána pro programovací jazyk „Java“. Tato knihovna je svobodný software a umožňuje vytváření, zobrazování a práci s grafy různých typů. Tuto knihovnu je možné použít jak při vytváření klientské aplikace, tak při vytváření serverové aplikace. Projekt „JFreeChart“ byl založen v únoru roku 2000 Davidem Gilbertem. Dnes je „JFreeChart“ široce používaná knihovna pro „Javu“ [7].

3.1.5 XML

„XML“, celým názvem „Extensible Markup Language“, je značkovací jazyk. Jde o velmi flexibilní a široce používaný textový formát. Data ve formátu „XML“ popisují a definují sami sebe v tom smyslu, že struktura dat je obsažená v datech samotných. Stavební bloky dokumentu jsou prvky definovány značkami neboli „tagy“. Každý tento prvek má počáteční a konečnou značku. Všechny prvky v „XML“ dokumentu jsou obsaženy ve vnějším prvku zná-

mém jako „*kořenový*“. Jednotlivé prvky je možné do sebe vnořovat a tvořit tím hierarchickou strukturu. Jména jednotlivých prvků popisují obsah prvků a struktura udává vztahy mezi prvky [11, 13, 12].

4 Realizace

4.1 Struktura a formát dat

Program při své činnosti využívá několik typů souborů, které jsou zde popsány.

1	2082R	1.0334	31230.0	31230.0	0	3	SUPQ49G2	0	0
2	1002	1.2422	0.0	0.0	0	0	SUPR49G2	0	0
3	3003	1.2938	0.0	0.0	0	0	SUPZ49G6	0	0
4	5004	1.2902	0.0	0.0	0	0	SUPZ49E6	0	0
5	7005	1.2338	0.0	0.0	0	0	SUPZ47G2	0	0
6	8006	1.2148	0.0	0.0	0	0	SUPZ44G2	0	0
7	9007	1.2721	0.0	0.0	0	0	SUPZ44G6	0	0
8	9008	1.2721	0.0	0.0	0	0	SUPZ44G6	0	0
9	2124R	0.9291	46000.0	46000.0	0	4	SUPP47G8	0	0
10	2112R	0.9343	45090.0	45090.0	0	4	SUPP47G8	0	0
11	2099R	0.9368	44650.0	44650.0	0	4	SUPP47G8	0	0
12	2098R	0.9385	41980.0	41980.0	0	4	SUPP44G6	0	0
13	2085R	0.9426	41300.0	41300.0	0	4	SUPP44G6	0	0
14	2137R	0.9468	42990.0	42990.0	0	4	SUPP47G8	0	0
15	2148R	0.9670	39930.0	39930.0	0	3	SUPP47G2	0	0
16	2097R	1.0092	32440.0	32440.0	0	3	SUPQ46G2	0	0
17	2126R	1.0125	31930.0	31930.0	0	3	SUPQ46G2	0	0
18	2114R	1.0249	32540.0	32540.0	0	3	SUPQ49G2	0	0
19	2101R	1.0296	31820.0	31820.0	0	3	SUPQ49G2	0	0
20	2088R	1.0423	30230.0	30230.0	0	3	SUPQ49G6	0	0
21	2147R	1.0800	24390.0	24390.0	0	2	SUPQ49G2	0	0
22	6016R	1.1030	17560.0	17560.0	0	1	SUPZ44G2	0	0
23	4024R	1.1363	17360.0	17360.0	0	1	SUPZ49E6	0	0
24	4054R	1.1372	17250.0	17250.0	0	1	SUPZ49E6	0	0
25	4034R	1.1373	17240.0	17240.0	0	1	SUPZ49E6	0	0
26	4044R	1.1381	17140.0	17140.0	0	1	SUPZ49E6	0	0
27	4064R	1.1381	17140.0	17140.0	0	1	SUPZ49E6	0	0
28	4014R	1.1536	15190.0	15190.0	0	1	SUPZ49E6	0	0

Listing 4.1: Ukázka relačního souboru

4.1.1 Gafuel

Jedním z těchto typů souborů je soubor typu „*gafuel*“. Tento typ souboru obsahuje informace o jednotlivých palivových kazetách a jejich vlastnostech.

Data v tomto souboru jsou uložena v čistě textové podobě. Každá kazeta je uložena na samostatném řádku a každý sloupec reprezentuje jednu vlastnost dané kazety. První sloupec určuje pořadové číslo kazety. Druhý sloupec je název kazety, poté následuje hodnota K-nekonečno. Čtvrtý sloupec udává úroveň vyhoření a v sedmém sloupci je uložen věk kazety. Ostatní sloupce jsou pro tento program nepodstatné a nejsou nijak využity. Ukázkou obsahu tohoto souboru je možné vidět na výpisu 4.1

4.1.2 Current

Tak, jako v souboru typu „*gafuel*“, jsou i zde, v souboru typu „*current*“, uloženy údaje v textové podobě a mají podobný formát. Každá řádka tohoto souboru označuje jednu vsázku s určitým rozložením kazet. V každém tomto řádku jsou tedy hodnoty výsledků výpočtů pro danou konfiguraci kazet. Tento soubor umožňuje vkládání řádkových komentářů pomocí vložení znaku středníku „;“ na začátek řádku.

V prvním sloupci tohoto souboru je uloženo číslo řádky (řádky s řádkovým komentářem nejsou do tohoto čísla započítávány). Pátý sloupec určuje délku cyklu. V sedmém řádku je hodnota KR a za ním následuje reálná hodnota MTC. Na každé řádce je vždy znak „*“. Za tímto znakem následuje vektor čísel kazet. Tento vektor je seřazen v pořadí, v jakém jsou kazety pro výpočet v reaktoru umístěny. Ostatní sloupce jsou z hlediska programu neúčinné. Tento soubor zpravidla obsahuje velké množství řádků, v řádech tisíců. Ukázkou jednoho řádku z tohoto souboru je možné vidět na výpisu 4.2. Jelikož řádky tohoto souboru jsou dlouhé, je řádek na výpisu 4.2 zalomen a je pouze zobrazen na několika řádcích, přesto se však jedná o jediný fyzický řádek uložený v souboru typu „*current*“.

```

0      39 654361 4 297.0920  1.6140 1.7120 0.000e+00 53 26 32 58 31 23 54
 24  9 57 33 36  1 28 38 42  5 10 12 37 46 29  3 40 47  7 14  2 30 59
34 49 55 21 27  4 35 25 51 45 16 39 41 50 52 20 18 43 48 56 44 19  6
 8 22 17 11 15 13 *  1  3  0  0  0  3  0  3  0  3  0  0  0  3  0  0  0
  0  3  0  0  0  0  0  0  0  3  0  0  0  0  0  0  3  3  0  0  3  0  0
 3  0  0  0  0  3  3  0  0  0  0  3  0  0  3  3  0  3  3

```

Listing 4.2: Ukázkou relačního souboru

4.1.3 Layout

Soubor typu „*layout*“ obsahuje informace o mapování šestiúhelných tlačítek na pomyslnou virtuální mřížku. Také obsahuje důležité informace o této mřížce. První číslo v tomto souboru je počet všech pozic v mřížce. Na další řádce jsou dvě čísla oddělená jedním znakem mezery. Tyto čísla znamenají počet pozic v mřížce na osách X a Y. Následně, na další řádce, následuje seznam pozic. Každá pozice je zapsána zvlášť na jedné řádce a je tvořena třemi čísly oddělenými jedním znakem mezery. Prvním z nich je číslo celé a nezáporné, reprezentuje pořadí pozice a je indexováno od čísla 1. Následující dvě čísla na řádce určují souřadnice pozice na ose X a Y. Tyto čísla jsou normalizována a číslo „1“ v těchto souřadnicích znamená velikost jednoho šestiúhelného tlačítka. Obě tato čísla jsou desetinná a jsou ukládána s přesností na jedno desetinné místo, jelikož tato přesnost je naprosto postačující pro mapování tlačítek na mřížku. Do tohoto souboru je také možné vkládat řádkové komentáře. Při vložení znaku „#“ na začátek řádky, je tato řádka považována za řádkový komentář a je při načítání programem ignorována. Taktéž je možné do souboru vložit prázdnou řádku, kterou program stejným způsobem přeskočí. Formát tohoto souboru je ukázán na výpisu 4.3

```
#count of all positions
28

#xCount and yCount
7 7

#position number, X, Y
1 0.0 6.0
2 1.0 6.0
3 2.0 6.0
4 3.0 6.0
5 4.0 6.0
6 5.0 6.0
7 6.0 6.0
8 1.5 5.0
9 2.5 5.0
10 3.5 5.0
11 4.5 5.0
12 5.5 5.0
13 6.5 5.0
14 2.0 4.0
15 3.0 4.0
16 4.0 4.0
17 5.0 4.0
18 6.0 4.0
19 2.5 3.0
20 3.5 3.0
21 4.5 3.0
22 5.5 3.0
23 3.0 2.0
24 4.0 2.0
25 5.0 2.0
26 3.5 1.0
27 4.5 1.0
28 4.0 0.0
```

Listing 4.3: Ukázka relačního souboru

4.1.4 Soubor projektu

Soubor projektu, viz výpis 4.4, je soubor umožňující vytvoření nového projektu a načtení veškerých potřebných dat pro vizualizaci z příslušných souborů. Tento soubor byl navržen tak, aby byl co nejjednodušší. Umožňuje vytváření komentářů pomocí znaku „%“. Vše za tímto znakem je považováno za komentář a není bráno programem, při vytváření projektu, v potaz. V tomto souboru se vyskytuje několik předdefinovaných názvů označujících identifikátory důležitých souborů. Ke každému tomuto identifikátoru je přiřazen skutečný název daného souboru. Formát tohoto souboru je takový, že na každém řádku je nejprve uveden identifikátor a za ním je uveden název souboru, oddělený znakem mezery.

```
% projekt pro EDU
SOLUTIONS_CURRENT current_EDU.out
FUEL_LIBRARY gafuel_EDU.lib
LAYOUT EDU_layout.txt
```

Listing 4.4: Ukázka projektového souboru

4.1.5 Soubor relace

Soubor relace slouží k uchování nastavení i rozložení komponentů v programu. Celý soubor je ve formátu „XML“. Tento soubor je rozdělen do několika sekcí. První ze sekcí je „*ProjectFiles*“. Tato sekce ukládá seznam datových souborů potřebných pro vizualizaci. Další sekcí je „*SelectionRange*“, do které se ukládá vertikální a horizontální vzdálenost, ve které jsou v grafu vybrány body od pozice kurzoru po kliknutí na graf. Třetí sekcí je sekce „*ImageSize*“, která ukládá nastavení výšky i šířky obrázku, jež bude vytvořen při ukládání vzhledu vsázky zobrazené některým z interních oken. Čtvrtou ze sekcí je „*ImageDirectory*“. Tato sekce ukládá umístění adresáře v souborovém systému – jedná se o adresář, do kterého naposledy uživatel zvolil ukládání obrázku vsázky. Tento adresář bude navržen i při příští možnosti volby umístění obrázku při jeho ukládání. Další ze sekcí je sekce „*Locale*“, kde je uložena zkratka, definující jazyk, kterým jsou zobrazovány všechny texty v programu. V šesté sekci s názvem „*Graph*“ je uloženo přiblížení grafu pomocí rozměrů a pozice zobrazované části grafu. Rozměr této vyobrazené části je v souboru rozdělen do dvou částí, „*DomainRange*“ a „*RangeRange*“. Obě tyto části obsahují horní a spodní hranici hodnot s názvem „*UpperBound*“ a „*LowerBound*“. Také je zde uložena pozice kříže, jež má v grafu funkci kurzoru. Dále se zde vyskytuje sekce „*Windows*“, tato sekce je rozdělena na část

„*MainWindow*“ a „*InternalWindows*“. V části „*MainWindow*“ je uložena pozice hlavního okna programu a také pozice dvou komponentů rozdělujících toto okno na tři sektory. V části „*InternalWindows*“ je uložen seznam všech otevřených interních oken s jejich nastaveným stylem obarvování a aktuálními rozměry i pozicemi daných interních oken. Formát tohoto relačního souboru je zobrazen na výpisu 4.5

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ProjectSession>
  <ProjectFiles>
    <FuelLibrary>gafuel_EDU.lib</FuelLibrary>
    <SolutionsCurrent>current_EDU.out</SolutionsCurrent>
    <Layout>EDU_layout.txt</Layout>
  </ProjectFiles>
  <SelectionRange>
    <Width>0.01</Width>
    <Height>0.01</Height>
  </SelectionRange>
  <ImageSize>
    <Width>960</Width>
    <Height>720</Height>
  </ImageSize>
  <ImageDirectory>C:\Users\username\Desktop</ImageDirectory>
  <Locale>en</Locale>
  <Graph>
    <DomainRange>
      <LowerBound>280.23323516845704</LowerBound>
      <UpperBound>342.2897445678711</UpperBound>
    </DomainRange>
    <RangeRange>
      <LowerBound>1.510099971294403</LowerBound>
      <UpperBound>1.9478999733924867</UpperBound>
    </RangeRange>
    <Crosshair>
      <DomainValue>308.1960144042969</DomainValue>
      <RangeValue>1.656000018119812</RangeValue>
    </Crosshair>
  </Graph>
  <Windows>
    <MainWindow>
      <Width>1200</Width>
      <Height>710</Height>
      <VerticalDividerLocation>500</VerticalDividerLocation>
      <HorizontalDividerLocation>400</HorizontalDividerLocation>
    </MainWindow>
    <InternalWindows>
      <InternalWindow id="0">
        <CurrentLineNumber>7916</CurrentLineNumber>
        <ColoredBy>COLORING_BY_AGE</ColoredBy>
        <Bounds>
          <X>0</X>
          <Y>0</Y>
          <Width>500</Width>
          <Height>500</Height>
        </Bounds>
      </InternalWindow>
    </InternalWindows>
  </Windows>
</ProjectSession>
```

Listing 4.5: Ukázka relačního souboru

4.2 Načtení dat

Pro načítání dat byla v programu vytvořena třída „*FileData*“ reprezentující načtená data ze souborů typu „*current*“ a „*gafuel*“. Tato třída obsahuje

seznamy instancí tříd „*CurrentDataRecord*“ a „*GafuelDataRecord*“ reprezentujících záznamy, neboli řádky ze souboru typu „*current*“ a „*gafuel*“. Obě tyto třídy, jak „*CurrentDataRecord*“, tak i „*GafuelDataRecord*“ v konstruktoru přijímají jako parametr jeden řádek z příslušného souboru, který si následně rozparsují do správných předdefinovaných proměnných, které obsahují. Jelikož číselné hodnoty v souborech mohou být občas odděleny více prázdnými znaky, je potřeba do budoucna předcházet problémům s načítáním tím, že budeme předpokládat libovolný počet prázdných znaků kdekoliv mezi daty. Třída „*FileData*“ obsahuje metodu „*CreateFileData*“, která se postará o vytvoření celých seznamů instancí obou tříd „*CurrentDataRecord*“ i „*GafuelDataRecord*“. Tato metoda tedy načte soubor „*current*“ i „*gafuel*“ do paměti programu, pro budoucí použití. Při vytváření instancí je však potřeba dbát na možnost prázdných či zakomentovaných řádků v souborech, které jsou z hlediska užitečnosti dat zcela irelevantní. Vytváření a následné ukládání instancí tříd s těmito řádky by se v budoucnu jistě jevilo více na obtíž, než k užítku. Tudíž je nejprve potřeba tyto řádky identifikovat a v případě jejich existence je ignorovat. To tedy znamená nevytvářet ani neukládat žádné instance tříd obsahující tyto řádky. Pokud při načítání dat ze souborů dojde k jakékoliv chybě, ať už z důvodu neexistence souboru, problému s právy souboru, nebo chybným datům uvnitř souboru, je vyvolána výjimka, která oznámí programu, že došlo k potížím. Tuto výjimku program zachytí, zpracuje a zobrazí uživateli vhodné upozornění.

Do části „*Načítání dat*“ můžeme zařadit i načítání dat ze souboru typu „*layout*“. Data z tohoto souboru se však načítají na jiném místě v programu. Pro uchování dat ze souboru typu „*layout*“ existuje v programu třída s názvem „*GridPositions*“. Tato třída reprezentuje pomyslnou mřížku pro umístění šestiúhelných tlačítek a obsahuje veškeré potřebné informace k této mřížce. Pro načtení dat zde slouží metoda „*loadPositionsFromFile*“, kterou obsahuje třída „*GridPositions*“. Tato metoda načte a uloží do příslušných proměnných všechna data obsažená v zadaném souboru. Při načítání samozřejmě odstraní všechny nepotřebné, prázdné, či zakomentované řádky, stejně jako výše zmíněné třídy při načítání ostatních datových souborů. Formát a data tohoto souboru byla vytvářena autorem tohoto programu, na rozdíl od souborů typu „*gafuel*“ a „*current*“, které byly předem stanoveny. Tudíž je možné definici tohoto formátu nastavit přísněji. Z toho důvodu tedy není třeba se zabývat problémy, jako jsou předem nedefinovaný počet prázdných znaků mezi daty, nebo možnost reprezentace a náhrady oddělovacího znaku celou sadou znaků, například prázdných znaků. Je tedy možné se při parsování obsahu souboru zaměřit přímo na jeden znak mezery, který od sebe odděluje významná data. Stejně jako v předchozím případě u souborů typu

„*current*“ a „*gafuel*“, i zde při načítání dat ze souboru „*layout*“, může dojít k chybnému načtení dat a i zde je vyvolána výjimka, jež je následně programem zpracována.

4.3 Načtení projektu

Pro načtení souboru projektu a založení nového projektu se v programu vyskytuje metoda s názvem „*loadProject*“. Tato metoda otevře zadaný projektový soubor a načte z něj všechny potřebné údaje. Pokud program narazí na prázdnou, nebo zakomentovanou řádku, tak jí přeskočí. Po úspěšném načtení všech potřebných údajů je ověřena existence všech datových souborů a taktéž jsou ověřena přístupová práva potřebná ke čtení z těchto souborů. Pokud tento test neuspěje, je vyvolána a následně zpracována chybová zpráva. Jestliže však test proběhne bez jakýchkoliv problémů, je řízení programu předáno jiné metodě, která obstará načtení dat z datových souborů. Pokud se ovšem nevyskytnou v souboru všechna potřebná data, nebo se je z jakéhokoli důvodu nepovede načíst, metoda „*loadProject*“ vyvolá chybovou výjimku. Program následně tuto výjimku zpracuje a zobrazí příslušné hlášení uživateli programu. Stejným způsobem je zpracována chyba, pokud metoda narazí na neplatné informace v souboru, které program nezná. Při neexistenci celého projektového souboru je výjimka vyvolána automaticky a program jí taktéž zpracuje a zobrazí uživateli. Nakonec, před ukončením této metody „*loadProject*“, je vytvořena nová relace ze získaných údajů a je uložena pod defaultně zvolený název. Tato relace však bude řešena detailněji v sekci 4.5 „*Uložení relace*“, kde bude ukládání relací popsáno podrobněji.

4.4 Načtení relace

Pro načítání relací v programu slouží metoda „*loadSession*“. Tato metoda ze souboru relace, za předpokladu jeho existence a vlastnění práv k jeho čtení, nejprve načte datové soubory projektu. Tyto soubory jsou stejně jako při načítání souboru projektu nejdříve otestovány, zda existují a lze z nich číst potřebná data. Pokud test selže, je uživateli programem zobrazena chybová zpráva. Pokud načtené soubory testem projdou, jsou z nich pomocí metody „*loadFiles*“ načtena všechna potřebná data. Jelikož jsou v souboru relace uloženy pouze názvy těchto datových souborů, předpokládá se jejich exis-

tence ve stejném adresáři, v jakém se nalézá relační soubor. Nejdříve je tedy nutné získat informaci o cestě k adresáři, ve kterém se tento relační soubor nalézá. Poté se tato cesta spojí s názvy datových souborů a tyto údaje je nakonec možné vložit do metody „loadFiles“, jež z nich načte data. Po načtení projektových souborů a jejich dat pokračuje metoda „loadSession“ v načítání dalších částí relačního souboru. V tomto souboru se dále vyskytuje sekce „SelectionRange“. Tato sekce určuje oblast, ve které se po kliknutí na graf označí body grafu. Tato oblast je velikost okolí místa kliknutí. V metodě „loadSession“ se tedy načte oblast „SelectionRange“. Po načtení této oblasti je načtena velikost ukládaných obrázků z oblasti „ImageSize“ a následně je načteno umístění adresáře z oblasti „ImageDirectory“, jež bude navrhován pro ukládání těchto obrázků. Po načtení všech těchto informací je načtena a následně aplikována lokalizace programu ze sekce s názvem „Locale“. V souboru relace se ještě vyskytuje sekce „Graph“, obsahující informace o grafu a sekce „Windows“, obsahující informace o hlavním okně a interních oknech. Obě tyto sekce jsou řešeny metodami s názvem „loadViewFromXML“, každá však náleží jiné třídě.

Sekce „Graph“ je řešena metodou „loadViewFromXML“ obsaženou ve třídě „SwitchingGraphs“. Této metodě je předán celý uzel reprezentující sekci „Graph“. Metoda nejprve ověří jeho existenci a poté začne načítat a získávat data z jeho obsahu. Po načtení dat ze sekce „DomainRange“ i „RangeRange“, obsažených v uzlu „Graph“, je grafu v programu nastavena výška a šířka zobrazované oblasti. To má prakticky za následek nastavení přiblížení grafu. Také se zde vyskytuje sekce „Crosshair“. Po jejím načtení dojde v programu k okamžitému nastavení kurzorového kříže grafu do příslušného místa určeného příslušnými, ze souboru načtenými, souřadnicemi. Po načtení a nastavení kurzorového kříže grafu se zavolá metoda, která obstarává obnovení grafu. To má za následek označení bodů okolo tohoto kříže v oblasti nastavené v „SelectionRange“ a dojde k přidání těchto označených bodů do „ListBoxu“ nacházejícím se pod grafem.

Sekce „Windows“ je načítána pomocí metody „loadViewFromXML“ ve třídě s názvem „MainFrame“. Nejprve jsou načteny informace o hlavním okně programu ze sekce „MainWindow“. Mezi tyto informace se řadí výška a šířka okna a pozice vertikálního i horizontálního rozdělovače okna. Po načtení těchto informací jsou hlavnímu oknu programu nastaveny rozměry a příslušným rozdělovačům obsaženým v tomto okně jsou nastaveny pozice. Dále se načte seznam všech interních oken programu a jejich vlastností. Při načtení každého interního okna se toto okno vytvoří v programu a nastaví se mu všechny příslušné údaje načtené ze souboru, kterými jsou číslo řádku

záznamu v souboru „*current*“, způsob obarvování šestiúhelných tlačítek a kompletní rozměry okna jako jsou šířka, výška a souřadnice X a Y místa, kde se dané interní okno nachází. Tyto interní okna jsou vytvořené přesně, v pořadí v jakém jsou uváděny v souboru a tudíž i v pořadí, v jakém jsou načítány informace o nich ze souboru. To umožňuje po načtení informací o těchto interních oknech mít všechna okna zobrazená a překrytá přesně ve stejné pozici, v jaké byly uloženy.

4.5 Uložení relace

K ukládání relačního souboru dochází na vyžádání, nebo automaticky při vytvoření nového projektu. Jelikož ihned po vytvoření nového projektu nejsou v programu vytvořena žádná okna, nejsou ani označeny žádné body v grafu. Je uloženo pouze základní nastavení a spousta informací v relačním souboru chybí. Tyto chybějící informace však v souboru nejsou povinné a je možné tento soubor bez problému načíst. Po načtení tohoto souboru se program ocitne ve stejném stavu jako při vytvoření nového projektu pomocí projektového souboru, pomocí kterého byl projekt vytvořen v době, kdy došlo k automatickému uložení této relace. Do souboru relaci ukládá metoda „*saveSession*“. Tato metoda potřebuje ke své funkčnosti mít zadány názvy všech datových souborů i název souboru relace, do kterého se tato relace uloží. Všechny tyto informace jsou obsaženy v parametrech, které metoda „*saveSession*“ přijímá. Nejprve se v paměti vytvoří všechn obsah relačního souboru a poté se tento obsah do souboru zapíše. Struktura tohoto souboru je ve formátu „*XML*“ a je podrobněji popsána výše v sekci 4.1.5. Nejprve se vytvoří hlavní uzel s názvem „*ProjectSession*“ a do něj se vkládají ostatní uzly. Pod uzel „*ProjectFiles*“ se uloží všechny, z parametru metody „*saveSession*“ získané, názvy datových souborů typu „*current*“, „*gafuel*“ a „*layout*“. Následně se uloží informace o „*SelectionRange*“, jeho výška i šířka. Za ním následuje uložení rozměru ukládaných obrázků pod název „*ImageSize*“ a uložení cesty k adresáři pro ukládání těchto obrázků pod název „*ImageDirectory*“. Dalším důležitým prvkem v nastavení je lokalizace. Aktuálně zvolená lokalizace se uloží pod název „*Locale*“. Pokud je v programu načten a zobrazen nějaký graf, program předá vypracování uzlu „*Graph*“ metodě „*saveViewToXML*“ obsaženou ve třídě „*SwitchingGraphs*“.

V této třídě „*SwitchingGraphs*“ metoda „*saveViewToXML*“ nejprve vytvoří uzel „*Graph*“ a poté do něho přidává další uzly. Vytvoří dva uzly „*DomainRange*“ a „*RangeRange*“ pro uchování přiblížení grafu. Do těchto

uzlů metoda přidá příslušné uzly obsahující spodní a horní hranici hodnot dané osy. Následně se zde do uzlu „*Graph*“ uloží uzel „*Crosshair*“ se souřadnicemi dvou os grafu, na kterých se momentálně vyskytuje kurzorový kříž.

Po vytvoření uzlu „*Graph*“ metodou „*saveViewToXML*“, je tento uzel uložen jako potomek uzlu „*ProjectSession*“ a poté následuje zpracování uzlu „*Windows*“. Tento uzel se taktéž zpracovává v jiné metodě i třídě, stejně jako uzel „*Graph*“. Jeho vypracování je dokonce zpracováno metodou téhož názvu, tedy „*saveViewToXML*“, ta se ale tentokrát vyskytuje ve třídě s názvem „*MainFrame*“.

Zde metoda „*saveViewToXML*“ vytvoří hlavní uzel „*Windows*“ a do něj ukládá ostatní uzly. Prvním z těchto uložených uzlů, hned pod uzlem „*Windows*“, je uzel „*MainWindow*“. Do něj se uloží aktuální rozměry hlavního okna programu a pozice všech rozdělovačů v tomto okně. Dalším z uzlů, uloženým přímo pod uzlem „*Windows*“, je uzel „*InternalWindows*“. V tomto uzlu se vyskytuje seznam uzlů „*InternalWindow*“, kde každý z těchto uzlů představuje informace o jednom vnitřním otevřeném okně. Proto tedy program zjistí kolik a která vnitřní okna jsou otevřená, a ke každému oknu vytvoří uzel „*InternalWindow*“ s příslušným identifikačním číslem uloženým v atributu uzlu, pojmenovaným jako „*id*“. V každém z těchto uzlů „*InternalWindow*“ se uloží aktuální informace o interním okně, jako jsou výška, šířka, aktuální pozice, způsob obarvování hexadecimálních tlačítek a číslo řádku z načteného souboru typu „*current*“, jenž toto okno vizualizuje.

Po vypracování uzlu „*Windows*“ se, stejně jako uzel „*Graph*“, uloží pod hlavní uzel „*ProjectSession*“. Po vypracování celého obsahu souboru relace je tento obsah uložen do souboru s názvem zadaným v parametru metody. Tento soubor je otevřen a vytvořen pro zápis. Pokud tento soubor již existuje, bude jeho obsah kompletně přepsán novým obsahem. Jestliže v průběhu zápisu do souboru dojde k jakékoliv chybě, je automaticky vygenerována chybová výjimka, kterou následně program odchytí, zpracuje a uživateli tohoto programu zobrazí příslušnou upozornovací chybovou zprávu.

4.6 Lokalizace

Program aktuálně obsahuje anglickou a českou lokalizaci. Pro tyto jazyky se u zdrojových souborů programu vyskytují soubory „*language_cs.properties*“ a „*language_en.properties*“. Je však možné jednoduše přidat do programu

novou lokalizaci přidáním nového souboru s obsahem informací o novém jazyce. V názvech těchto souborů se mezi znakem tečky a podtržítkem vyskytuje zkratka jazyka, pro který tento soubor obsahuje informace. Pomocí této zkratky se v programu k daným lokalizovaným textům přistupuje. Do programu se však informace z těchto souborů dostanou při kompilaci programu, nikoliv až při běhu programu. Není proto možné spustit program s lokalizací, kterou předem nepodporuje. Každý z těchto lokalizačních souborů obsahuje texty v jazyce, který lokalizuje. Každý z těchto textů je uložen zvlášť na řádce. Na každé řádce se v těchto souborech vyskytují také konstanty, ke kterým se tyto texty přiřazují a pomocí kterých se v programu přistupuje k lokalizovaným textům. Řádek lokalizačního souboru má tedy tvar takový, že na začátku se vyskytuje zmiňovaná konstanta, za ní následuje znak rovnítko „=“ a za tímto znakem je uložen lokalizovaný text, který je reprezentován právě touto konstantou, ke které je přiřazen.

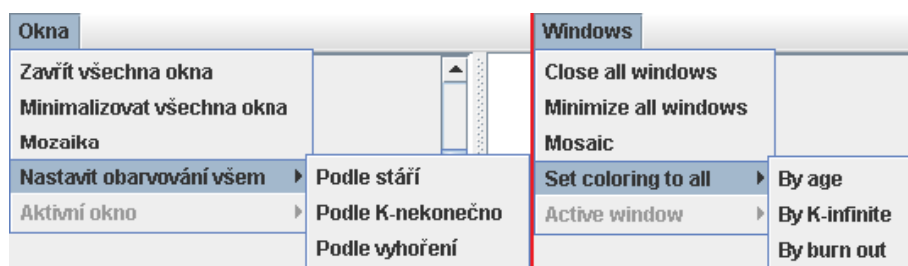
K použití lokalizačních textů v programu je nejprve nutné inicializovat instanci třídy, která se v „*Jave*“ využívá právě k lokalizacím programů, a jmenuje se „*ResourceBundle*“. K tomu je však nutné zvolit jazyk, respektive zkratku jazyka, který chceme použít pro nastavení textů v programu. Protože při spuštění programu není jasné, v jakém jazyce mají být texty programu zobrazeny, musí být zvolen defaultní jazyk, který se při spuštění programu nastaví. Jako defaultní jazyk byla zvolena angličtina. K nastavení jazyka pro lokalizaci programu dochází v několika případech a těmi jsou spuštění programu, načtení relace ze souboru, nebo nastavením jazyka přímo přes uživatelské rozhraní programu uživatelem. Zkratka aktuálně nastaveného jazyka se ukládá i do souboru relace, proto je při uložení relace aktuálně nastavená lokalizace uchována.

K nastavení lokalizovaných textů v celém programu dojde při zavolání metody „*setComponentsText*“ nad instancí třídy „*MainFrame*“, která reprezentuje hlavní okno programu. Tuto metodu je možné zavolat kdykoli a vyvolat tak změnu textů v celém programu na aktuálně nastavenou lokalizaci. Metoda „*setComponentsText*“ nastaví všem komponentům v hlavním oknu programu příslušné texty, které získá pomocí reprezentativních konstant. Nakonec tato metoda v cyklu projede všechny interní okna, které hlavní okno programu obsahuje, a zavolá nad nimi jejich vlastní stejnojmennou metodu „*setComponentsText*“. Ta má na starost nastavení lokalizovaných textů uvnitř daného interního okna. Tato metoda reaguje podobně jako její stejnojmenná, výše zmíněná, „*kolegyně*“. A to tím způsobem, že projde všechny komponenty obsažené v daném interním okně, nad kterým je tato metoda volána, a nastaví jejich texty na příslušné lokalizované texty získané

pomocí reprezentativních konstant.

V programu existuje ještě jedna metoda s názvem „*setComponentsText*“, která je obsažena ve třídě s názvem „*SelectionDialog*“. Tato metoda ovšem nemusí být v řetězci propojena s ostatními metodami, protože je volána v konstruktoru třídy „*SelectionDialog*“ a tudíž se všechny lokalizované texty nastaví při vytváření této třídy. Tato třída však reprezentuje okno programu a během jeho zobrazení není možné změnit v programu nastavení lokalizace. Volání metody „*setComponentsText*“ v konstruktoru je tedy dostačující. Jak jistě rozbor této metody napovídá, stejně jako ostatní metody s tímto názvem, i tato metoda nastavuje všechny texty všech komponentů v celé třídě na lokalizované texty pro aktuálně zvolený jazyk.

Názorně je změna textů v programu ukázána na obrázku 4.1.



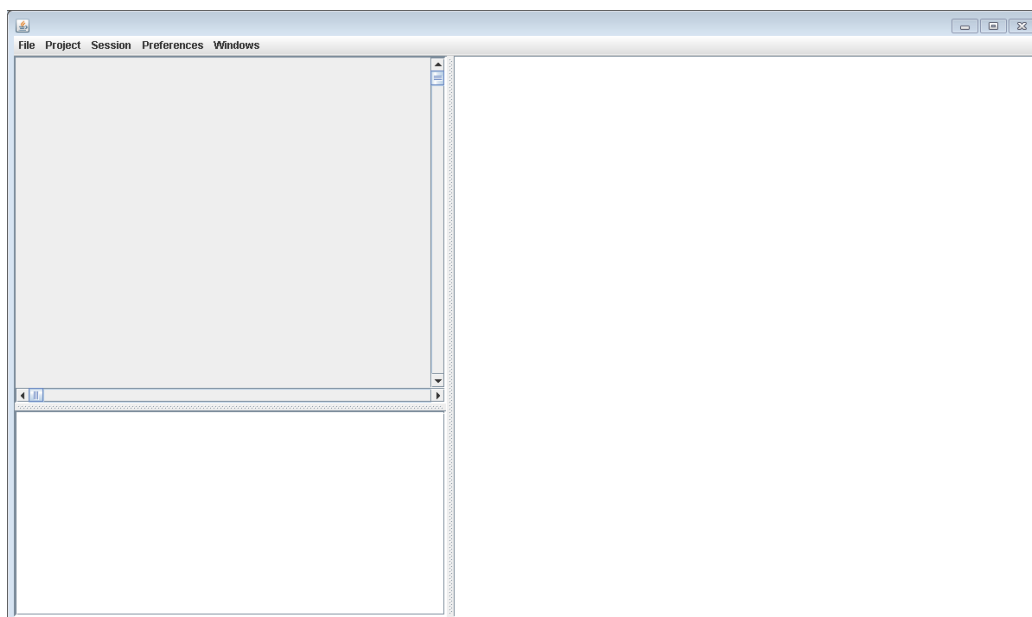
Obrázek 4.1: Ukázka různých lokalizací

4.7 Komponenty programu

Celé uživatelské rozhraní programu je složeno z několika různých komponentů poskytnutých knihovnou „*Swing*“, napsanou pro programovací jazyk „*Java*“. Všechny vlastnosti a rozměry těchto komponentů byly upraveny a všechny jejich možné akce, které podporují, byly naprogramovány podle potřeby, za cílem zkonstruovat výsledný cílový program, který bude intuitivní pro koncového uživatele.

4.7.1 Hlavní okno

Hlavní okno programu se vytvoří ihned po spouštění programu. Toto okno obsahuje převážnou většinu všech komponentů v programu. Tyto komponenty



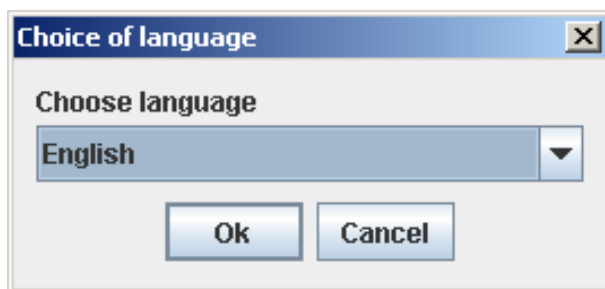
Obrázek 4.2: Hlavní okno

jsou níže popsány podrobněji. V horní části celého hlavního okna se vyskytuje menu. Pod tímto menu je obsah celého okna a je rozdělen na tři části. Tyto části jsou odděleny komponenty s názvem „*JSplitPane*“. Tyto oddělovací komponenty jsou schopny se posouvat po celé délce okna, vertikálně, nebo horizontálně, podle toho jakým způsobem jsou nastaveny. Tímto posouváním upravují velikosti částí okna, které rozdělují. Jelikož reagují na interakci s uživatelem, může si uživatel pomocí nich upravit velikosti všech částí hlavního okna. První část se vyskytuje vlevo nahoře a obsahuje graf. Druhá část se nalézá vlevo dole pod grafem a obsahuje seznam všech označených bodů v grafu. Poslední, třetí, část zabírá celou pravou polovinu hlavního okna a vyskytuje se v ní interní plocha pro interní okna. Obsahy všech těchto částí jsou detailněji popsány níže. Vzhled hlavního okna po spuštění programu je možné si prohlédnout na obrázku 4.2

4.7.2 Výběrové okno lokalizace

Kromě hlavního okna se v programu vyskytuje několik dalších menších oken, která slouží k nastavování konkrétních vlastností, pro které je dané okno vyrobené. Ke všem těmto oknům lze přistupovat pomocí menu v horní části obrazovky. Jedním z těchto menších oken, doprovázející okno hlavní, je okno pro

nastavení lokalizace programu. To se otevře po kliknutí na tlačítko „Změnit jazyk“, které obsahuje, v defaultně nastaveném jazyce, text „Change language“. Celé toto okno obsahuje pouze tři komponenty. Komponent s názvem „JListBox“, který se zde vyskytuje, slouží pro zobrazení seznamu dostupných jazyků, pro které je možné nastavit lokalizaci celého programu. V tomto seznamu je tedy možné zvolit jakýkoliv z těchto jazyků. Komponent „JListBox“ však umožňuje zvolení pouze a právě jednoho prvku v zobrazovaném seznamu, což je v tomto případě přímo žádoucí a z tohoto důvodu byl tento komponent zvolen. Dále toto okno obsahuje dvě tlačítka, tedy komponenty s názvem „JButton“. Jedno z těchto tlačítek, umístěného vpravo, má stornovací funkci. Po stisknutí tohoto tlačítka tedy nejsou provedeny žádné změny ani úpravy v nastavení lokalizace programu a celé okno se po jeho stisknutí uzavře. Stejným způsobem okno reaguje na kliknutí kurzorem na uzavírací ikonku okna, připomínající písmeno X, takzvaného „křížku“. Druhé tlačítko, nalézající se v tomto popisovaném okně, je umístěno vlevo od předešlého tlačítka a plní funkci aplikování změn. Po jeho stisknutí se provedou změny v nastavení programu a všechny texty obsažené v programu se změni na lokalizované texty právě nastaveného jazyka. Stejně jako u předchozího tlačítka se po obslužení události kliknutí na toto tlačítko uzavře celé okénko, obsahující toto tlačítko. Obě tlačítka zobrazují text reprezentující jejich funkčnost. Tyto texty však nejsou neměnné, ale jsou změněny podle aktuálně nastavené jazykové lokalizace. Proto zde nemá smysl zmiňovat veškeré texty zobrazované těmito tlačítky. Všechny tyto texty však mají něco společného a tím jsou jejich reprezentativní konstanty. Lokalizované texty zobrazené levým tlačítkem, tedy tlačítkem ukládající změny, jsou získány pomocí reprezentativní konstanty s názvem „ok“. Oproti tomu lokalizované texty zobrazované pravým tlačítkem, tedy tlačítkem zahazujícím změny, jsou získávány pomocí reprezentativní konstanty s názvem „cancel“. Posledním komponentem tohoto okna je „JLabel“, jehož jediným úkolem je zobrazovat text. V tomto případě jde o lokalizovaný text získaný pomocí reprezentativní konstanty „chooseLanguage“. Ještě se zde vyskytuje jeden lokalizovaný text, ten však není obsažen v žádném komponentu uvnitř okna, nýbrž je přímo součástí okna a to jako nadpis v záhlavní liště. Tento text je získáván pomocí reprezentativní konstanty s názvem „choiceOfLanguage“. Toto okno může mít v programu pouze jednu instanci a po dobu zobrazení tohoto okna je ovládnutí hlavního okna dočasně zablokované. Jak toto okno vypadá, je možné si prohlédnout na obrázku 4.3.

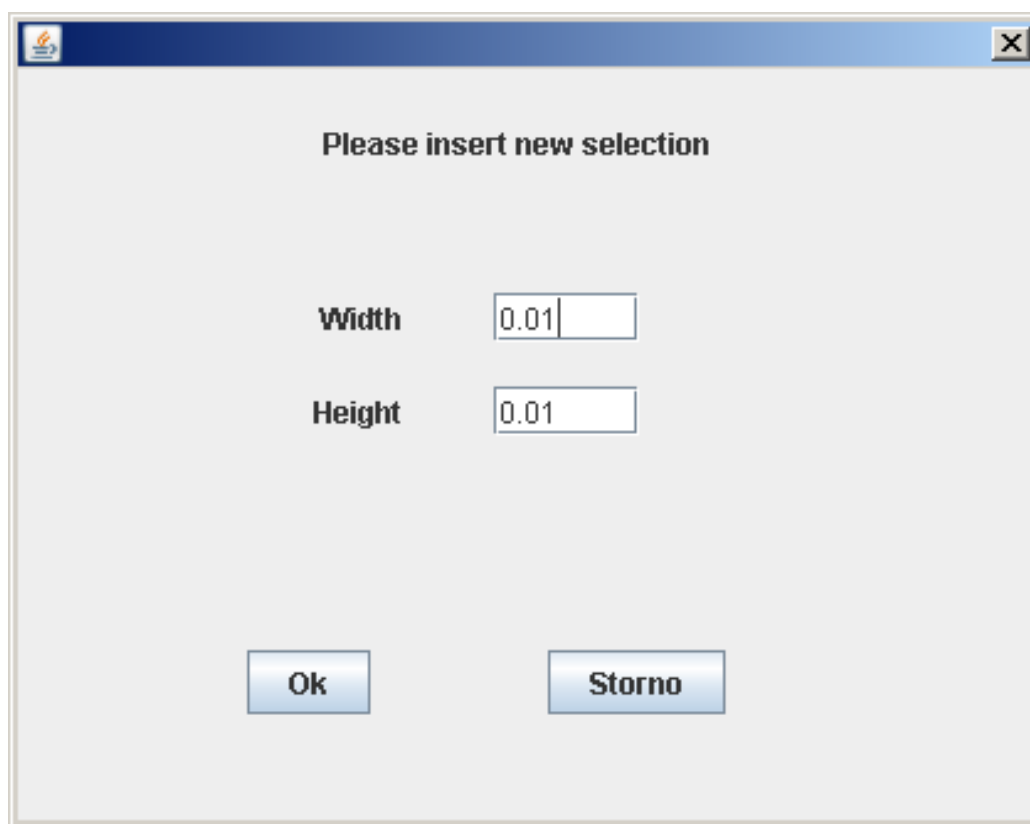


Obrázek 4.3: Výběrové okno lokalizace

4.7.3 Výběrové okno proměnné „*SelectionRange*“

Toto okno, viz obrázek 4.4, slouží k nastavování a úpravě výšky a šířky proměnné „*SelectionRange*“, jejíž význam byl zde již několikrát zmíněn. Spouští se kliknutím na tlačítko v menu, jehož reprezentativní konstanta je „*changeSelection*“. Pro toto okno je v programu vytvořena speciální třída s názvem „*SelectionDialog*“ starající se pouze o obsluhu a zobrazení tohoto okna. Tato třída je oddělena od třídy „*JDialog*“, jež je součástí „*Javovské*“ knihovny „*Swing*“. Okno, které třída „*SelectionDialog*“ zobrazuje, obsahuje tři komponenty typu „*JLabel*“, kde všechny pouze zobrazují nastavené texty.

První z nich je umístěn nahoře uprostřed okna a vyzývá uživatele k zadání nové oblasti výběru. Jeho text je získán pomocí konstanty s názvem „*plsSetSelection*“. Další dva komponenty zobrazují popisný text k políčkům pro vyplnění. Tyto texty znamenají výšku a šířku v různých jazycích, pro jaké je zrovna nastavena lokalizace programu. Toto okno má, stejně jako předchozí, výše popsané okno, dvě tlačítka. Jedno pro uložení a aplikování změn v nastavení programu a druhé pro zahození těchto provedených změn. Navíc se zde však vyskytují dvě textová políčka, která jsou určena k zapsání číselných hodnot uživatelem. Tato zapsaná čísla reprezentují výšku a šířku oblasti uloženou v proměnné „*SelectionRange*“. Po kliknutí na tlačítko, jež aplikuje změny, tak program rozpozná texty ve vyplňovacích políčkách, převede je na čísla a poté je uloží do proměnné „*SelectionRange*“. Pokud však při rozpoznávání těchto čísel dojde k problémům, například z toho důvodu, že je v políčkách zadán text s jinými, než číselnými znaky a desetinnou tečkou, či čárkou, je uživateli zobrazena chybová zpráva informující o tom, že zadaná čísla jsou neplatná.



Obrázek 4.4: Výběrové okno proměnné „*SelectionRange*“

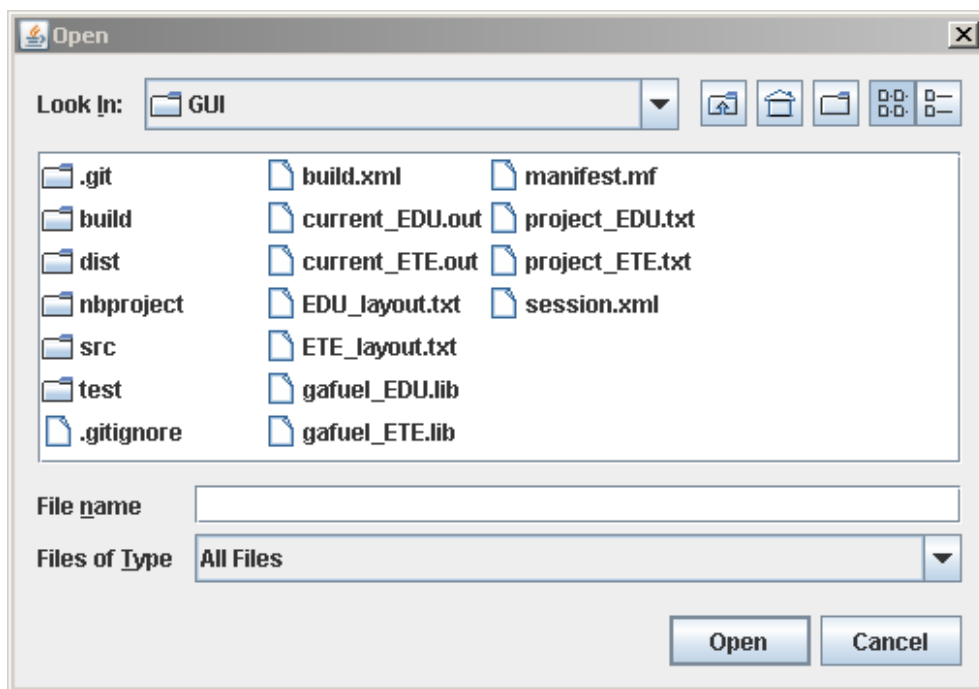
4.7.4 Výběrové okno rozměrů obrázku

Toto okno slouží pro zvolení rozměrů obrázku, který se bude ukládat při příštím vyžádání uživatele. Volba rozměrů probíhá podobně jako v předchozím případě v sekci 4.7.3. Toto okno také vypadá podobně jako okno v předchozí sekci, liší se jen zobrazenými texty.

4.7.5 Dialogové okno načítání a ukládání

V programu se vyskytuje několik dialogových oken pro ukládání a načítání souborů. Ta jsou použita pro uložení relace, načtení relace a pro vytvoření nového projektu pomocí načteného projektového souboru. Všechna tato dialogová okna mají podobný vzhled a liší se jen detaily, jakými jsou například texty. Pro porovnání je rozdíl mezi těmito okny vidět na obrázcích 4.6 a 4.5.

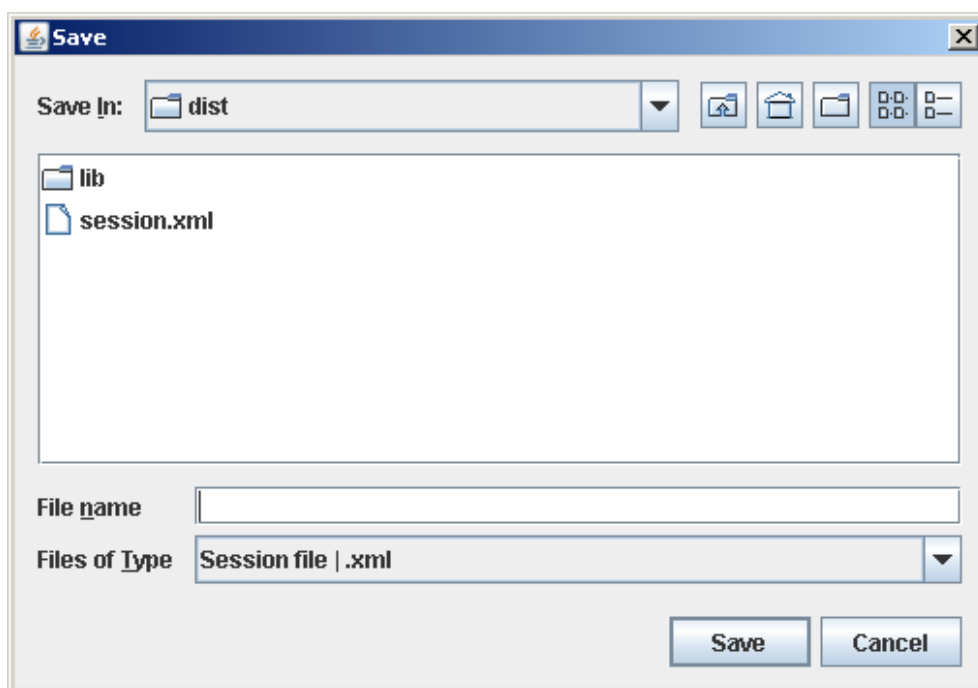
Jsou vytvářeny pomocí třídy „*JFileChooser*“ obsažené v knihovně „*Swing*“ pro platformu „*Java*“. Tato třída umožňuje snadné vytváření dialogových oken pro zvolení souborů pro práci. Všechny texty v těchto dialogích jsou lokalizovány v metodě „*setComponentsText*“, obsažené ve třídě „*MainFrame*“, a to pomocí nastavení výchozích hodnot v programu pro grafické rozhraní. Tyto výchozí hodnoty umožňuje měnit statická metoda s názvem „*put*“ uložená ve třídě „*UIManager*“. Jelikož všechna tato dialogová okna dočasně zablokují ovládání hlavního okna, není možné změnit jazyk lokalizace textů při otevření některého z těchto dialogových oken. Z toho důvodu není potřeba obstarávat přenačtení a aktualizování textů dialogových oken v průběhu jejich zobrazení, ale stačí tyto texty nastavit před otevřením těchto dialogových oken, nebo kdykoliv v průběhu, kdy jsou tato okna zavřena. V celém programu je možné mít zároveň otevřené pouze jedno z těchto oken.



Obrázek 4.5: Dialogové okno načítání souborů

4.7.6 Menu

Menu se v programu vyskytuje v listě, v horní části hlavního okna. Toto menu slouží k ovládání základních pokynů, operací a nastavení programu.



Obrázek 4.6: Dialogové okno ukládání souborů

Menu obsahuje tlačítka s lokalizovaným textem. Při spuštění programu je zpočátku, bez interakce s uživatelem, vidět pouze několik tlačítek. Všechna tato tlačítka je možné rozkliknout, což způsobí zobrazení celého seznamu dalších tlačítek, které se pod nimi skrývají.

Prvním z těchto tlačítek je tlačítko s textem „*Soubor*“ v podání různých jazyků, podle toho jaký jazyk je aktuálně nastaven v jazykové lokalizaci celého programu. K tomuto tlačítku je přiřazena reprezentativní konstanta textů s názvem „*file*“. Po jeho rozkliknutí je zobrazeno tlačítko pro ukončení aplikace. Dále se zde vyskytují tlačítka s názvy „*Projekt*“, „*Relace*“, „*Nastavení*“ a „*Okna*“. Pod tlačítkem „*Projekt*“ se skrývají tlačítka s reprezentativními konstanty „*newProject*“ a „*closeProject*“. Pod tlačítkem „*Relace*“ se vyskytují tlačítka s konstantami „*loadSession*“, „*saveSession*“ a „*closeSession*“. Dalším z těchto hlavních tlačítek obsahuje pouze tři tlačítka. Tímto tlačítkem je tlačítko „*Nastavení*“, které pod sebou skrývá „*Změnit výběr*“, „*Změnit jazyk*“ a „*Nastavit velikost obrazu pro uložení*“.

Nejvíce tlačítek pod sebou skrývá tlačítko „*Okna*“. Toto tlačítko pod sebou skrývá dokonce další rozklikávací menu, takže některá tlačítka jsou zde zanořena hlouběji. Je zde tlačítko pro uzavření všech oken, tlačítko pro sesku-

pení všech oken do mozaiky a tlačítko pro minimalizování všech oken. Dále se zde vyskytují dvě menu, první z nich dává uživateli na výběr tři možnosti, který z typů obarvování zvolí pro všechna interní okna v aplikaci. Druhé menu je totožné s kontextovým menu, jenž se objeví po kliknutí pravým tlačítkem na některé z interních oken. Toto menu se týká aktuálně označeného a aktivního interního okna a tudíž je nepřístupné ve stavu, kdy se v programu nevyskytuje žádné interní okno. V tomto podmenu k aktivnímu oknu je možné zvolit akce jako uložení aktuálně zobrazené vsázky v podobě obrázku, zkopírování původních dat vsázky, vyhledání vsázky v grafu, obnovení poměru stran interního okna, zavření ostatních interních oken vyjma onoho aktivního a akce minimalizování všech ostatních interních oken.

Tlačítka s reprezentativními konstantami „*newProject*“, „*loadSession*“ a „*saveSession*“ užívají dialogová okna vytvořená pomocí třídy „*JFileChooser*“ ke zvolení umístění a názvu souboru, se kterým potom dále pracují. Tlačítko s konstantou „*newProject*“ slouží k vytvoření nového projektu. Projekt ovšem vytváří pomocí dat v projektovém souboru, proto využívá dialogové okno k získání tohoto souboru. Následně načte tento soubor a pomocí dat získaných z tohoto souboru vytvoří nový projekt. Detailněji je tento postup popsán výše, v sekci 4.3 s názvem „*Načtení projektu*“. Tlačítko nesoucí texty získané pomocí reprezentativní konstanty s názvem „*loadSession*“ pomocí dialogového okna získá soubor relace a poté načte data z tohoto souboru a obnoví relaci programu pomocí těchto dat. Struktura relačního souboru je popsán v sekci 4.1.5 s názvem „*Soubor relace*“ a postup při načítání tohoto souboru a obnovování relace programu je popsán v sekci 4.4 s názvem „*Načtení relace*“.

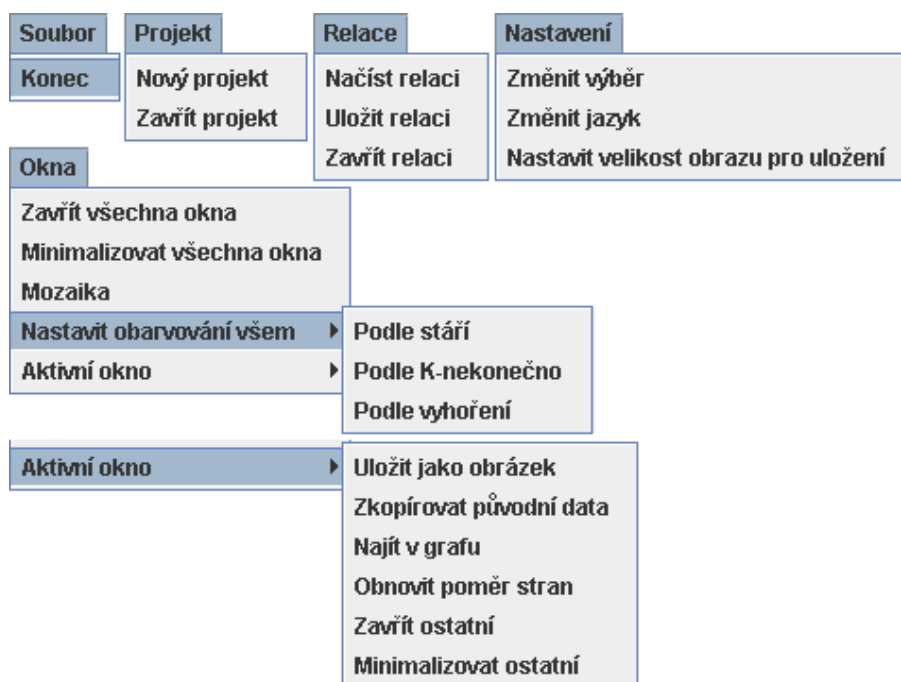
Další tlačítko nesoucí konstantu „*saveSession*“ slouží k uložení aktuální relace programu do souboru relace. K tomu ovšem potřebuje znát jméno a umístění tohoto souboru, který má za úkol vytvořit. Proto vyvolá dialogové okno, kde má uživatel programu možnost sdělit přesně tyto údaje. Podrobně je způsob uložení relace popsán v sekci 4.5 s názvem „*Uložení relace*“ a struktura relačního souboru je popsána v sekci 4.1.5 s názvem „*Soubor relace*“. Položka menu s názvem „*closeProject*“ je tlačítko, které již dialogová okna nepotřebuje ke své činnosti. Jejím jediným úkolem je totiž uzavřít projekt, či relaci. Stejnou funkci plní i tlačítko nesoucí konstantu „*closeSession*“. Tudíž po stisknutí tohoto tlačítka se program ocitne ve stavu, ve kterém byl ihned po spuštění, před načtením relace či vytvořením nového programu. Většina komponentů programu v tomto stavu neobsahují žádné údaje. Funkce provozovaná tímto tlačítkem se s nutností využívá taktéž před načtením jiné relace, než je aktuálně otevřená, nebo při vytvoření nového projektu ve chvíli, kdy program již nějaká načtená data vizualizuje. Stisknutím tohoto tlačítka je

však možné tuto funkcionalitu vyvolat manuálně.

Prvním a posledním tlačítkem, které se zobrazí při stisknutí tlačítka s konstantou „file“, je tlačítko nesoucí reprezentativní konstantu „end“. Tímto tlačítkem je možné zavřít hlavní okno programu a ukončit tím celý program, stejně jako po kliknutí na uzavírací křížek v rohu okna.

Tlačítko s konstantou „changeSelection“ otevře dialogové okno pro nastavení oblasti „SelectionRange“. Detailněji je toto okno popsáno v sekci 4.7.3 s názvem „Výběrové okno proměnné SelectionRange“. Poslední tlačítko, nacházející se v menu, nese reprezentativní konstantu „changeLanguage“. Toto tlačítko slouží ke změně jazyka a nastavení jazykové lokalizace programu. Z tohoto důvodu je po kliknutí na tlačítko vyvoláno dialogové okno umožňující uživateli programu zvolení tohoto jazyka. Detailněji je toto dialogové okénko popsáno v sekci 4.7.2 s názvem „Výběrové okno lokalizace“.

Rozložené celé menu je vyobrazeno na obrázku 4.7.



Obrázek 4.7: Ukázka položek menu

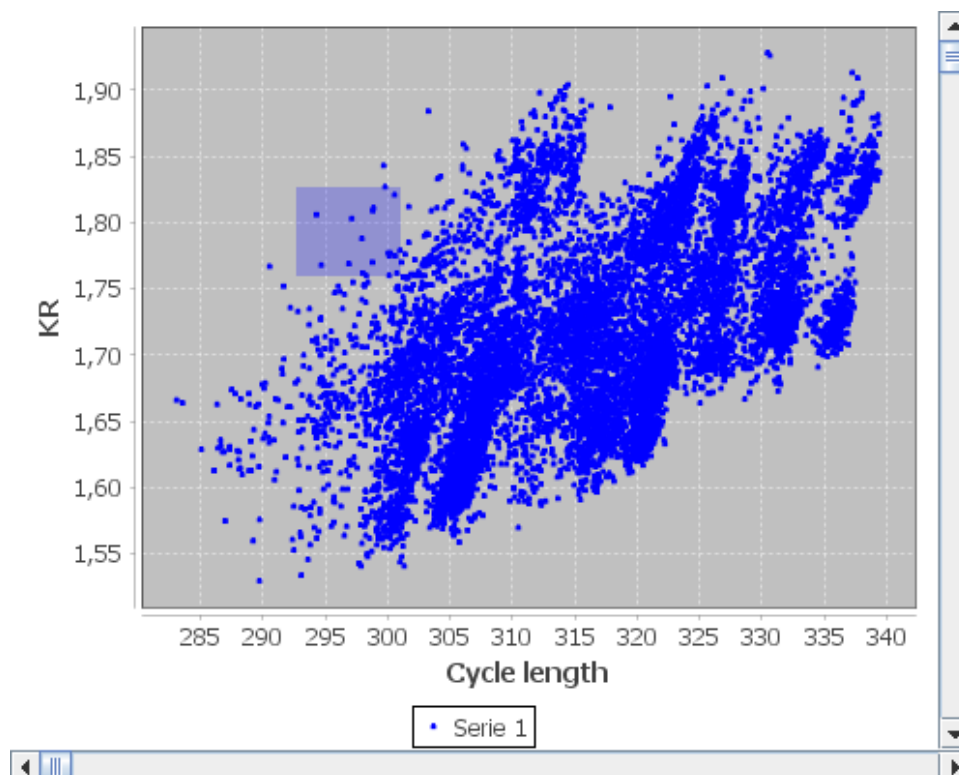
4.7.7 Graf

Graf je jednou z nejdůležitějších částí programu. Vyskytuje se v levé horní části hlavního okna programu. Vizualizuje data získaná ze souborů typu „*current*“. Každá řádka v tomto souboru představuje jeden bod v grafu. Osa Y grafu představuje hodnotu „*KR*“ a osa X je určena hodnotou „*Délka cyklu*“. Názvy těchto hodnot jsou v grafu u obou os zobrazeny. Obě tyto hodnoty jsou uloženy jako dvě samostatné čísla ve výše zmíněných řádcích souboru typu „*current*“. V grafu je vyznačená mřížka a u obou os jsou vypsány orientační hodnoty popisující přímkou této mřížky.

Celý graf je realizován pomocí knihovny s názvem „*JFreeChart*“. V této knihovně se vyskytuje několik typů grafů s odlišnými vlastnostmi. Některé typy grafů jsou přizpůsobeny na zobrazování velkého počtu bodů a jiné jsou optimalizovány pro pokročilejší operace s body a grafem, jako je například označení bodu v grafu. Jelikož tento program byl původně navrhován pro budoucí rozšíření o zobrazování o několik řádů více bodů najednou, než je aktuálně používán a než je momentálně zapotřebí, je v programu vytvořena třída s názvem „*SwitchingGraphs*“, která z návrhového hlediska předpřipravuje program pro přepínání mezi vícero typy grafů v průběhu zobrazení, pomocí rozpoznávání a reagování na velikost a přiblížení aktuálního zobrazení plochy grafu. Tato knihovna umí zobrazovat data pouze ve formátu dvourozměrného pole „*doublů*“, nebo „*floatů*“, proto je před použitím grafu toto pole vytvořeno a naplněno údaji ze záznamů získaných ze souboru typu „*current*“, které jsou uloženy v proměnné „*currentDataRecords*“, čímž jsou v paměti programu uloženy stejné údaje vícekrát různými způsoby. Z hlediska návrhu rychlosti a přehlednosti kódu se však toto řešení jeví jako nejvhodnější a s momentálně používanými a potřebnými počty bodů v grafu toto řešení zabírá pouze zanedbatelné množství paměti a při testování nevykazuje žádné problémy.

Uživatel má možnost přiblížit zobrazení grafu a to tím, že vyznačí novou oblast grafu, která se má zobrazit přes celou plochu grafu, čímž dojde ke zvětšení zobrazení dané oblasti. Označení této oblasti probíhá pomocí kliknutí a tažení kurzoru myši směrem vpravo a dolů v zobrazené ploše grafu. Při této činnosti volení oblasti přiblížení grafu, je tato oblast zvýrazňována tmavší barvou, než jakou má pozadí grafu. Pro dokončení akce je nutné uvolnit tlačítko myši a aktuálně označená, tedy i zvolená oblast, se přiblíží a zobrazí přes celou plochu grafu. Zvýraznění této oblasti je možné vidět na obrázku 4.8, kde je zobrazen oddálený graf. Přiblížený graf je zobrazen na obrázku 4.9. Přiblížení oblasti grafu lze vrátit do původního stavu opačným postu-

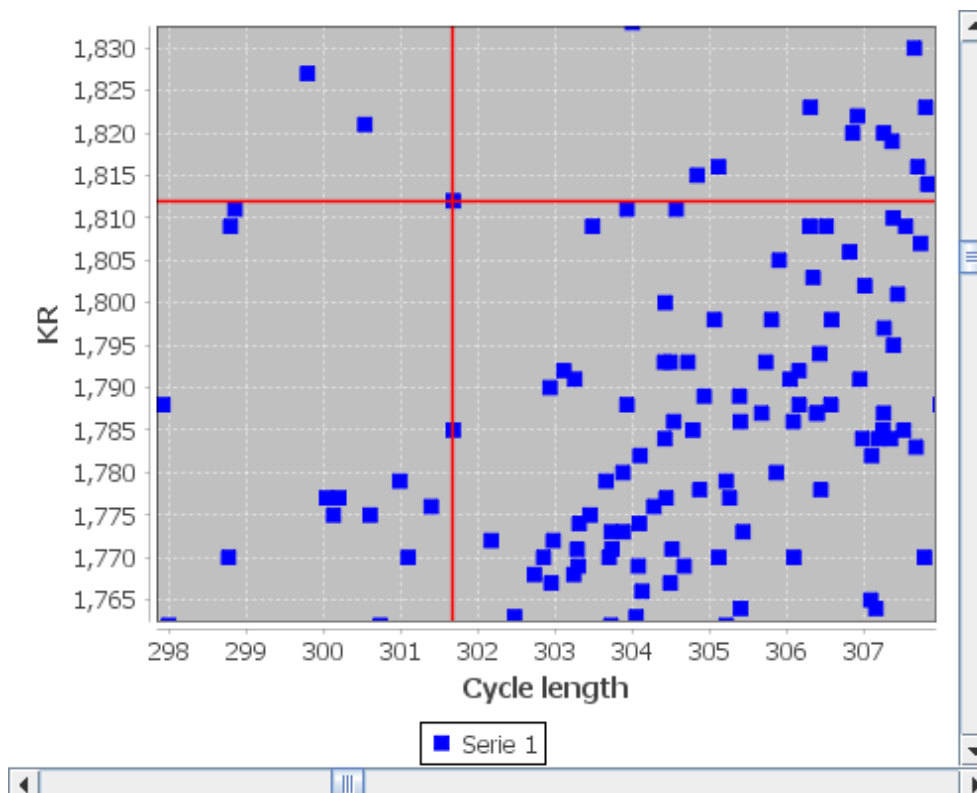
pem, tedy kliknutím a tažením kurzoru myši směrem nahoru a vlevo. Po uvolnění tlačítka myši se zobrazená oblast grafu vrátí do původní pozice. Při této akci však není zapotřebí zvýrazňovat označenou oblast grafu, jelikož se při této akci žádná oblast nevolí. Protože knihovna „JFreeChart“ implicitně neumožňuje uživatelsky přívětivější ovládání, byla ke grafu přidána možnost přibližovat a oddalovat graf i pomocí kolečka myši a posouvání zobrazenou oblast grafu po osách pomocí přidanych posuvníků u okraje grafu.



Obrázek 4.8: Ukázka oddáleného grafu s vyznačenou oblastí

Graf také zobrazuje kurzorový kříž. Pozice tohoto kříže je vždy automaticky zvolena na poslední místo v ploše grafu, na které bylo kliknuto kurzorem myši. Tento kurzorový kříž program využívá na výběr bodů v grafu. Po změně pozice kurzorového kříže se automaticky zvolí bod pod ním, na který tento kříž ukazuje. Jelikož se v grafu mohou v určitých oblastech vyskytovat body ve vyšší koncentraci, některé se mohou dokonce překrývat svou pozicí, bylo by obtížné zvolit jeden konkrétní požadovaný bod. Z tohoto důvodu je v programu zavedena proměnná s názvem „*SelectionRange*“, vymezující oblast do určité vzdálenosti od kurzorového kříže. V této oblasti jsou označeny všechny body grafu po změně pozice kurzorového kříže. Po označení bodů v

grafu jsou všechny označené body vypsané v seznamu bodů detailněji popsaném v sekci 4.7.8 s názvem „*Seznam bodů*“. Tato selekce bodů je realizována pomocí obsluhy události změny pozice kurzorového kříže. Následné předání označených bodů do komponentu se seznamem bodů probíhá ve formě odkazů na příslušné záznamy uložené v proměnné „*currentDataRecords*“. Vzhled kurzorového kříže, jehož barva byla zvolena jako kontrastní, oproti barvě bodů v grafu, je ukázán na obrázku 4.9.

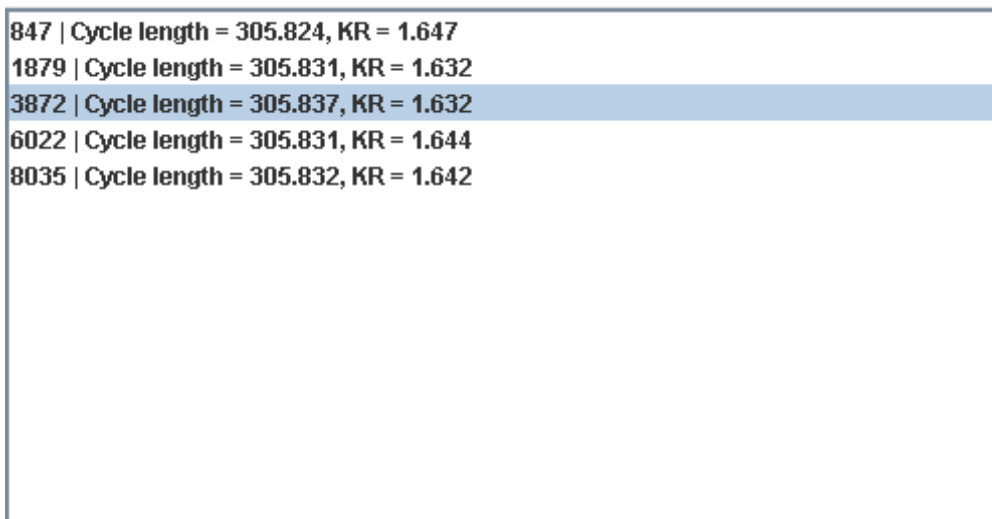


Obrázek 4.9: Ukázka přiblíženého grafu s kurzorovým křížem

4.7.8 Seznam bodů

Seznam bodů se vyskytuje v levé části hlavního okna programu, v oblasti pod grafem, a můžeme ho vidět na obrázku 4.10. Je tvořen komponentem nazvaným „*JListBox*“. Tento komponent umožňuje zobrazit seznam textových položek. Také je možné jakékoliv z těchto položek označit pomocí kliknutí na tyto položky. Tento seznam bodů tedy zobrazuje vybrané body z grafu

a jejich základní informace v textové podobě. Tyto body jsou do seznamu přidány ihned po označení v grafu pomocí kurzorového kříže. Tento seznam je aktualizován po každém dalším zvolení nových bodů v grafu. Staré položky jsou tedy přepsány novými body po každé aktualizaci. Při novém spuštění programu, či při uzavření aktuálně načtené relace či projektu programu, je tento seznam prázdný. Každá položka seznamu je zobrazena zvlášť, na jednotlivém řádku. V tomto řádku je jako první napsáno číslo řádku, ze kterého je tento záznam načten. Poté následuje znak „|“ a za ním text označující délku cyklu v různých jazykových variacích podle aktuálně nastavené jazykové lokalizace. Za tímto textem následuje znak rovnítka „=“ s hodnotou odpovídající délce cyklu pro tento záznam. Po této hodnotě se zde vyskytuje znak čárky „,“ a mezera. Za touto mezerou je zkratka KR a další znak rovnítka „=“. Za tímto druhým rovnítkem následuje hodnota KR pro daný záznam. Po kliknutí na některý z prvků v seznamu se tento prvek označí a pomocí zpracování události kliknutí na komponent „*JListBox*“ vytvoří a zobrazí program nové interní okno uvnitř interní plochy, vizualizující údaje právě pro tento vybraný záznam.



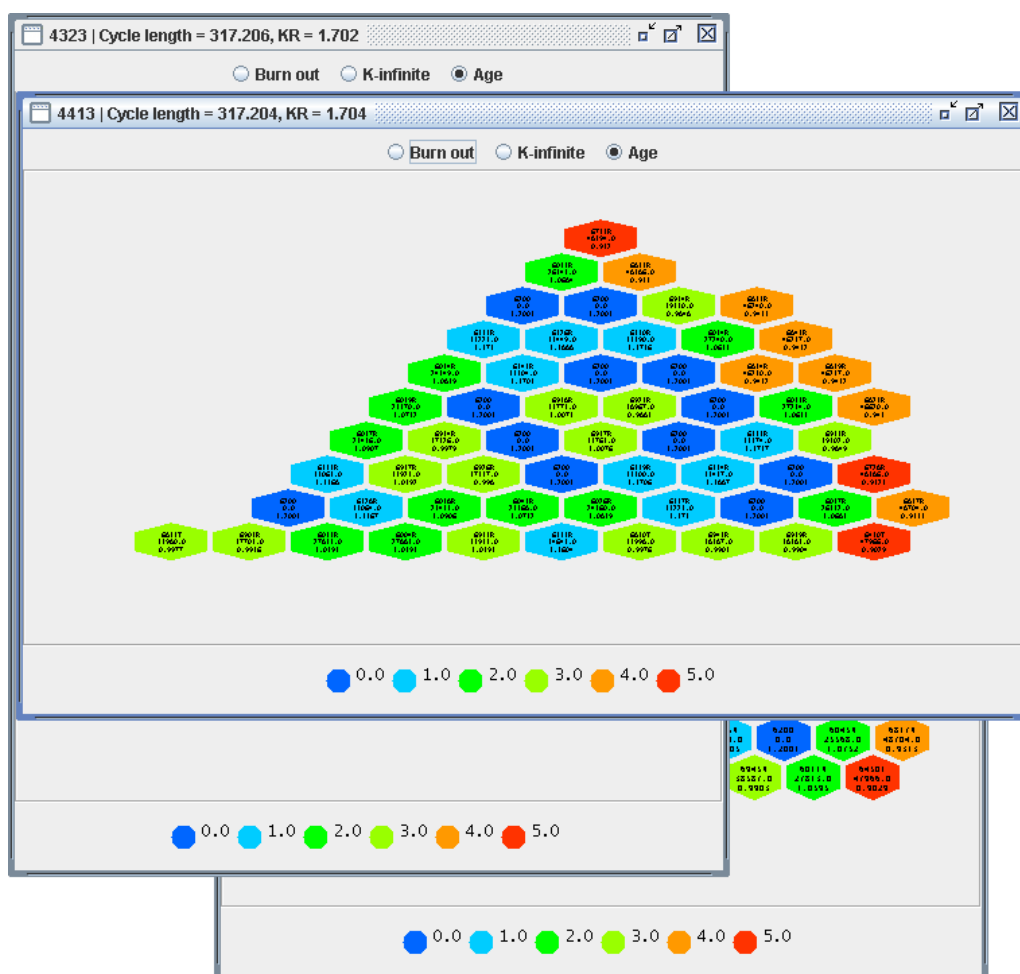
847		Cycle length = 305.824,	KR = 1.647
1879		Cycle length = 305.831,	KR = 1.632
3872		Cycle length = 305.837,	KR = 1.632
6022		Cycle length = 305.831,	KR = 1.644
8035		Cycle length = 305.832,	KR = 1.642

Obrázek 4.10

4.7.9 Interní plocha

Interní plocha je umístěna v pravé části hlavního okna programu. Slouží jako základ pro zobrazení jakéhokoliv počtu virtuálních, interních a různě se

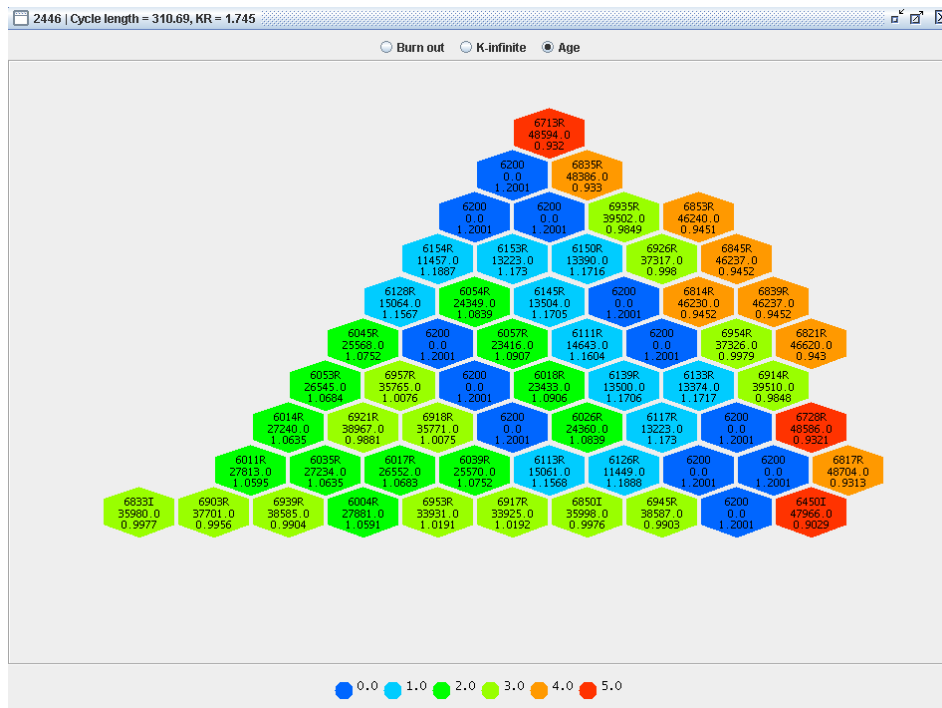
překrývajících oken. Pouze na této ploše se mohou zobrazovat interní okna uvnitř programu. Je tvořena komponentem s názvem „JDesktopPane“. Tento komponent je součástí knihovny „Swing“. Počet oken zobrazených v této interní ploše je omezen pouze pamětí. Velikost této plochy je možné upravovat pomocí oddělovačů, tedy komponentů s názvem „JSplitPane“, do kterých je v programu tato plocha zasazena. Vzhled takovéto interní plochy je možné si prohlédnout na obrázku 4.11.



Obrázek 4.11: Interní plocha

4.7.10 Interní okna

Interní okna jsou v programu zobrazena uvnitř interní plochy. Jsou tvořena komponenty typu „*JInternalFrame*“. Ačkoliv jsou tato okna interní a vyskytují se uvnitř jednoho hlavního okna programu, je možné s nimi provádět manipulace podobné oknům vyskytujících se přímo v operačním systému – například úprava rozměrů velikostí oken pomocí tažení za jejich okraje, čí minimalizace a maximalizace oken. Veškerou tuto funkcionalitu zaručuje právě komponent „*JInternalFrame*“. Dále bylo k tomuto oknu přidáno několik funkcionalit navíc, jako je například uchování poměru stran pomocí držení klávesy „*shift*“ při změně rozměrů okna. Také je v tomto okně možné vyvolat kontextové menu, které nabídne funkce jako je obnovení výchozího poměru stran, uložení vzhledu okna do obrázku, získání původních dat, které toho okno vizualizuje, minimalizovat ostatní okna, vyjma aktivního, zavřít ostatní okna, vyjma aktivního a najít v grafu bod, který dané okno vizualizuje.



Obrázek 4.12: Interní okno při obarvení podle stárí

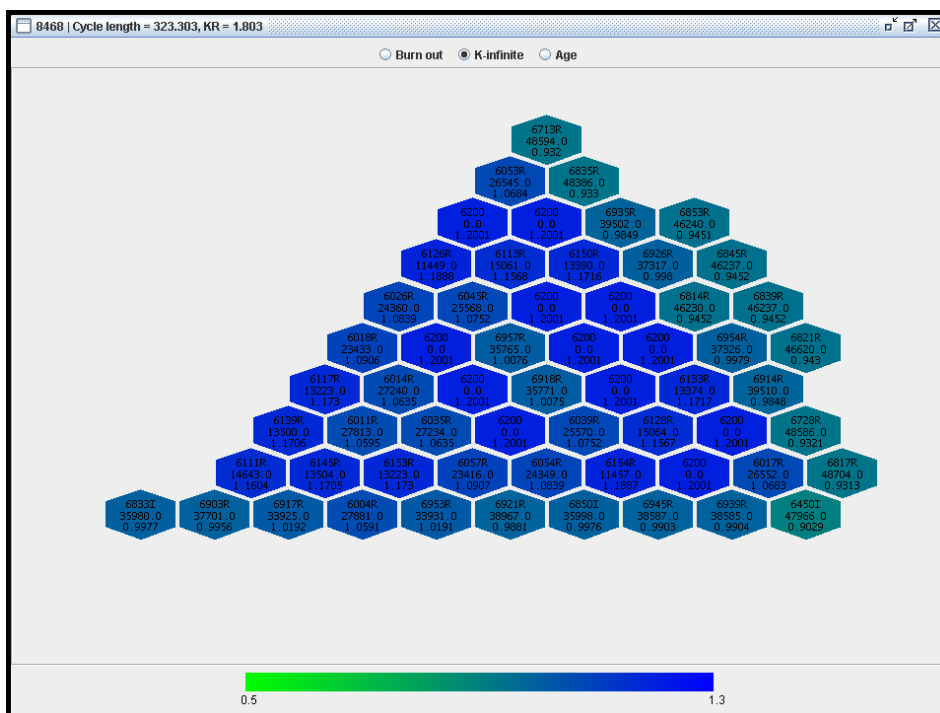
Toto okno reprezentuje vyobrazení vsázek části reaktoru. Jelikož je reaktor symetrický, není potřeba zobrazovat více než právě jednu z jeho symetrických částí. V každém tomto okně je vizualizována jedna řádka ze souboru typu „*current*“, tedy jeden bod v grafu a tedy jedna konkrétní palivová vsázka

v reaktoru. V nadpisu každého tohoto okna, umístěného v horní liště okna, je zobrazeno číslo aktuálně zobrazovaného řádku ze souboru typu „*current*“, které danou vsázku unikátně identifikuje. Pro informaci jsou zde ale zobrazeny i další užitečné údaje o zobrazené vsázce.

Tato vizualizovaná konfigurace kazet je zobrazena pomocí šestiúhelníkových tlačítek různé barvy, kde každé toto tlačítko obsahuje další tři řádky dodatečných informací o jednotlivé kazetě. Pozice těchto tlačítek závisí na druhu zobrazovaného reaktoru. Pozice pro umístění kazet pro daný reaktor jsou načteny ze souboru typu „*layout*“. Ovšem pro zjištění skutečného umístění, na které se mají tlačítka zobrazit, je zapotřebí k těmto pozicím zohlednit navíc velikost a poměr interního okna, ve kterém jsou tato tlačítka obsažena. Jelikož pozice pro kazety v zobrazované části reaktoru udávají pouze pozici v pomyslné mřížce, je nutné nejprve tuto mřížku sestavit na aktuální velikost a poměr interního okna a poté přepočítat skutečné umístění tlačítek pomocí souřadnic jednotlivých pozic v mřížce a zobrazit každé tlačítko na správnou pozici ve vypočtené mřížce. Nakonec, po vypočtení skutečného umístění tlačítek, je ještě před jejich vykreslením zapotřebí vycentrovat celý zobrazovaný obraz na střed interního okna. Při tomto zobrazování jsou samozřejmě upraveny i rozměry jednotlivých tlačítek aby odpovídaly rozměrům interního okna pořád ve stejném poměru. Vzhled takového interního okna je možné si prohlédnout na obrázcích 4.12, 4.13 a 4.14.

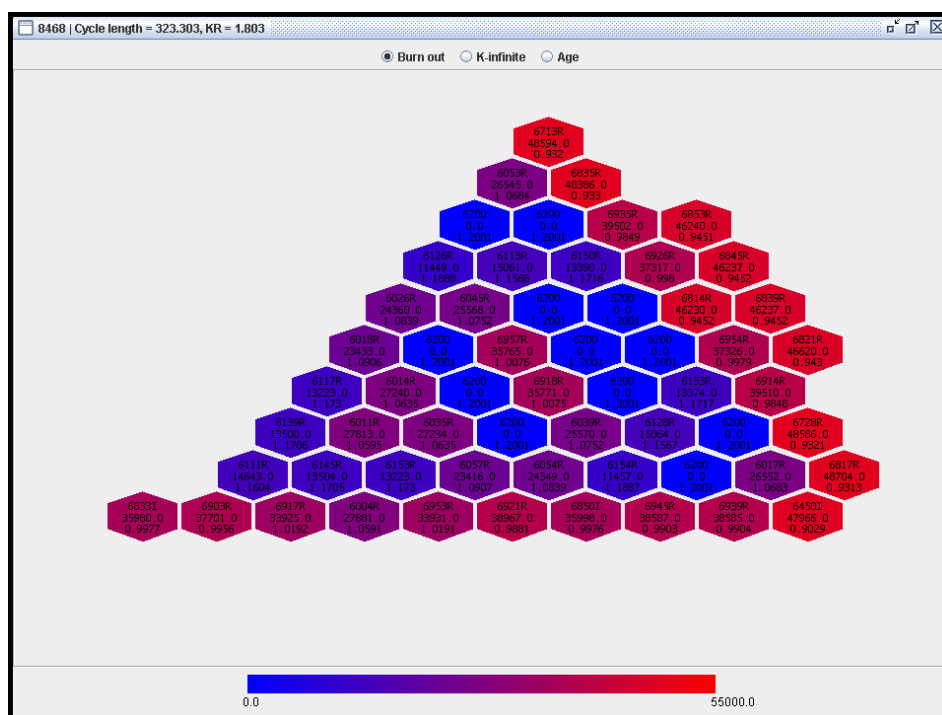
Každé ze zobrazovaných tlačítek má tvar šestiúhelníku a umožňuje jeho vybarvení zvolenou barvou. Dále je schopno rozpoznat kliknutí do oblasti uvnitř šestiúhelníku, nikoliv mimo něj, ačkoliv toto tlačítko vychází z komponentu s názvem „*JButton*“, jehož původní defaultní tvar je obdélník. Pro tyto šestiúhelné tlačítka je proto vytvořen speciální komponent s názvem „*HexButton*“, který dědí, a v případě potřeby upravuje, důležité vlastnosti z komponentu „*JButton*“. Tento komponent „*HexButton*“ navíc umožňuje zobrazení tří textů. Každý z těchto textů je zobrazen zvlášť na jednu řádku a je horizontálně vycentrován do středu tlačítka. V programu tyto tři texty obsahují konkrétní údaje ohledně vyobrazené kazety. Text na první řádce, tedy první a vertikálně umístěný nejvýše ze všech tří textů, obsahuje název kazety. Druhý, tedy prostřední text, zobrazuje hodnotu vyhoření a třetí, poslední a nejnižší položený text, obsahuje hodnotu K-nekonečno. Vzhled tohoto šestiúhelného tlačítka je možné si prohlédnout na obrázku 4.15.

Způsoby obarvení tlačítek jsou tři, podle stáří vsázky, podle vyhoření a podle hodnoty K-nekonečno. Tyto způsoby obarvení tlačítek lze přepínat pomocí přepínacích tlačítek v liště v horní části interního okna. Všechna



Obrázek 4.13: Interní okno při obarvení podle K-nekonečno

tato přepínací tlačítka jsou popsána výstižnými lokalizovanými texty a jsou tvořena pomocí komponenty s názvem „*JRadioButton*“, která je opět součástí knihovny „*Swing*“. Tato přepínací tlačítka jsou vyobrazena na obrázku 4.16. Barevná škála pro obarvování tlačítek pomocí vyhoření a hodnoty K-nekonečno je rozprostřena lineárně, kde jsou barvy jednotlivých tlačítek vypočítány pomocí interpolace vizualizované hodnoty mezi dvěma barvami, zatímco barevná škála pro obarvování tlačítek pomocí stárí je rozložena diskrétně. Konkrétní barva pro zobrazované tlačítko je tedy vybrána ze sady přednastavených barev pomocí hodnoty, jež je k této barvě přiřazena. Kompletní vzhled celého interního okna pro všechny tři typy obarvení je možné si prohlédnout na obrázcích 4.12, 4.13 a 4.14. Aktuálně použité barvy, i s jejich hodnotami, jsou znázorněny ve vysvětlující legendě zobrazené ve spodní části interního okna. Pro zobrazení této legendy je v programu vytvořen speciální komponent s názvem „*HexLegend*“, který dědí všechny užitečné vlastnosti z třídy „*JComponent*“. Vzhled této legendy pro všechny tři typy obarvování jsou zobrazeny na obrázcích 4.17, 4.18 a 4.19



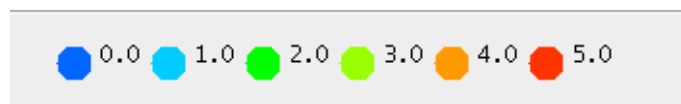
Obrázek 4.14: Interní okno při obarvení podle vyhoření



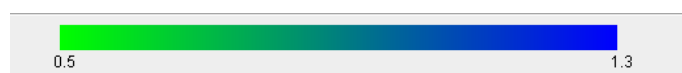
Obrázek 4.15: Šestiúhelné tlačítko



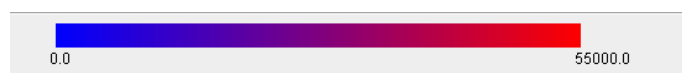
Obrázek 4.16: Přepínací tlačítka na volbu obarvení



Obrázek 4.17: Legenda interního okna pro obarvování podle stáří



Obrázek 4.18: Legenda interního okna pro obarvování podle hodnoty K-
nekonečno



Obrázek 4.19: Legenda interního okna pro obarvování podle vyhoření

5 Závěr

Při tvorbě tohoto programu se vyskytla komplikace s neznalostí některých technologií a knihoven, například knihovna „*JFreeChart*“. Tento problém byl řešen prohledáváním internetových zdrojů, které posloužily k nalezení užitečných a potřebných metod a tříd knihovny „*JFreeChart*“, které mohly být využity k plánovaným záměrům. Hlavním zdrojem informací o těchto metodách a třídách byly především vzorové příklady a úseky kódů, ve kterých byla ukázána jejich funkčnost a použitelnost.

Program by bylo možné do budoucna rozšířit o funkcionalitu cyklického přenačítání zobrazených dat a přidávání nových bodů do grafu. Toto rozšíření by umožnilo uživateli přistoupit i k datům, o kterých program nevěděl při načítání relace nebo při vytváření nového projektu. Přístup k této možnosti by mohl být přes hlavní menu programu. Dále by bylo možné program rozšířit o další jazyk, podle kterého by tento program uměl lokalizovat zobrazené texty. Tato funkce by byla přístupná z dialogového okna pro výběr jazyků lokalizace, do kterého by přibyl tento jeden jazyk navíc.

Literatura

- [1] *Java™ Platform, Standard Edition 7 API Specification* [online]. [cit. 18.4.2015]. Dostupné z: <http://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html>.
- [2] *AWT vs Swing* [online]. [cit. 19.4.2015]. Dostupné z: <http://edn.embarcadero.com/article/26970>.
- [3] *Introduction to the AWT* [online]. [cit. 19.4.2015]. Dostupné z: <http://www.javaworld.com/article/2077188/core-java/introduction-to-the-awt.html>.
- [4] *Introduction Java Swing* [online]. [cit. 12.4.2015]. Dostupné z: <http://zetcode.com/tutorials/javaswingtutorial/introduction/>.
- [5] *The History of Java Technology* [online]. [cit. 19.4.2015]. Dostupné z: <http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>.
- [6] *JFreeChart Samples* [online]. [cit. 19.4.2015]. Dostupné z: <http://www.jfree.org/jfreechart/samples.html>.
- [7] *JFreeChart* [online]. [cit. 19.4.2015]. Dostupné z: <http://www.jfree.org/jfreechart/>.
- [8] *Learn About Java Technology* [online]. [cit. 19.4.2015]. Dostupné z: <https://www.java.com/en/about/>.
- [9] *Úvod do formulářových aplikací v Java Swing* [online]. [cit. 19.4.2015]. Dostupné z: <http://www.itnetwork.cz/java-tutorial-swing-okenni-aplikace-uvod>.

-
- [10] *What is Java technology and why do I need it?* [online]. [cit. 19.4.2015]. Dostupné z: https://www.java.com/en/download/faq/whatis_java.xml.
- [11] *XML (Extensible Markup Language)* [online]. [cit. 19.4.2015]. Dostupné z: <http://searchsoa.techtarget.com/definition/XML>.
- [12] *Extensible Markup Language (XML)* [online]. [cit. 19.4.2015]. Dostupné z: <http://www.w3.org/XML/>.
- [13] *XML základy* [online]. [cit. 19.4.2015]. Dostupné z: <http://www.tvorba-webu.cz/xml/>.

A Uživatelská příručka

A.1 Návod na spuštění

Program je možný spustit na všech zařízeních s nainstalovaným virtuálním strojem „Javy“. Díky tomu, že je celý program napsán v programovacím jazyce „Java“, je tímto způsobem možné ho spustit na většině zařízení.

Program má název „*GUI.jar*“ a je možné ho spustit buďto poklikáním na tento soubor, nebo přes příkazovou řádku příkazem „*java -jar GUI.jar*“. Aby bylo možné program takto spustit, musí se ve stejném adresáři, jako se nachází tento soubor, nalézat i složka „*lib*“, která obsahuje dvě knihovny „*jcommon-1.0.23.jar*“ a „*jfreechart-1.0.19.jar*“.

A.2 Načtení dat

Načtení dat pro jejich následné zobrazení je možné dvěma způsoby.

Prvním z nich je vytvořit nový projekt. Tuto možnost uživatel nalezne v menu programu pod tlačítkem „*Projekt*“, kde se nachází položka „*Nový projekt*“. Toto menu se nalézá v horní části hlavního okna programu. Tento projekt se vytváří pomocí projektového souboru, ve kterém musí být vypsány všechny potřebné názvy datových souborů, které se musí nalézat ve stejném adresáři jako tento projektový soubor. Tento projektový soubor může být bez problémů vytvořen uživatelem tohoto programu a musí splňovat následující podmínky.

Musí obsahovat názvy třech datových souborů. Každý z těchto souborů musí být zapsán na samostatné řádce a před názvem každého tohoto souboru musí existovat konstanta oddělená mezerou od tohoto názvu souboru. Tato konstanta udává, o jaký typ datového souboru se jedná. Její možné hodnoty jsou „*SOLUTIONS_CURRENT*“, „*FUEL_LIBRARY*“ a „*LAYOUT*“. Tento projektový soubor může, ale nemusí, obsahovat také komentář. Ten se vyskytuje vždy za znakem „*%*“, který musí být vždy umístěn na začátku řádku.

Druhým způsobem, jak načíst data, je načtením relace. Relační soubor, ze kterého je možné načíst relaci, obsahuje kromě datových souborů, i nastavení programu a stav oken. Po načtení této relace ze souboru je tedy program převeden do stavu uloženém v tomto relačním souboru. Samozřejmě jako v předchozím případě, u projektového souboru, i zde platí, že se datové soubory musí pro správné načtení vyskytovat ve stejném adresáři jako tento relační soubor. Relační soubor není tak jednoduché vytvořit manuálně, jako projektový soubor, ale je možné ho vytvořit automaticky pomocí možnosti uložení relace programu. Po uložení relace se aktuální stav a nastavení programu uloží do zvoleného relačního souboru. Obě možnosti, jak načtení, tak uložení relace, se nacházejí v menu, v horní části hlavního okna programu, pod tlačítkem s názvem „*Relace*“.

Je možné přinutit program přestat zobrazovat data a následně uzavřel všechny datové soubory – pomocí možnosti „*Zavřít projekt*“ v menu programu pod tlačítkem „*Projekt*“.

A.3 Změna nastavení

V programu je možné změnit některé jeho nastavení. Všechny dialogové nabídky pro změnu některého nastavení se vyskytují pod tlačítkem s názvem „*Nastavení*“ v menu programu.

Jedním z nastavení, které je v programu možné upravit, je jazyková lokalizace. Jazyk této lokalizace je možný zvolit v dialogovém okně, které se zobrazí po stisknutí tlačítka s názvem „*Změnit jazyk*“. Na výběr je zde ze dvou jazyků – angličtina a čeština. Po zvolení jazyka a potvrzení tlačítkem „*Použít*“ v tomto dialogovém okně se veškeré texty obsažené v programu ihned lokalizují pro zvolený jazyk.

Dalším možným nastavením programu, které je možné upravit, je nastavení proměnné s názvem „*SelectionRange*“. Tato proměnná udává rozsah oblasti okolo kurzorového kříže v grafu, ve které se ještě mají body, po kliknutí na graf, zvolit za označené. Tato oblast se nastavuje v nabídce s názvem „*Změnit výběr*“. Po kliknutí na tuto nabídku se zobrazí dialogové okno s políčky, do kterých je možné napsat velikost rozměrů požadované oblasti. Po uložení změn v nastavení kliknutím na tlačítko „*Použít*“, je toto nastavení v programu ihned používáno.

A.4 Ovládání komponentů

Jednou z nejdůležitějších součástí programu je graf. Tento graf má náležitě popsané osy, takže se v něm uživatel snadno zorientuje. Graf lze přibližovat kliknutím a tažením kurzoru směrem doprava dolů. Tím dojde ke zvýraznění přibližované oblasti. Po uvolnění tlačítka myši se tato oblast požadovaným způsobem přiblíží. Oddálení grafu probíhá podobným způsobem, s tím rozdílem, že je nutno myší táhnout doleva a nahoru. Toto přiblížení a oddálení je možné uskutečnit i pomocí otáčení kolečkem myši. Dále tento graf umožňuje posun vyobrazeného úseku po osách pomocí postranních posuvných lišt.

V grafu je dále možné označit vyobrazené body. To je docíleno kliknutím na některý z dostupných bodů v grafu. Body jsou označeny v určité oblasti od kliknutí. Tuto oblast lze změnit a postup této změny je popsán v sekci A.3. Po označení bodu se všechny body, v jeho zvoleném okolí, vypíší v seznamu umístěném pod grafem.

Ve výše jmenovaném seznamu bodů, umístěném pod grafem, vidí uživatel nejdůležitější informace o zvolených bodech. Po kliknutí na kterýkoliv z těchto bodů se v pravé části hlavního okna programu otevře interní okno, popisující právě vybraný bod. Těchto oken může být otevřeno i několik zároveň, což z pohodlní a zřehlední práci s nimi. S vybraným oknem lze různě, v rámci vymezené plochy, pohybovat, měnit jeho velikost či volit u jeho zobrazení ze tří různých barevných variant, kterými jsou „Stáří“, „K-nekonečno“ a „Vyhoření“. Okno lze dále příslušnými tlačítky, v jeho pravém horním rohu, maximalizovat, minimalizovat či úplně zavřít.

Pokud je otevřeno více oken najednou, byla do menu v horní liště aplikace přidána záložka „Okna“, která ulehčí práci s těmito okny. První volbou v této záložce je „Zavřít všechna okna“, po jejímž stisknutí dojde k zavření všech vpravo otevřených oken. Další volbou je „Minimalizovat všechna okna“. Pomocí této volby lze všechna otevřená okna jednoduše minimalizovat na spodní hranu části programu, kde probíhá práce s okny. Volba „Mozaika“ zobrazí všechna otevřená okna do přehledné mozaiky, ve které velikost oken mozaiky závisí na počtu otevřených oken. Pomocí další volby, „Nastavit obarvování všem“, lze pro všechna otevřená okna nastavovat jejich stejné zabarvení (s možností voleb „Stáří“, „K-nekonečno“ a „Vyhoření“).

Poslední volbou v této záložce je „Aktivní okno“. Ta slouží pro manipulaci s právě označeným oknem a její možnosti jsou shodné s těmi, kterých lze dosáhnout pravým stisknutím tlačítka nad právě zvoleným oknem. Zde lze

volit opět z několika možností. Těmi jsou:

Uložit obrázek

Pomocí této volby lze vyobrazení označeného okna uložit jako obrázek. Ten lze uložit jako obrázek s příponou JPG či PNG.

Zkopírovat původní data

Tato možnost zobrazí a do schránky zkopíruje popis aktivního okna, který je dán daty v souboru „*Current*“.

Najít v grafu

Zvolením této možnosti dojde v grafu, v levé části programu, k označení pozice právě vybraného okna. To je užitečné například v situaci, kdy uživatel po otevření aktuálního okna prohlížel jiné oblasti grafu.

Obnovit poměr stran

Jelikož si lze velikost oken libovolně přizpůsobovat, je zde možnost nechat si u okna obnovit poměr stran, který vrátí zobrazení okna zpět do původního čtvercového tvaru.

Zavřít ostatní

Touto volbou lze jednoduše zavřít všechna okna, která momentálně nejsou aktivní. Tedy všechna okna vyjma toho, pomocí kterého je tato volba vyvolána.

Minimalizovat ostatní

Tato možnost funguje podobně jako možnost *Zavřít ostatní*, s tím rozdílem, že ostatní okna pouze minimalizuje ke spodní hraně pravé části programu, nedojde tak k jejich zavření.