Martin Sloup, A04372
msloup@students.zcu.cz

ZADÁNÍ

Navrhněte a vytvořte program pro analýzu logovacích souborů z předzápisu. Vstupem programu bude xml soubor ze seznamu akcí, který provedl uživatel během předzápisu. Úkolem programu je zjistit časy mezi dvěma požadavky uživatele a tyto časy přehledně zobrazit pomocí grafického histogramu. Uživatel musí mít možnost si navolit mez a krok histogramu.

STRUČNÁ ANALÝZA

Program je tvořen třívrstvou architekturou. Obsahem je datová, aplikační a prezenční vrstva.

Základem datové vrstvy je interface Loader, od kterého jsou následně zděděny třídy DOMLoader, SAXLoader a TXTLoader. První dvě slouží k načítání losovacího souboru ve formátu xml. První z nich používá DOM parser, druhá SAX parser. Zbylá třetí není v konečném programu použita – ta sloužila pouze pro ladění histogramu za pomocí načítání textového programu. Všechny tři vrstvy vrací rozebraný logovací soubor v podobě ArrayListu.

Tento rozebraný obsah se dále zpracovává v aplikační vrstvě, ve třídě histogram. Tato třída provádí hledání správné sekvence (celkem tři možné sekvence). Po nalezení správné sekvence je zapamatován čas začátku sekvence u uživatele, který tuto sekvenci vytvořil. Pokud je již nějaký čas zapamatován a rozdíl starého a nového času je menší než hodnota disconnectTimeotu (doba po kterou je uživatel veden jako přihlášený), je tento rozdíl času zapsán do tabulky (pozice tabulky je rozdíl času a hodnota je zvednuta o jedničku). Tímto se vytvoří tzv. „neminimalizovaný“ histogram, který následně zminimalizuju použitím parametrů n (mez histogramu) a k (krok histogramu). Toto provádím z důvodu, abych nemusel projíždět sekvence znova. Tím pádem, pro jiné hodnoty n a k, stačí provést pouhé „minimalizování“ histogramu. Takto minimalizovaný histogram se předá do Prezenční vrstvy. Aplikační vrstva dále obsahuje i třídu Configuration, která se stará o uchování nastavení programu.

V Aplikační vrstvě minimalizovaný histogram zpracuje třída GraphicHistogram. Ta se postará o správné vykreslení hodnot histogramu, případně dokreslení dalších prvků grafu. Obsahem aplikační vrstvy je i třída MainWindow, která se snaží ovládání programu zpříjemnit použitelným GUI.

POPIS OVLÁDÁNÍ PROGRAMU

Možnosti zpuštění

Program je možné spustit jak se zadáním parametrů z příkazové řádky, čímž se docílí okamžité načtení souboru, tak i bez zadání parametrů.

Martin Sloup, A04372
msloup@students.zcu.cz

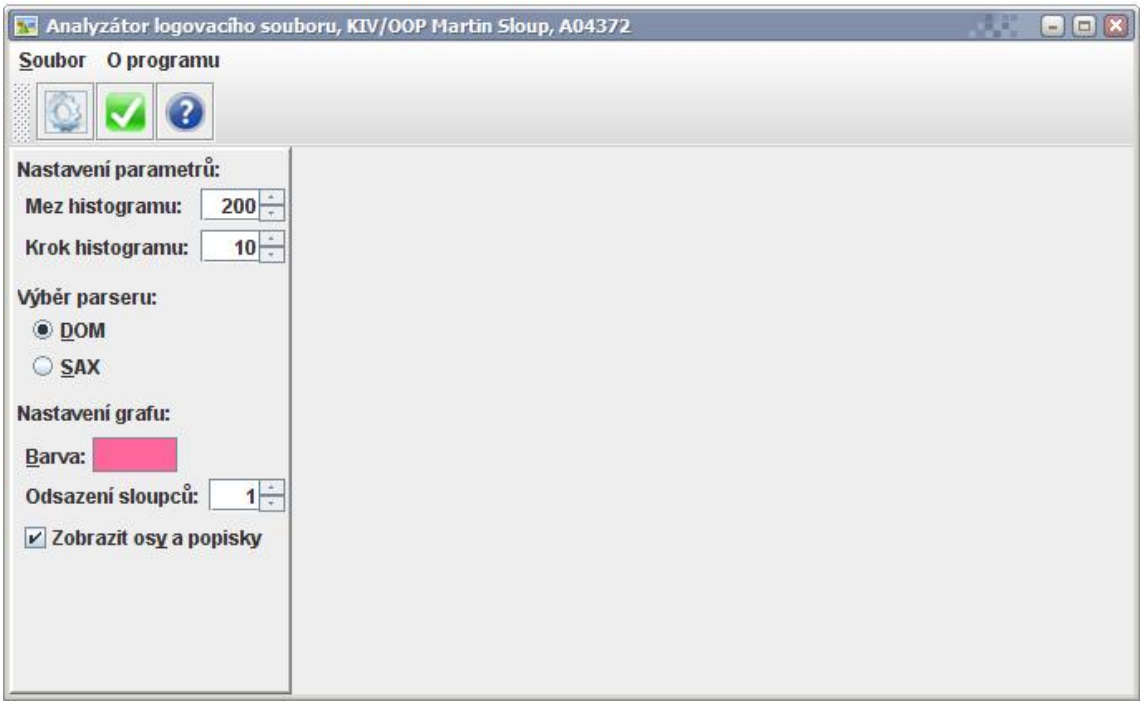
Spuštění bez parametrů:
```
java -Xmx200M -jar oop.jar
```

Spuštění s parametry:
```
java -Xmx200M -jar oop.jar (d|s) soubor_s_logem
```

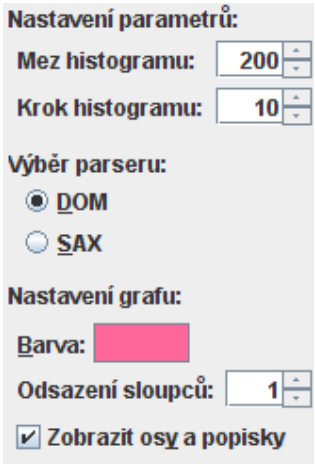
kde d nebo s určuje výběr parseru (d=DOM, s=SAX) a soubor_s_logem je vstupní soubor, který je nutné analyzovat.

Např:
```
java -Xmx200M -jar oop.jar s c:\data-win.xml
```

Popis obrazovek



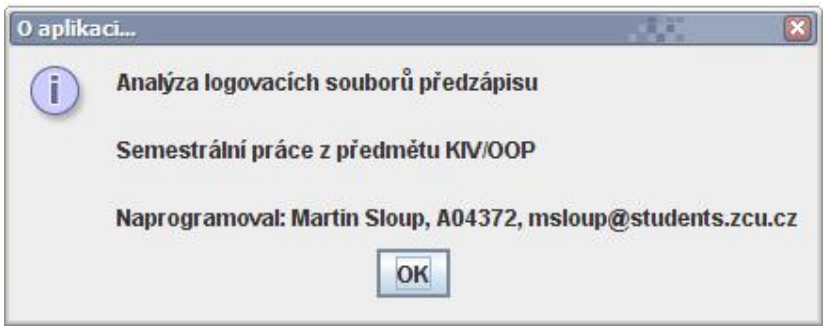
Okno programu



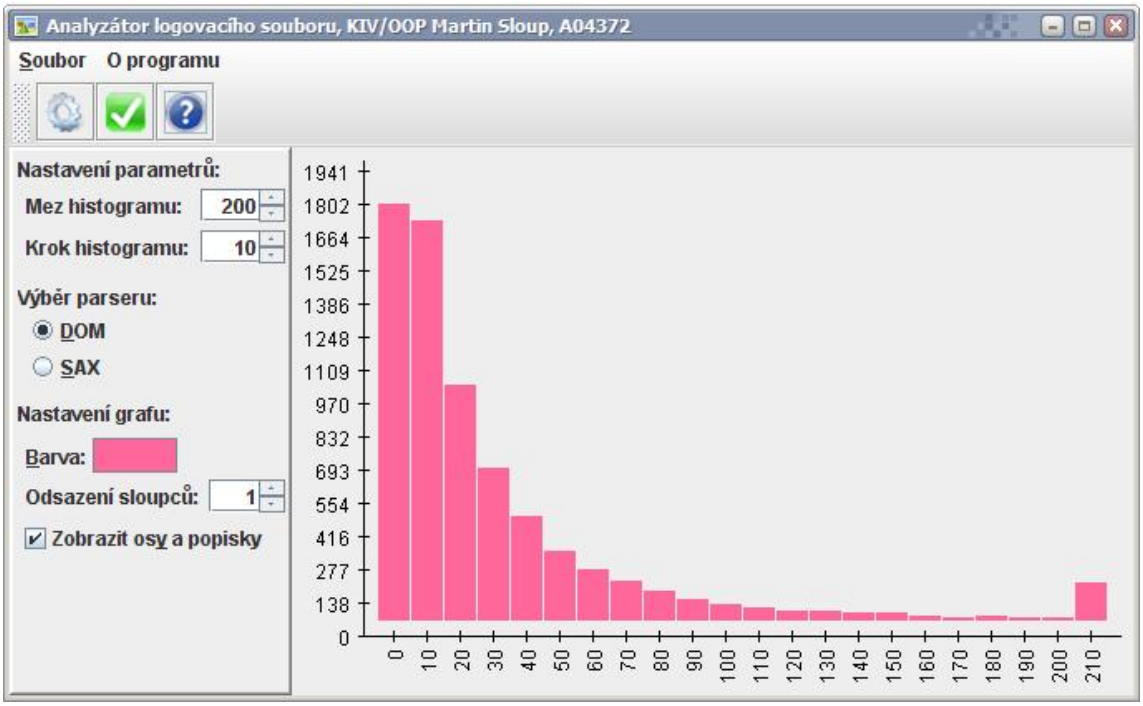
Program obsahuje podle zadání všechny důležité ovládací prvky. Načtení souboru se provede přes menu Soubor-> Vstupní soubor, kdy parsování logu bude provedeno metodou vybranou v levém panelu programu. Tento panel dále obsahuje kromě výběru parseru i možnost volby meze a kroku histogramu, volby barvy sloupců grafu, odsazení sloupců gramu a zobrazení, případně skrytí os a popisků os grafu.

Martin Sloup, A04372
msloup@students.zcu.cz

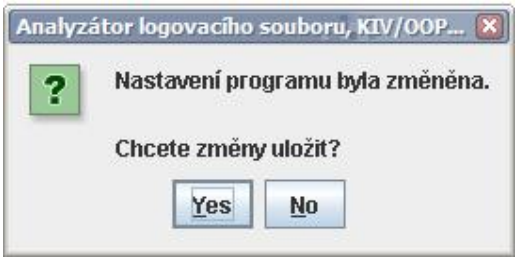
Program taktéž obsahuje tlačítkovou nabídku nástrojů. První tlačítko slouží pro zobrazení nebo skrytí levého panelu nastavení. Druhé provádí validaci konfiguračního souboru a poslední, třetí, zobrazí informace o tvůrci programu.



Dialog zobrazující informace o tvůrci programu.



Program v akci.



Pokud je změněno nastavení programu, uživatel bude dotázán, zda chce nastavení uložit.

Martin Sloup, A04372
msloup@students.zcu.cz

ZPRACOVÁNÍ

Program byl vytvořen pomocí nástroje EasyEclipse Desktop verze 1.2.1 s použitím VisualEditoru verze 1.2.1, kterým mi pomohl při vytváření GUI programu.

Při vývoji jsem občas měl problémy s prostředím Eclipse, jelikož jsem se snažil vyvíjet i na místech kde zrovna nebyl výkonný počítač po ruce. Samotný elipse je docela dost „nenažraný“ a pro svou práci potřebuje nejméně 512 MB ram a trochu slušnější procesor. Jediné co jsem nedokázal pochopit, proč tak dlouho trvá, než vyfiltruje Content Assist. Hodně to bylo znát na slabších strojích, kde zobrazení samotného okna Content Assist trvalo i nepříjemných 10 sekund. Osobně hodně programu ve Visual Studiu 2005 a tam je podobná funkce daleko příjemnější a použitelnější. Jinak na druhou stranu je Eclipse vesměs dobrý vývojový prostředí.

Jako testovací stroj jsem použil, ten kde byla semestrální práce převážně tvořena. Tento stroj obsahuje procesor AMD Athlon XP 2500+ Barton o velikosti paměti 1GB ram. Rozlišení monitoru bylo po celou dobu na konstantních 1152x864.

PŘÍNOSY, POTÍŽE A NÁZORY

Tato semestrální práce byla vesměs přínosná. Každopádně bych třeba ocenil možnost získání bonusových bodů za určité nástavby programu, tak jako to třeba bylo u předmětu KIV/ZPG. Na druhou stranu jsem se díky této semestrální práci konečně naučil tvorbu UML diagramů.

Samotný vývoj algoritmu histogramu mi nečinil problém. Občas bylo nutné některé konstrukce hledat po internetu, čímž pomohl Google. Párkrát jsem se i potrápilo s validací xml dokumentu, kdy jsem nemohl za boha přijít na kloub, proč to z vývojového prostředí Eclipse nehlásí žádnou chybu a při spuštění z jaru chybu při validaci, že se mu nepovedlo najít kořenový element. Nakonec jsem i toto zdárně vyřešil.

ZÁVĚR

Na testovacím stroji trvá výpočet histogramu za pomocí SAX parseru něco kolem 2 vteřin. S použitím DOM parseru se již čas vyšplhá na 8 vteřin. Prostředí bylo navrženo pro horní mez 200 a krok 10. Při extrémních hodnotách horní meze a kroku může dojít k nepříjemnému překrytí značek u os. Okno programu je možně jakkoliv zvětšovat a zmenšovat, GUI se samo přizpůsobí.