

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Diagramatická konverzia súborov

2014

János Farkas

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFOMATIKY

Diagramatická konverzia súborov

Bakalárska práca

Študijný program : Aplikovaná informatika

Študijný odbor: 9.2.9 Aplikovaná informatika

Školiace pracovisko: Katedra aplikovanej informatiky

Školiteľ: Mgr. Pavel Petrovič, PhD.

Bratislava 2014

János Farkas



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: János Farkas
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.9. aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský

Názov: Diagramatická konverzia súborov / *File Conversion Using Diagrams*

Cieľ: Navrhnuť a naimplementovať vizuálny formalizmus, grafický editor, i interpreter pre diagramatickú konverziu súborov. Diagramatická konverzia súborov je popísaná konverzným diagramom v architektonickom štýle pipes and filters. Konverzné bloky (filtre) sú poprepájané dátovodmi so zdrojmi (source) a cieľmi (sink/target) dát. Úlohou študenta je vymyslieť (navrhnuť a implementovať) vhodnú sadu filtrov a ich zrozumiteľnú a prehľadnú vizualizáciu, tak aby bolo možné vytvárať rozličné jednoduché i zložité konverzie pracujúce nad jedným alebo i viacerými súbormi. K aplikácii by mala existovať možnosť spustenia konverzie v dávkovom režime a rozhranie z príkazového riadku. Bakalárska práca nadväzuje na prototyp [1], úlohou študenta je analyzovať jeho nedostatky a vypracovať nový návrh a použiteľnejšie riešenie.

Literatúra:

1. Adam Paldauf: Vizualný jazyk na konverziu súborov, bakalárska práca, FMFI UK, 2011.
2. The UNIX and Linux Formus: GNU sed(1) manual page, prístupné online: www.unix.com/man-page/linux/1/sed/
3. The UNIX and Linux Formus: GNU awk(1) manual page, prístupné online: www.unix.com/man-page/linux/1/awk/
4. Regular-Expressions.info: Regular Expressions Tutorial, prístupné online: www.regular-expressions.info/tutorial.html

Kľúčové

slová: ikonografický jazyk, konverzia súborov, pipes and filters

Vedúci: Mgr. Pavel Petrovič, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. PhDr. Ján Rybár, PhD.
Dátum zadania: 24.10.2013

Dátum schválenia: 24.10.2013

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

Farkas

študent

Pavel Petrovič

vedúci práce

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím citovaných zdrojov.

.....

Pod'akovanie

Moja vďaka patrí môjmu školiteľovi, Mgr. Pavlovi Petrovičovi, PhD. za odbornú pomoc pri vypracovaní práce, za jeho rady a pripomienky, ktoré boli pre mňa cenným prínosom.

Abstrakt

Farkas, János: Diagramatická konverzia súborov (bakalárska práca)

Univerzita Komenského v Bratislave. Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej informatiky.

Vedúci práce: Mgr. Pavel Petrovič, PhD

FMFI UK, 2014

Cieľom mojej práce bolo navrhnúť a implementovať textový editor, ktorý umožňuje spracovávať textové súbory podľa grafického predpisu. Pri návrhu som analyzoval bakalársku prácu Adama Poldaufa. Výsledná aplikácia umožňuje vytvorenie konverzie súboru v grafickom prostredí, jej vykonávanie, uloženie pre ďalšie použitie.

Kľúčové slová: ikonografický jazyk, konverzia súborov, pipes and filters.

Abstract

Farkas, János: File Conversion Using Diagrams (Bachelor thesis),

Comenius University in Bratislava. Faculty of Mathematics, Physics and Informatics.

Department of Applied Informatics.

Supervisor: Mgr. Pavel Petrovič, PhD

The goal of my bachelor thesis is to design and implement tool which is capable of creating visual scripts for file processing and also process files with created scripts. Created application enables user to create visual script for file processing, execution of created script, and save it for next use.

Keywords: iconographic language, file conversion, pipes and filters

Obsah

1. Úvod	10
1.1. Cieľ bakalárskej práce	10
2. Analýza východísk	10
2.1. (G)AWK	10
2.2. SED (stream editor)	11
2.3. LabView	12
2.4. Adam Poldauf: Vizuálny jazyk na konverziu súborov, bakalárska práca, FMFI UK, 2011	13
2.5. Použité technológie	13
2.5.1 JAVA	13
2.6. Užívateľské rozhranie	14
2.6.1. JavaFX Scene Builder	14
3. Funkcionalita systému	14
3.1. Požiadavky na systém	14
3.2. Grafická konverzia	15
3.3. Bloky	15
3.3.1. Source bloky	15
3.3.2. Manipulačné bloky	16
3.3.3. Target bloky (output)	17
4. Návrh riešenia	18
4.1. Členenie návrhu	18
4.2. Triedy	18
4.2.1. Block	18
Hlavnou úlohou triedy je vykreslenie blokov na grafickú plochu.	18
Konštruktor- má 3 vstupné parametre, 2x integer (x a y súradnice) a 1x string (typ bloku).	18
4.2.2. Pipe	19
4.2.3. SL_conv	19
4.2.4. CM	19
5. Implementácia	21
5.1. GUI	21
5.1.1. TabPane blokov	21

5.1.2. mainAnchorPane	21
5.1.3. options_panel.....	22
5.1.4 minimap.....	22
5.2. Vykónávanie konverzie z príkazového riadku	22
5.3. Príklad použitia	23
5.3.1. Jednoduchý filter	23
5.3.2. Priemerná hodnota riadku	24
5.3.3. Unix –Dos konverzia	26
5.3.4 Práca s viacerými súbormi.....	27
6. Záver	28
7. Použitá literatúra a zdroje	29

1. Úvod

1.1. Cieľ bakalárskej práce

Navrhnuť a naimplementovať grafický editor, i interpreter pre diagramatickú konverziu súborov. Príklad: textový súbor obsahuje zoznam študentov (meno, priezvisko, login, rok narodenia, atď.) a chceme mená a priezviská zlúčiť s údajom o známke študenta (spolu s priemerom), ktorý sa nachádza v inom súbore, čo obsahuje len priezviská a známky. Pomocou programu môžeme „nakresliť“ konverziu, vypočítať priemer a uložiť konverziu do xml súboru. Vytvorenie konverzie by malo byť omnoho jednoduchšie, ako vytvorenie skriptu pre sed alebo awk.

Podľa dostupných informácií neexistuje program, ktorý by umožňoval spracovanie textových súborov podľa grafického predpisu. Existuje mnoho nástrojov, ktoré riešia konverziu súborov pomocou komplexných predpisov (ako sú awk alebo sed), a cieľom je vytvoriť aplikáciu pre ľahšiu prácu.

2. Analýza vychodísk

2.1. (G)AWK

AWK je programovací jazyk, čo umožňuje užívateľovi manipulovať so súbormi, ktoré sú štrukturované ako stĺpce dát a reťazce. V roku 1977 bola vytvorená prvá verzia. Najväčšia výhoda oproti SED-u, že môžeme použiť premenné. ^[1]

AWK obsahuje v sebe built-in funkcie, taktiež umožňuje písanie vlastných funkcií, je v ňom možné definovať premenné, polia, definovať podmienky a používať cykly.

AWK je určený pre spracovanie stĺpcov orientovaných textových dát, ako sú napríklad tabuľky. V textovom súbore nemôžeme vytvoriť tabuľky v klasickom zmysle, len pomocou oddel'ovačov.

napr.:

```
aaa bbb ccc  
ddd eee fff
```

Tabuľka obsahuje 2 riadky a 3 stĺpce (v predvolenom nastavení oddel'ovač stĺpcov je medzera).

V premennej \$1, \$2, atď. sú stĺpce aktuálneho vstupného riadku. Napr. ak chceme vypísať durhý stĺpec súboru, môžeme použiť nasledujúci awk skript:

```
awk < file '{ print $2 }'
```

V predvolenom nastavení awk rozdeľuje vstupné riadky do polí založených na medzery. To môžete zmeniť pomocou voľby-F na awk a nahradiť medzeru iným znakom.

Aritmetika

Awk je slabo typový jazyk, premenné môžu byť buď reťazce, alebo čísla. Všetky čísla sú floating-point. Napr.: Máme v súbore 3 čísla (a, b, c), a chceme vypísať (a-b)*c, môžeme použiť nasledujúci script:

```
awk <file.txt '{ print ($1-$2)*$3}'
```

Premenné

AWK má zabudované premenné, ako sú napríklad \$0, \$1, \$2, \$NF, atď. (\$NF je posledný stĺpec, \$0 je celý riadok). Okrem týchto, môžeme použiť aj vlastné premenné. (Skript v nasledujúcom príklade vypíše na každom riadku priemernú hodnotu zo všetkých stĺpcov na riadku):

```
awk < a.txt '{ tot=0; for(i=1; i<=NF; i++) tot +=$i; print tot/NF;}'
```

2.2. SED (stream editor)

SED (stream editor) nie je interaktívny textový editor, namiesto toho sa používa na filtrovanie textu (pomocou regulárnych výrazov). SED prečíta celý súbor po riadkoch a vykoná na ňom operácie (alebo sadu operácií), a potom vypíše zmenený text. Typicky sa používa na editovanie častí súboru pomocou vzoru, alebo na nahradenie viacerých výskytov reťazca v súbore. [3]

Ukážka

```
>cat subor.txt  
http://yahoo.com/nieco.html  
>sed -e 's@http://yahoo.com@http://google.com@' subor.txt  
http://google.com/nieco.html
```

Script vymení yahoo.com v textovom súbore na google.com

Môžeme použiť aj zložitejšie regulárne výrazy.
napr.:

```
>cat subor.txt  
Tobe be, or not to be: that is the question:  
Whether 'tis nobler in the mind to suffer The slings and arrows of outrageous fortune, Or to  
take arms against a sea of troubles, And by opposing end them?  
  
>sed -e '/^Tobe.*question/s/Tobe/To/' -e 's/question:/question/' -e '/^Whether/d'
```

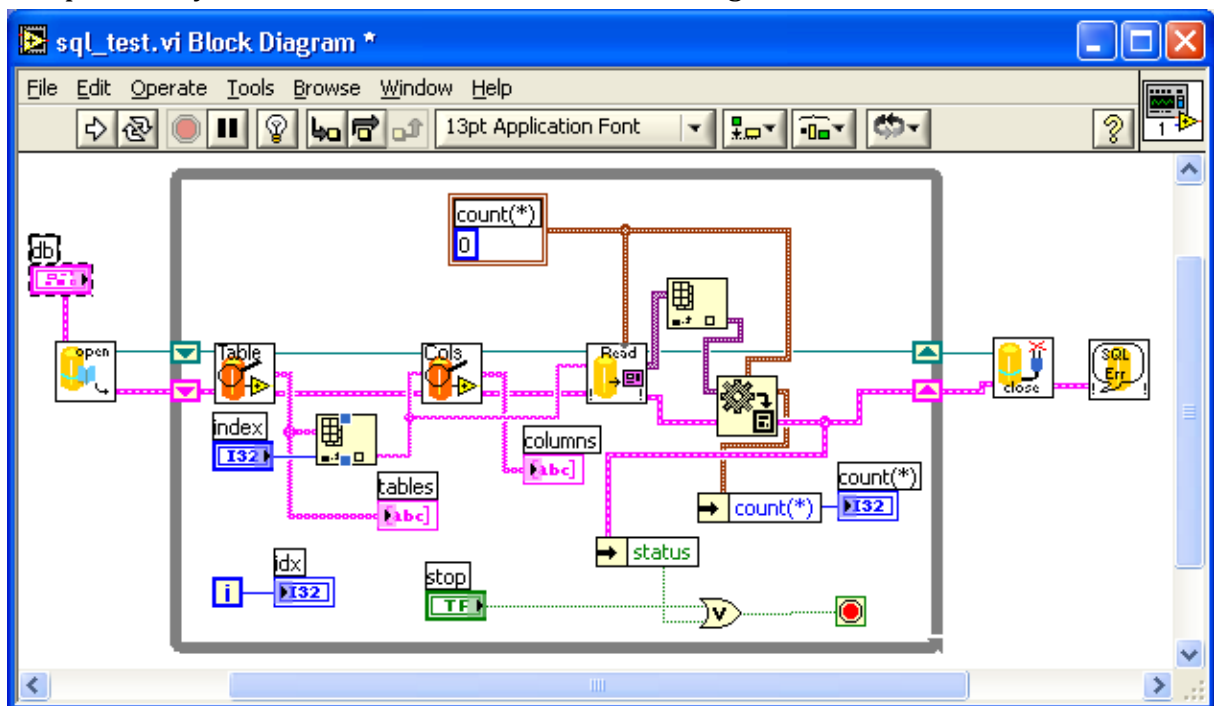
```
subor.txt
To be, or not to be: that is the question
```

Ukážka najpr vyhladáva riadok, ktorý má na začiatku slovo 'Tobe' (znak ^) a obsahuje slovo 'question'. Potom prepíše 'Tobe' na 'To', 'question:' na 'question', a vymaže všetky riadky, ktoré majú na začiatku slovo 'Whether' (posledný riadok).

2.3. LabView

Programy v LabVIEW sa nazývajú virtuálne inštrumenty alebo VI, pretože ich vzhľad a chod imituje fyzické inštrumenty, ako napríklad osciloskopy a multimetre. LabVIEW obsahuje obsiahlu skupinu nástrojov pre akvizíciu, analýzu, zobrazenie a uloženie dát ako aj nástroje, ktoré vám pomôžu riešiť problémy vo vyvíjanom kóde.

LabVIEW vykonáva VI na základe modelu toku dát. Uzol na blokovom diagrame sa vykoná, až keď má prístupné všetky požadované vstupy. Po ukončení vykonania uzol vytvorí výstupné dáta, ktoré sa prenesú do ďalšieho uzla v toku dát. Pohyb dát cez uzly určí poradie vykonávania VI a funkcií v blokovom diagrame. [7]



Obrázok 1. Grafické rozhranie programu LabView

GUI programu LabView obsahuje veľa podobných elementov, ako program Adama Poldaufa (bloky a hrany).

2.4. Adam Poldauf: Vizualny jazyk na konverziu súborov, bakalárska práca, FMFI UK, 2011

Konverzie sú vytvorené graficky na grafickej ploche programu pomocou stavebných prvkov systému. Okrem konverzie textových súborov môžeme vytvoriť aj zoznamy obrázkov, pomocou bloku MetaMiner a DirectoryReader (blok MetaMine má 1 vstup a 4 výstupy- meno súboru, dátum, šírka, výška).

GUI

Grafické rozhranie programu je niekedy trochu komplikované (napr. pri modifikovaní existujúcich blokov na začiatku používateľ ani nevie kam má kliknúť, lebo nastavenia sa objavujú v dolnom rohu programu). Medzi komponentmi grafického rozhrania funguje drag-and-drop.

Usability

Program v praxi som najskôr nevedel použiť. Pri vymazávaní blokov sa graf častokrát rozpadá na komponenty, niekedy všetky bloky zmiznú (aj tie, ktoré používateľ pridal po vymazávaní), pri pridávaní ďalšieho bloku znovu sa objavia, ale potom už je program nepoužiteľný.

Funkcie

Program by mal obsahovať všetky funkcie SED-u programu AWK, ale v skutočnosti nemôžeme nahradiť ich s programom Adama (nemôžeme riešiť matematické úlohy, atď. ako v ukážke č. 2 alebo 3).

2.5. Použité technológie

Systém bude naprogramovaný v programovacom jazyku Java a v prostredí NetBeans. GUI bude realizované v Java FX, pomocou Scene Builder-u. Preto som si zvolil Javu, lebo je cross-platform a poskytuje bohaté API s množstvom užitočných tried.

2.5.1 JAVA

Java je objektovo orientovaný programovací jazyk podobný jazyku C++ (nakolko má oproti nemu viacero obmedzení, niekedy býva označovaný položartom ako C++ --). Java bola navrhnutá tak, aby bola ľahko prenositeľná na rôzne počítačové platformy. Na rozdiel od C++ sa v prípade Javy zdrojový text kompiluje do strojovo nezávislého, veľmi efektívneho bajtového kódu. Ten sa potom interpretuje prostredníctvom modulu JVM (Java Virtual Machine) na ľubovoľnej počítačovej platforme, podporujúcej tzv. Java-runtime, (Windows 95/NT 4.0, Sun OS 4.1, Sun Solaris 2.4, a.i.). Javovské programy tak

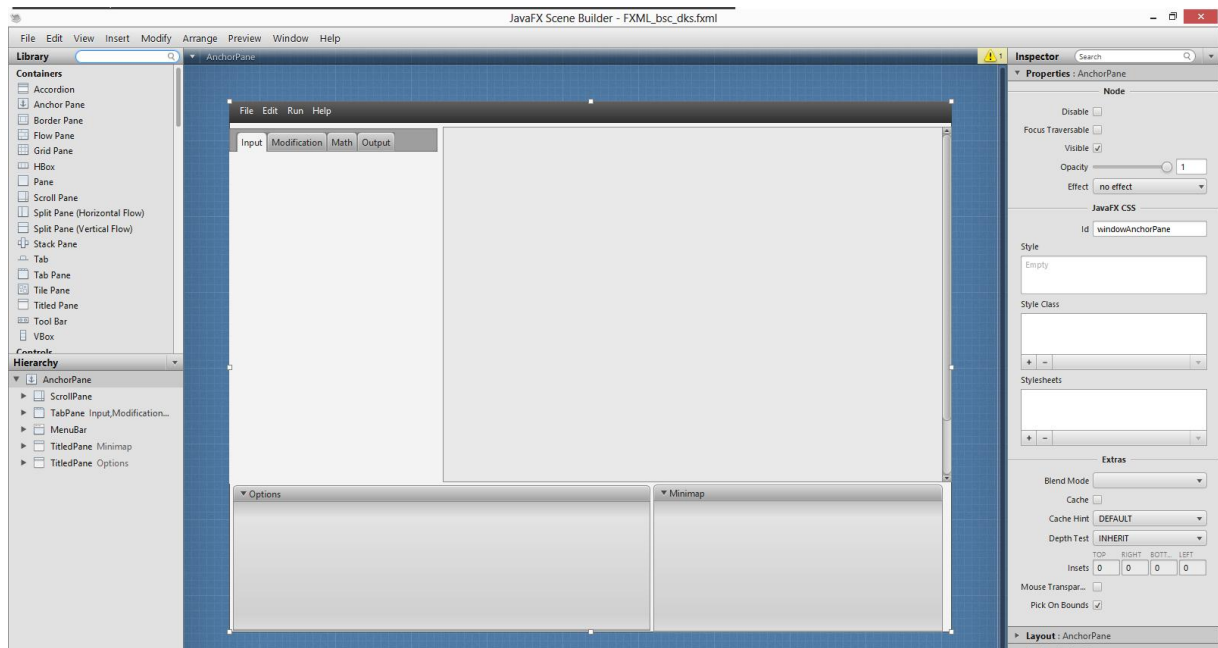
môžu byť prostredníctvom Internetu presúvané z jedného počítačového systému na druhý bez akejkoľvek transformácie a bez akéhokoľvek zásahu používateľa.[5]

2.6. Užívateľské rozhranie

Všetky prvky užívateľského rozhrania pochádzajú zo štandardnej knižnice Java, a z knižnice JAVA FX. GUI je vytvorená pomocou programu Java FX Scene Builder.

2.6.1. JavaFX Scene Builder

JavaFX Scene Builder je jednoduchý grafický editor na vytvorenie GUI pre programy JavaFX. Grafické rozhranie sa uloží do súboru typu fxml. Fxml súbory majú xml štruktúru.



Obrázok 2. Prostredie JavaFX Scene Builder

3. Funkcionalita systému

3.1 Požiadavky na systém

Pri návrhu som sa zameril na dva ciele:

- User friendly grafické rozhranie, aby spracovanie bolo jednoduchšie, ako pomocou sed alebo awk.
- Všetky konverzie by dali uložiť a spúšťať z command-line.

Stručný opis funkcionality systému

- Konverzia súborov (pridávať informácie, spájať viacero súborov, rozdeliť súbor(y) do viacerých súborov)
- Práca s XML (save/load konverzie z xml súboru)

3.2. Grafická konverzia

Všetky konverzie budú vytvorené na grafickom rozhraní pomocou jednotlivých prvkov konverzie. Všetky prvky konverzie (bloky- vrcholy) budú mať určitý počet vstupov, ktoré budú spájané pomocou datovodov (pipes- hrany v grafe).

3.3. Bloky

Ako v prototype aj v tomto projekte budú bloky predstavovať základné stavebné prvky. Všetky bloky budú mať svoje vstupy/výstupy. Počet vstupov/výstupov môžeme meniť počas vytvárania.

1. Input (source): obsahuje informácie o input súbore
2. Manipulation: obsahuje bloky na manipulácie dát
3. Output: bloky, ktoré zapisujú dáta.

3.3.1. Source bloky

LineReader

Číta vstupný súbor po riadkoch. Posiela na výstup string (celý riadok)

Vstupné parametre:

-vstupný súbor (FileName)

-separator (string)

DirectoryReader

Prechádza priečinok a vyhľadáva všetky súbory zadaného typu.

Vstupné parametre:

-zložka (path)

-filetyp

ColumnReader[*]

Číta súbor po riadkoch a na výstup posiela 1 stĺpec (slovo). Môže nahradiť kombináciu blokov LineReader + Splitter.

Vstupné parametre:

-vstupný súbor

-oddeľovač

Konštantný blok

Posiela preddefinovanú hodnotu na výstup.

3.3.2. Manipulačné bloky

Slúžia na manipuláciu so súbormi/reťazcami.

Splitter

Delí reťazec zo vstupu na podreťazce a posiela 1 string na výstup podľa kritérií.

Vstupné parametre:

- oddeľovač
- poradové číslo výstupného stĺpca.

Príklad vstupu: "Meno Priezvisko e-mail Login"

Oddeľovač: ' ' (medzera)

Poradové číslo výstupného stĺpca: 4

Výstup: "login"

Glue

Spája vstupné reťazce do jedného výstupu.

Vstupné parametre:

- spojovník
- input reťazce

Príklad použitia:

1.

Vstupy: 1. "Login" 2. "domain name"

Spojovník: '@'

Výstup: "login@domain name"

2.

Vstupy: 1. "uniba" 2. "sk"

Spojovník: '.'

Výstup: "uniba.sk"

Math[*]

Ak vstupné reťazce splnia kritéria (dajú konvertovať na integer alebo real), blok vykonáva matematickú úlohu, a pošle výsledok do výstupu. Skupina blokov math obsahuje niekoľko preddefinovaných operácií.

Vstupné parametre:

-2x reťazec (stĺpec alebo riadok obsahujúci integer/real)

-operátor (+, -, /, *)

Filter

Hľadá vo vstupnom reťazci podreťazec podľa input parametra. Ak nájde, pošle riadok na výstup, ak nenájde, pošle prázdny reťazec. (Ak riadok obsahuje len prázdne reťazce, LineWriter ich ignoruje, aby výstupný súbor neobsahoval prázdne riadky)

Vstupné parametre:

-reťazec

RegEx[*]

Aplikuje regulárny výraz na vstupný reťazec pomocou knižnice JAVA Regex.

Vstupné parametre:

-regex výraz

3.3.3. Target bloky (output)

Output:

Zapisuje jeho vstup do výstupného súboru.

*Nové funkcie

4. Návrh riešenia

4.1. Členenie návrhu

Návrh sa člení na dve väčšie časti:

- Užívateľské rozhranie
- Dátová časť

Užívateľské rozhranie:

Všetky konverzie sú vytvorené graficky, na grafickej ploche programu pomocou stavebných prvkov (blokov). Bloky predstavujú základnú funkčnú jednotku systému, počas konverzie spracovávajú ich vstup a posielajú výsledok do výstupu. Každý blok má určitý počet vstupov a 1 výstup (1 výstup ale môže poslať jeho výstup do niekoľko vstupov).

Dátová časť:

-vykonávanie konverzie

-ukladanie/načítavanie konverzie

-spúšťanie uloženej konverzie v batch mode pomocou príkazového riadku

4.2. Triedy

4.2.1. Block

Hlavnou úlohou triedy je vykreslenie blokov na grafickú plochu.

Konštruktor- má 3 vstupné parametre, 2x integer (x a y súradnice) a 1x string (typ bloku).

newPos – pomocná funkcia drag-n-drop-u, slúži na posunutie blokov (2 vstupné parametre, X a Y súradnice).

add_ng- slúži na pridávanie susedov (1 vstupný parameter typu Block).

create_pipe- funkcia na vytváranie datovodov medzi 2 blokmi konverzie.

draw- 1 vstupný parameter typu AnchorPane, úlohou je vykresliť block na grafickú plochu (resp. na AnchorPane).

list_options- 1 vstupný parameter (AnchorPane), slúži na výpis možností jednotlivých blokov.

getType- vráti typ aktuálneho bloku (String).

clicked- vráti true, ak používateľ klikol na blok, inak false.

clicked_on_in/output- funguje presne, ako funkcia clicked, rozdiel je v tom, že vráti číslo len vtedy, ak používateľ klikol na vstup/výstup bloku, môže vrátiť 1 alebo 2. (1 vtedy, ak používateľ klikol na 1 input, 2, ak na 2. input).

4.2.2. Pipe

Úlohou je spájať 2 bloky, okrem konštruktora obsahuje 2 funkcie, draw a new_pos.

Konštruktor- 6 vstupných parametrov, 4x súradnice (x1,y1 -začiatkové, x2,y2- a koncové body pipe-u), AnchorPane, na vykresľovanie a parameter, kam bude smerovať pipe. (1. alebo 2. input)

draw- úlohou je vykresliť hranu.

new_pos-4 vstupné parametre, (x1,x2,y1,y2) prepisuje začiatkové a koncové súradnice.

4.2.3. SL_conv

SL- skratka SaveLoad. Konverzie som rozhodol ukladať vo formáte XML, lebo je ľahko čitateľný pre človeka, editovateľný pomocou textového editora, a Java má dobre vypracované knižnice na spracovanie XML súborov.

save- 2 vstupné parametre, ArrayList<Block> (arraylist obsahuje všetky bloky) a filename (meno súboru). Pomocou DOM-u uloží všetky bloky do XML súboru.

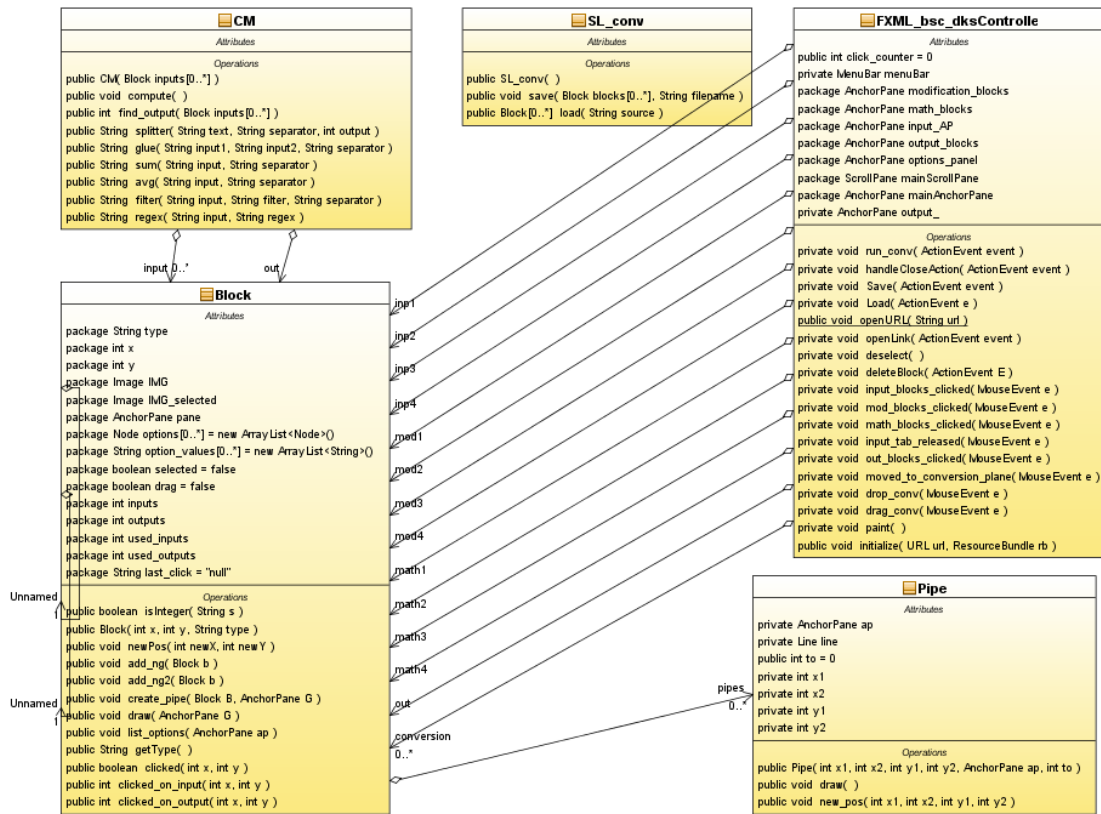
load- 1 vstupný parameter, meno súboru (filename), pomocou DOM-u prečíta celý súbor, a vráti ArrayList, čo bude obsahovať všetky bloky.

4.2.4. CM

Výpočtový model (Model of Computation)- hlavná časť projektu, slúži na vykonávanie konverzií. Všetky bloky majú príslušnú funkciu (regex, filter, math, atď.).

Konštruktor- 1 vstupný parameter (arraylist), pomocou funkcie *compute* vykonáva konverziu.

compute-hlavná časť, slúži na vykonávanie konverzie.



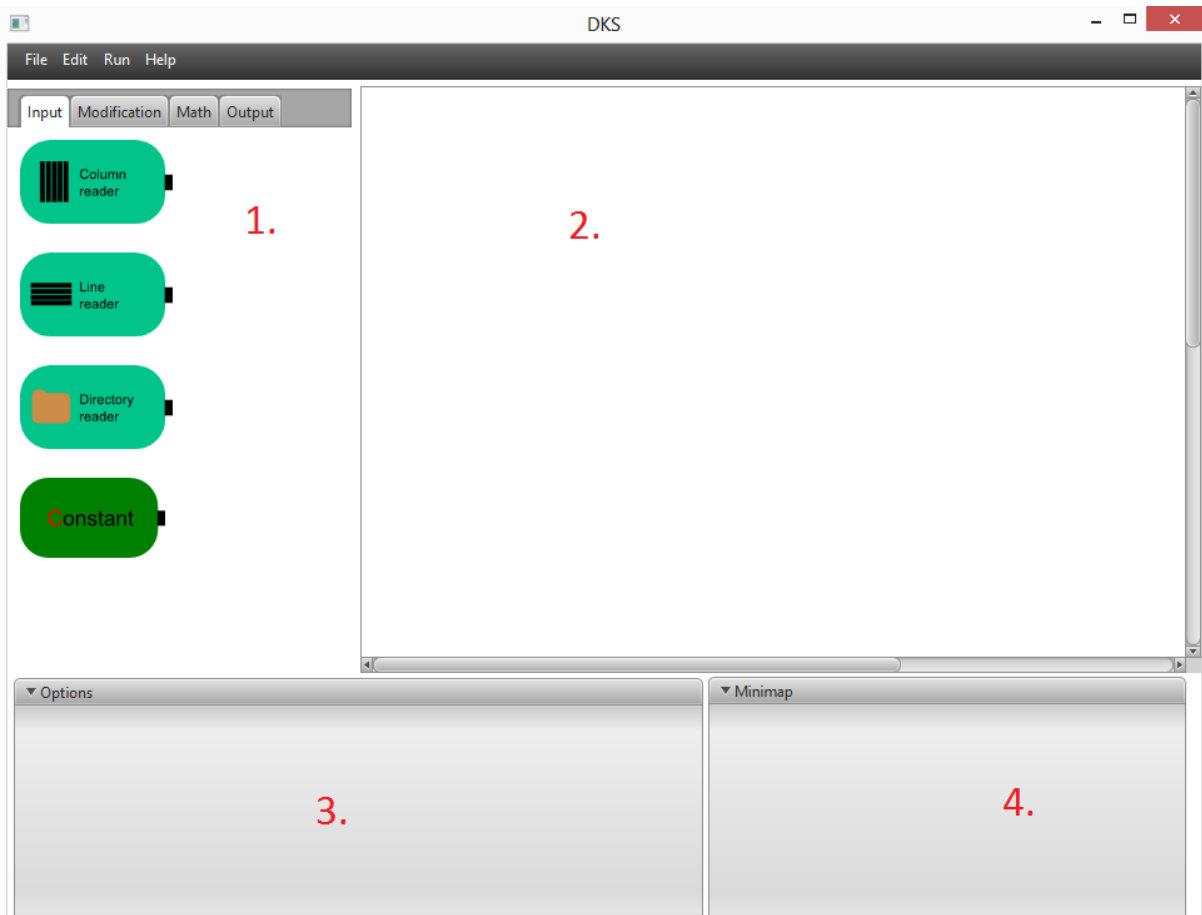
Obrázok 3. Class Diagram

5. Implementácia

5.1. GUI

Grafické rozhranie projektu sa člení na 4 veľké časti:

1. TabPane blokov
2. mainAnchorPane
3. options_panel
4. minimap



Obrázok 4. Prostredie DKS

5.1.1. TabPane blokov

Obsahuje ponuku všetkých blokov, bloky sú rozdelené podľa funkcionality do 4 tabov:

- Input
- Modification
- Math
- Output

5.1.2. mainAnchorPane

Grafická plocha na vytváranie konverzií.

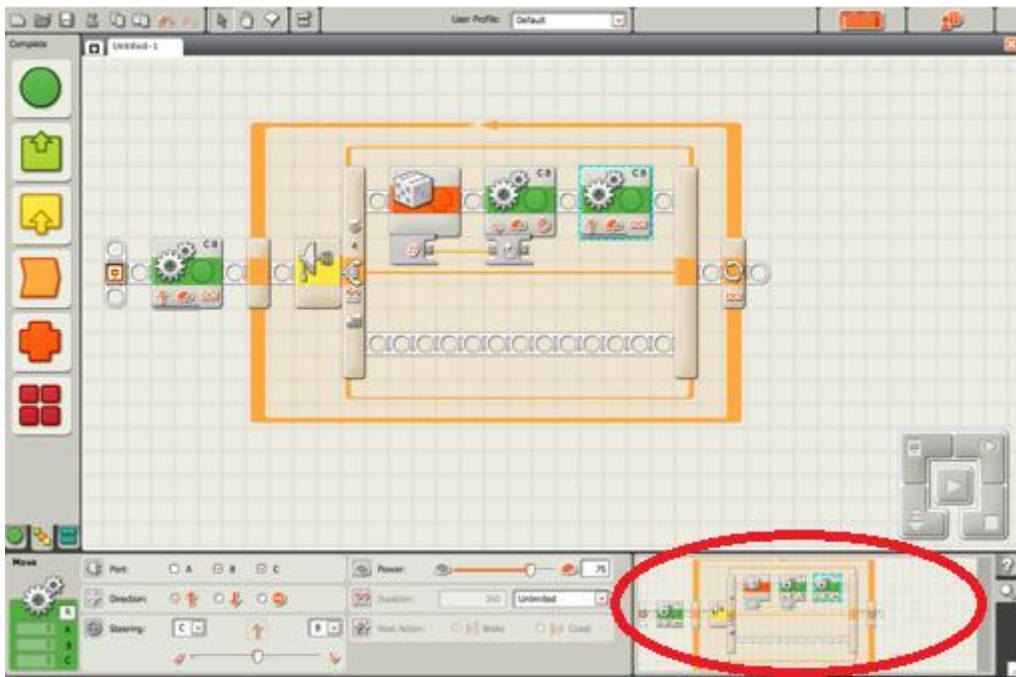
Drag and drop- bloky nemajú zafixovanú pozíciu, používateľ ich môže posunúť pomocou drag and drop funkcie. (drag_conv funkcia controller-u)

5.1.3. options_panel

Všetky bloky majú uloženú sadu možností, po kliknutí sa možnosti zobrazia na AnchorPane-e Options.

5.1.4 minimap

Konverzie môžu byť oveľa väčšie, ako grafická plocha programu, a preto nemôžu zobraziť naraz všetky bloky. Idea minimap-u je taká, že zobrazí celú plochu škalovalne, ako v programe Lego Minndstorms NXT, a používateľ môže vybrať, že aktuálne ktorú časť chce vidieť.



Obrázok 5. Prostredie Lego Mindstorms NXT

5.2. Vykonávanie konverzie z príkazového riadku

Uloženú konverziu bude možné spúšťať aj z príkazového riadku.

Pri zavolaní interpretera uloženej konverzie sú dve možnosti:

1. \$DKS konverzia.xml
2. \$DKS konverzia.xml input.txt

1. Ak konverzia.xml obsahuje cestu k textovému súboru, vtedy v batch mode konverzia jednoducho vykonáva.

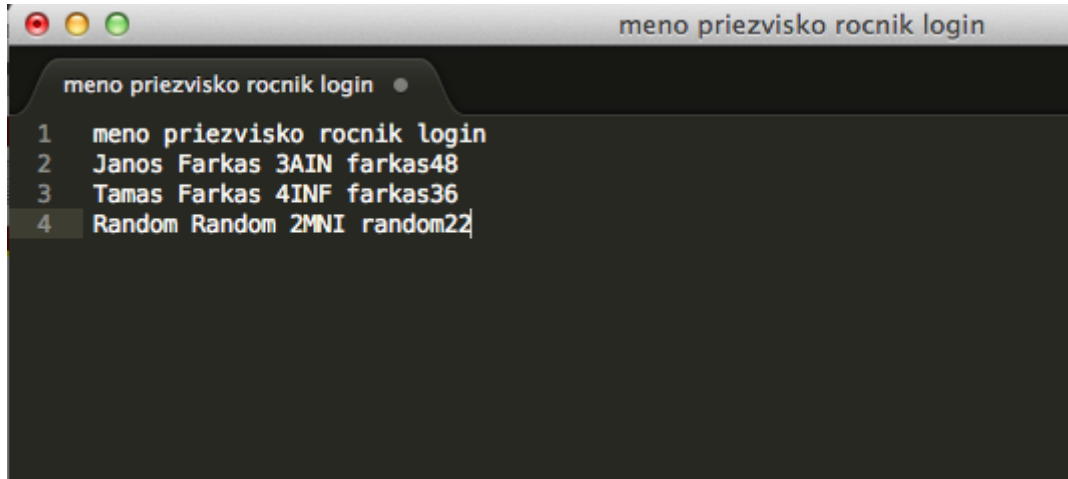
2. konverzia.xml neobsahuje žiadnu cestu, namiesto toho obsahuje parameter \$1, \$2, atď. vtedy cesta k súboru bude argument z príkazového riadku.(\$1- 1. argument, 1. LineReader, \$2- 2. argument, 2. LineReader, atď.)

5.3. Príklad použitia

V tejto kapitole sú uvedené vzorové príklady konverzií, ktoré je možné vytvoriť pomocou systému.

5.3.1. Jednoduchý filter

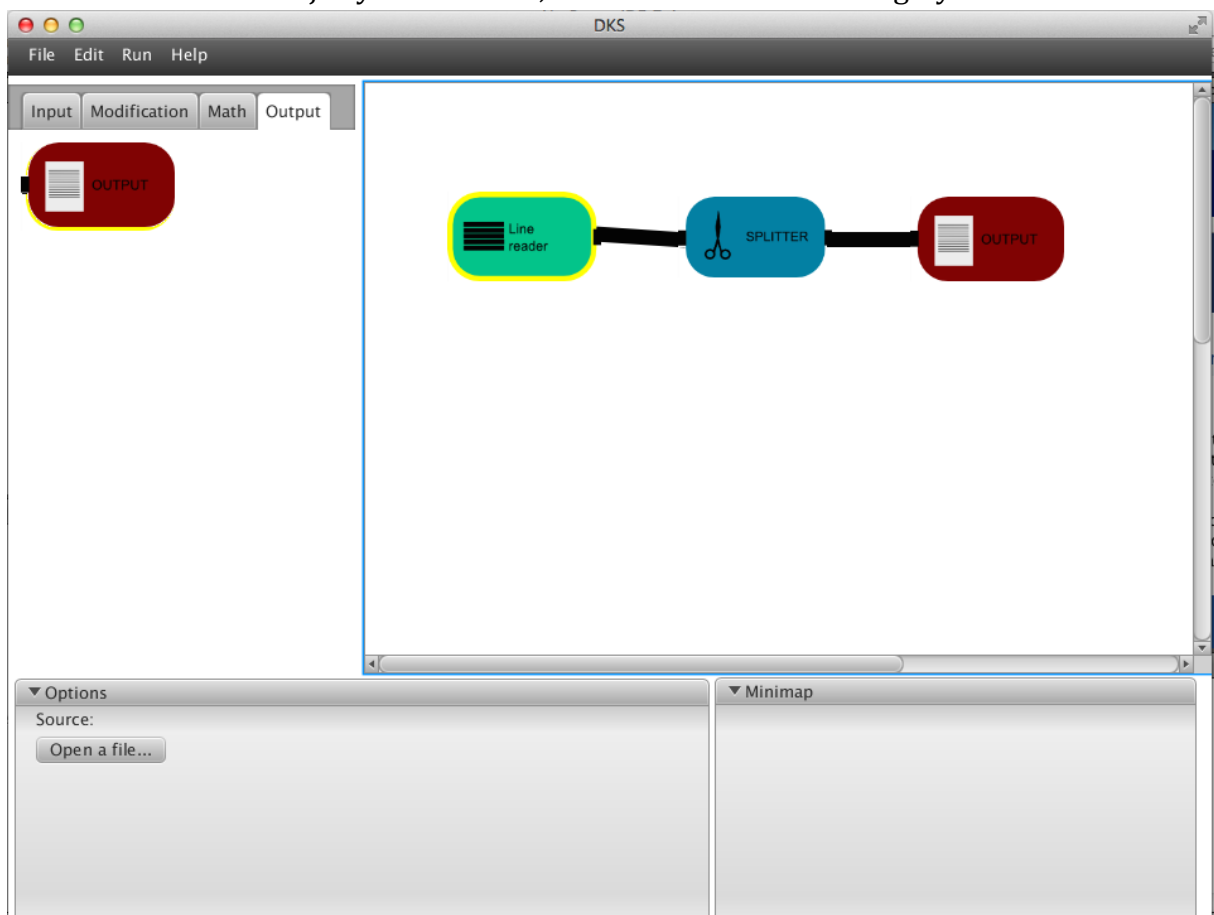
Vstupný súbor obsahuje údaje študentov.



```
meno priezvisko rocnik login
1  meno priezvisko rocnik login
2  Janos Farkas 3AIN farkas48
3  Tamas Farkas 4INF farkas36
4  Random Random 2MNI random22
```

Obrázok 6. Input 1.

Našou úlohou je vytvoriť súbor, čo bude obsahovať len loginy študentov.

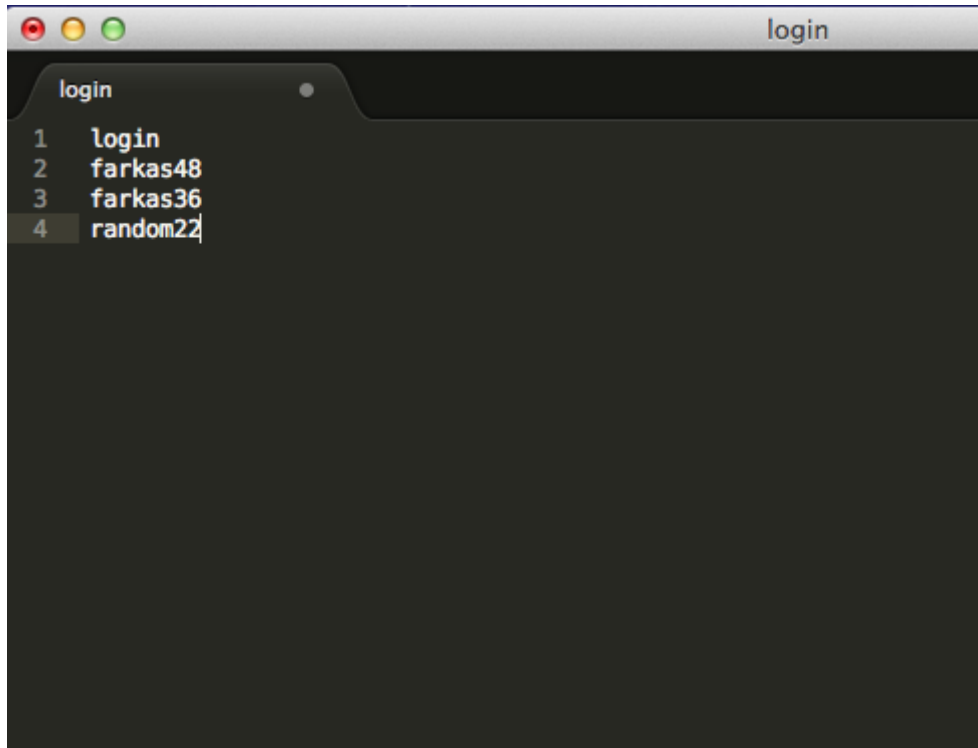


Obrázok 7. 1. Príklad- bloky

Postup riešenia:

1. vytvoriť grafický predpis
2. nastaviť všetky možnosti (line reader- vstupný súbor, splitter- separator a poradové číslo stĺpca, output- výstup)
3. Run-> Run

Výsledok:



```
login
1 login
2 farkas48
3 farkas36
4 random22
```

Obrázok 8. Výsledok konverzie 1.

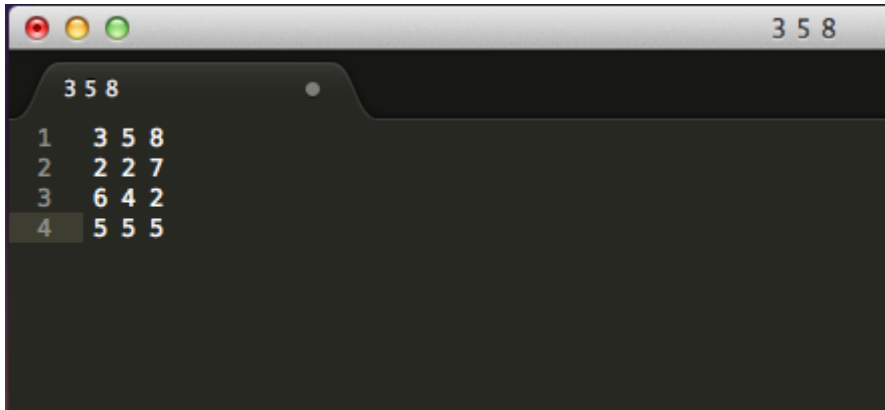
5.3.2. Priemerná hodnota riadku

Riešenie pomocou AWK:

```
awk < a.txt '{ total=0; for(i=1; i<=NF; i++) total += $i; print total/NF;}'
```

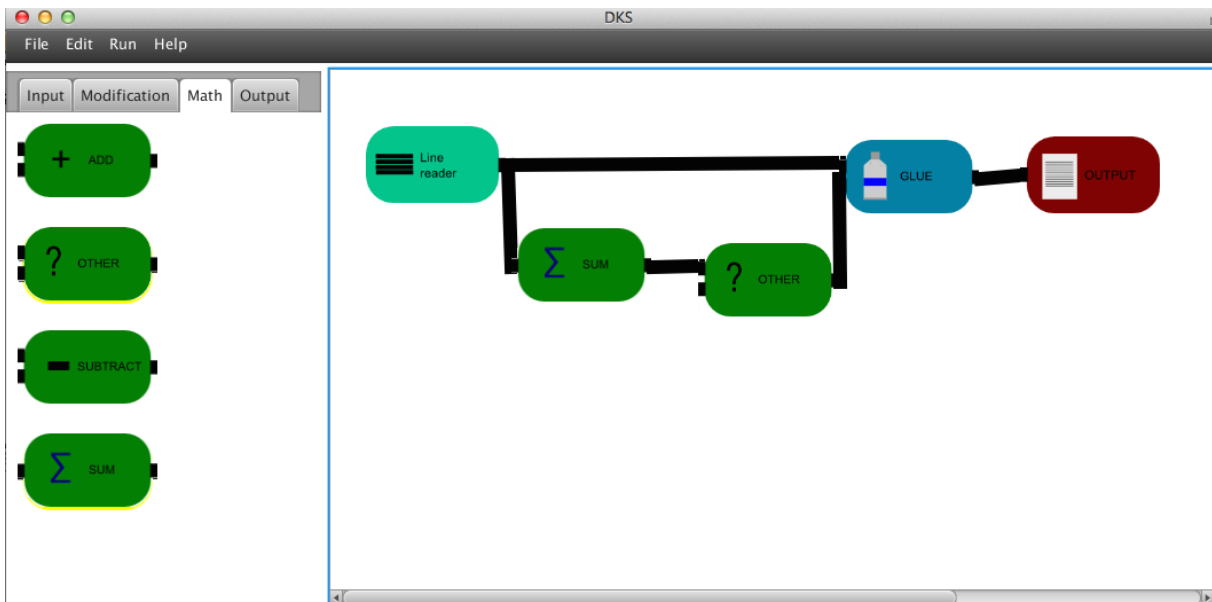
Skript vypíše na každom riadku priemernú hodnotu.

Vstupný súbor:



Obrázok 9. Input 2

Grafická konverzia:

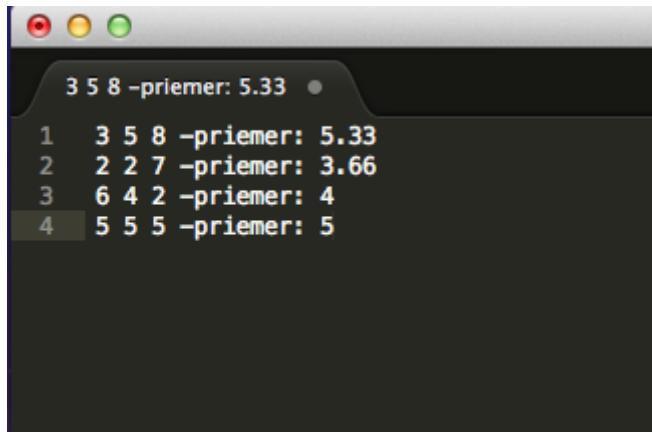


Obrázok 10. 2. Konverzia- bloky

Postup riešenia:

1. linereader posiela celý riadok na 2 rôzne miesta.
2. SUM- vráti sumu všetkých čísel riadku.
3. Math-Other, nastavíme jeden input (sumu) 2. číslo bude konštanta (3, počet stĺpcov), operator "/".
4. Glue dostane ako 1. input celý riadok, a 2. input číslo, nastavíme connector na "-priemer: "."
5. Output
6. Run-Run

Výsledok:



```
3 5 8 -priemer: 5.33
1 3 5 8 -priemer: 5.33
2 2 2 7 -priemer: 3.66
3 6 4 2 -priemer: 4
4 5 5 5 -priemer: 5
```

Obrázok 11. Výsledok 2

5.3.3. Unix -Dos konverzia

Riadky v Unix-ových súborov končia znakom '\n'. V DOS-e je to inak, koniec riadku je označené reťazcom '\r\n'. Pomocou programu DKS môžeme konvertovať Unixove textové súbory na DOS-ové a naopak.

DOS-UNIX:

Blok LineReader číta súbor po riadkoch. (Preto ignoruje znak '\n' na konci riadku). Na konverziu potrebujeme LineReader, Splitter a LineWriter. Splitter dostane separator '\r' a číslo stĺpca 1, a posiela na výstup.

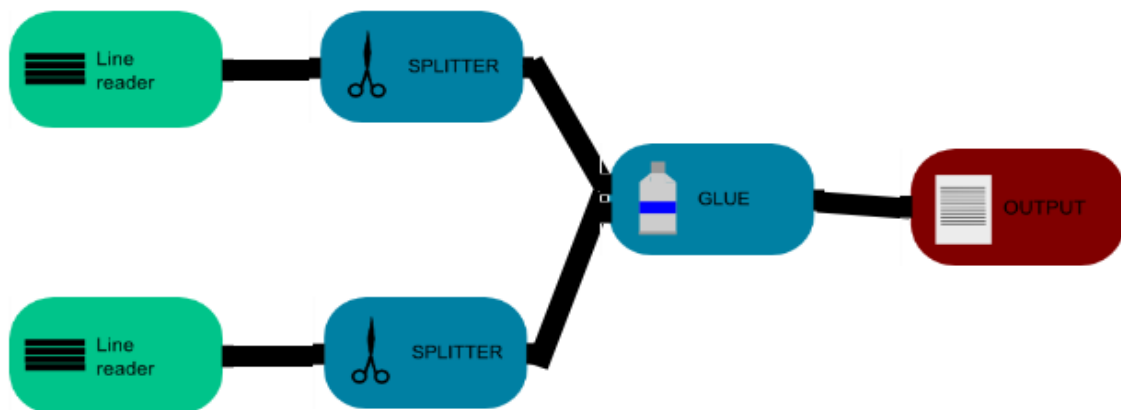
UNIX-DOS:

Na konverziu potrebujeme tiež 3 bloky, ale v tomto prípade namiesto bloku Splitter budeme používať Glue. Glue pripájame na LineReader a 2. input necháme prázdne. Ako spojovník nastavíme znak 'r' a posielame na výstup.

5.3.4 Práca s viacerými súbormi

Príklad: textový súbor obsahuje zoznam študentov, údaje o každom študentovi. Zo súboru potrebujeme vytiahnuť iba login a zlúčiť s údajom o známke študenta, ktorý sa nachádza v inom súbore.

Grafická konverzia:



Obrázok 11. 3. Konverzia- bloky

Postup riešenia:

1. 1. linereader posiela riadok do splitteru.
2. 2. linereader posiela riadok do 2. splitteru.
3. 1. splitter posiela na výstup login, 2. splitter posiela známky študenta.
4. Glue spája 2 vstupy
5. Output

6. Záver

Cieľom mojej práce bolo navrhnuť a implementovať nástroj na vytváranie grafov (grafických predpisov) na konverzie súborov.

Vývoj systému prebehol podľa očakávaní s drobnými problémami, no všetky problémy boli odstránené vyhľadáním pomoci na internete alebo vďaka konzultácii so školiteľom.

V budúcnosti by som chcel rozšíriť aplikáciu, preto som rozhodol zverejniť zdrojový kód na GitHub-e, a urobil som z toho open-source projekt. (https://github.com/gyeprefos/BC_praca)

7. Použitá literatúra a zdroje

GNU.org Documentation (online) (použití: 15.4.2014)

<http://www.gnu.org/software/sed/>

Advanced Bash-Scripting Guide: AWK help (použití: 4.2.2014)

(*syntax*)<http://tldp.org/LDP/abs/html/awk.html>

AWK (použití: 3.2.2014)

http://ihorls.ru/od/glossary/l/bldef_cmd_awk.htm?-Defin%C3%ADcia---awk-Command

GNU.org SED manual page (použití: 2.2.2014)

<http://www.gnu.org/software/sed/manual/sed.html>

Julius CSOŇTO- JAVA –online verzia (použití: 15.4.2014)

<http://hron.fei.tuke.sk/~csonto/kniha/java.htm>

Ing. Mgr. Márk Jónás (ANV s.r.o., Bratislava), Ing. Zuzana PetrÁková (ANV s.r.o., Bratislava),

Mgr. Silvia Mókosová (ANV s.r.o., Bratislava), Ing. Gregor Izrael, PhD. (ANV s.r.o., Bratislava)

Vývoj aplikÁcií v LabVIEW pre meracie úlohy s CompactRIO systémmi (13.-15. 8. 2012 a 20.-24. 8. 2012) (použití: 14.4.2014)

http://www.crr.vutbr.cz/system/files/brozura_08_1208.pdf

LabView example (použití: 14.4.2014)

<http://sql-lv.sourceforge.net/>

Príloha č. 1

CD s kompletnými zdrojovými kódmi aplikácie a aplikáciou samotnou. Súčasťou CD je aj súbor „Readme.txt“ popisujúci obsah CD.