

EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103005/B/2014/3711682507

SOFTVÉROVÉ SPRACOVANIE ŠTATISTICKÝCH ÚLOH

Bakalárska práca

2014

Tomáš Lopusník

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

SOFTVÉROVÉ SPRACOVANIE ŠTATISTICKÝCH ÚLOH

Bakalárska práca

Študijný program:	Hospodárska informatika (Jednoodborové štúdium, bakalársky I. st., externá forma)
Študijný odbor:	9.2.10 Hospodárska informatika
Školiace pracovisko:	KŠ FHI – Katedra štatistiky FHI
Vedúci záverečnej práce:	doc. RNDr. Viera Labudová, PhD.

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

Dátum:

.....

podpis študenta

Rád by som sa poďakoval doc. RNDr. Viere Labudovej, PhD. za odbornú pomoc a podnetné návrhy pri tvorbe programu aj dokumentácie, ktoré mi poskytla.

ABSTRAKT

LOPUŠNÍK, Tomáš, *Softvérové spracovanie štatistických úloh* - Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra štatistiky. - doc. RNDr. Viera Labudová, PhD. - Bratislava: FHI EU, 2014, 40s.

Cieľom záverečnej práce je vytvoriť vlastný program na vykonanie výpočtov popisných charakteristík poskytnutého štatistického súboru. Pri tvorbe programu sme využili naše skúsenosti nadobudnuté vývojom Flash aplikácií a interaktívnych animácií. Počas vývoja sme nadobudli nové skúsenosti pri práci s tabuľkovými dátami a ich zobrazením do grafov. Práca je rozdelená do štyroch kapitol. Obsahuje tri UML schémy a menšie množstvo vysvetľujúcich obrázkov. Prvá kapitola je venovaná súčasnému stavu problematiky softvérových riešení na spracovanie štatistických údajov a predstavuje niektoré existujúce riešenia a ich základné vlastnosti. V druhej kapitole predstavujeme cieľ práce a čo všetko od nášho programu očakávame. Tretia kapitola je venovaná samotnému postupu pri tvorbe programu a použitia dostupných prostriedkov. Záverečná, štvrtá kapitola predstavuje finálny produkt, ktorým je inštalačný balík a jeho použitie. Výsledkom riešenia danej problematiky je vlastný, nami vytvorený program a zdrojový kód, ktorý môžeme v budúcnosti ďalej rozširovať a zdokonaľovať.

Kľúčové slová:

Flash, opisné charakteristiky, štatistický program R

ABSTRACT

LOPUŠNÍK, Tomáš, *Data processing with open source program* - University of Economics in Bratislava. Faculty of Economic Informatics; Department of statistics. - doc. RNDr. Viera Labudová, PhD. - Bratislava: FHI EU, 2014, 40p.

The goal of the thesis is to create a custom program to carry out calculations of descriptive characteristics of provided statistical data. During the creation of the program, we've used our previous experience working with Flash applications and interactive animations. During the development we've acquired new experience working with tabular data and their plots. The work is divided into four chapters. It contains three UML schemes and several helpful images. The first chapter is devoted to current state of available software solutions for processing statistical data and presents some of the existing software and their basic properties. Second chapter introduces our thesis' goal, which is a working program and its features. The third chapter describes programming procedures using available resources. The final fourth chapter introduces the final product and describes its operation. The result of our work is our own program and source code which can be expanded and advanced in the future.

Key words:

Flash descriptive characteristics, statistical program R

OBSAH

Úvod	7
1 Súčasný stav riešenej problematiky doma a v zahraničí	8
2 Cieľ práce	10
3 Metodika práce a metódy skúmania	12
3.1 Platforma	12
3.2 Prostriedky	14
3.2.1 Flex	14
3.2.2 ActionScript + XML	15
3.2.3 Štruktúra MVC	16
3.2.4 Data binding	18
3.2.5 ASDoc	20
3.3 Postupy	21
3.3.1 Údaje	22
3.3.2 Spracovanie	23
3.3.3 Výpočty popisných charakteristík	25
4 Výsledky práce	31
4.1 Inštalácia	31
4.2 Používanie	32
4.2.1 Tabuľkové údaje	33
4.2.2 Výsledkové údaje	35
4.2.3 Ostatné údaje	35
Záver	37
Zoznam použitej literatúry	40

Úvod

Predmet štatistiky nachádza využitie v mnohých aspektoch nielen ekonomiky, ale aj v ďalších odvetviach. Ako matematická vedná disciplína slúži na zber, analýzu, interpretáciu a prezentáciu dát. Štatistické metódy popisnej štatistiky sumarizujú a popisujú zozbierané dáta, čo je veľmi užitočné pri prezentovaní výsledkov rôznych experimentov a prieskumu.

Popisná štatistika často pracuje s veľkým množstvom vstupných údajov a na dosiahnutie jej výsledkov je potrebné množstvo výpočtov. Tieto výpočty sú veľmi náročné na manuálne rátanie človekom, kde navyše môže dôjsť k omylom, keďže výpočty sú náročné a z časového hľadiska dlho trvajúce. Moderné informačné systémy dokážu vykonávať obrovské množstvá matematických a teda aj štatistických výpočtov nepomerne rýchlejšie a presnejšie ako človek. So správnym softvérovým vybavením je tak možné vytvoriť program na spracovanie komplexných štatistických výpočtov.

Pre bakalársky projekt "*Softvérové spracovanie štatistických úloh*" sme sa rozhodli vytvoriť úplne nový program na spracovanie zdrojových štatistických údajov a to z niekoľkých dôvodov. Jedným z prvých je mať vlastný program, ktorého zdrojové súbory poznáme a môžeme ďalej podľa potreby rozširovať a použitie ktorého nie je obmedzené rôznymi licenciami. Ďalším dôvodom je dynamickosť. Zmenou zdrojových údajov dôjde okamžite ku zmene popisných charakteristík v programe na rozdiel napríklad od programu Excel, kde sú popisné charakteristiky napevno zapísané v tabuľke. V neposlednom rade je dôvodom grafický a estetický vzhľad, ktorý je možné vo vlastnom programe upraviť podľa potreby. Vytvorený vlastný program je možné tiež využiť aj ako výukový program, kde si užívateľ môže pozrieť ako zmena zdrojových štatistických údajov vplýva na celkový výsledok popisných charakteristík daného štatistického súboru.

1 Súčasný stav riešenej problematiky doma a v zahraničí

V súčasnosti existuje niekoľko viac či menej sofistikovaných softvérových riešení na spracovanie štatistických výpočtov. Tieto softvérové riešenia vedia používateľovi poskytnúť informácie o jeho štatistickom súbore od úplne základných popisných charakteristík až po komplexné spracovanie štatistických údajov s tabuľkami, grafmi, dokonca aj predikciou vývoja do budúcnosti, v prípade časových radov.

Medzi najzákladnejšie a najjednoduchšie na použitie patrí doplnok "Analytické nástroje" programu Microsoft Excel. Tento vie po spustení poskytnúť o štatistickom súbore niektoré jeho základné údaje ako modus, medián, smerodajnú odchýlku, či šikmosť alebo špicatosť. Jeho nevýhodou je, že zobrazené údaje sú často krát v inom jazyku ako jazyk programu Excel, čo je spôsobené inou jazykovou verziou spomínaného doplnku a to kazí celkový dojem a ide skutočne len o základné údaje. Ďalšou nevýhodou je, že výsledky štatistickej analýzy sú fixne zapísané v tabuľkách bez dynamického prepojenia, čo znamená, že zmenou zdrojových štatistických údajov nedôjde automaticky ku zmene popisných charakteristík. Pre užívateľa môže byť máťúce, že zmenou údajov v zdrojovej tabuľke nevyvolá zmenu popisných charakteristík.



Medzi sofistikovanejšie programy na štatistickú analýzu patrí R. R je jazyk a prostredie pre štatistické výpočty a grafiku. Je to GNU Projekt podobný jazyku a prostrediu S. R sa dá považovať za inú implementáciu S, pričom je medzi nimi značný rozdiel, ale veľa kódu napísaného pre S bude bez zmeny fungovať aj pre R. R poskytuje rozličné grafické a štatistické techniky a je vysoko prispôsobivý. S je často používaný jazyk pre štatistické výpočty a R k nemu ponúka open-source alternatívu. [1] Napriek tomu, že poskytuje vysokú flexibilitu a mocné nástroje na uskutočňovanie štatistických výpočtov, pre bežného používateľa sa môže zdať práca s ním až príliš komplikovaná.



Jedným z ďalších programov využívajúcich štatistické metódy, no viac zameraný na ekonometrickú analýzu ako na popisnú charakteristiku, je program gretl (Gnu Regression, Econometrics and Time-series Library). Je



to open-source program napísaný v jazyku C. [2] Je menej komplexnejší ako R a nie tak náročný na použitie, no je orientovaný na ekonomické výpočty.

Súčasnú softvérovú riešenia na spracovanie štatistických údajov sú teda buď príliš jednoduché a neposkytujúce pokročilejšie nástroje, alebo príliš komplikované na použitie, čo má za následok odradenie používateľa od práce s nimi. Takisto žiaden z nich nie je určený na vzdelávanie v oblasti štatistiky, ale priamo na prácu.

2 Cieľ práce

Ako cieľ práce sme si stanovili vytvoriť vlastný program na spracovanie štatistických údajov. Program by mohol slúžiť aj ako prostriedok na vzdelávanie, kde si používateľ môže sám vyskúšať ako zmena jednotlivých vstupných údajov ovplyvní celkový výsledok, keďže štatistické výpočty pre získanie popisných charakteristík prebiehajú hneď ako príde ku zmene zdrojových údajov. Program navyše môže používateľovi vysvetliť ako k jednotlivým výsledkom výpočtov došiel, napríklad zobrazením použitých vzorcov s dosadenými konkrétnymi hodnotami.

E-Learning predstavuje v súčasnosti najmodernejší spôsob výučby s využitím informačných technológií. Výučba sa zväčša praktikuje dištančnou formou, kde sú študentovi k dispozícii výukové kurzy prostredníctvom internetových portálov. Takáto forma výučby, kde je tútor fyzicky oddelený od študentov a slúži len na koordináciu výučby už existuje veľa rokov a rozvoj moderných informačných technológií zasiahol aj do vzdelávania a výrazne vylepšil vyučovací proces.

Vzhľadom na to, že sme sa rozhodli vytvoriť program, ktorý bude možné nainštalovať ako balík pre Adobe AIR a jeho primárna činnosť je vykonávať štatistické výpočty, sme sa rozhodli ho pomenovať *AirStat*. Ide v podstate o pracovný názov, ktorý môžeme pred samotným vypustením programu do ostrej prevádzky zmeniť.

Ešte pred samotným začiatkom tvorby programu AirStat sme si uvedomili, že nie je možné vytvoriť plnohodnotný program na štatistické výpočty za obdobie jedného školského roka, teda ako praktickú časť bakalárskej práce. Súčasná softvérová riešenia, ktoré trh ponúka, sú vo vývoji už dlhé roky a neustále sa zdokonaľujú. Tvorbou vlastného programu sme však získali neoceniteľné skúsenosti pri softvérovom spracovaní a uchovávaní údajov a vytvorili sme aj základný program, ktorý je možné v budúcnosti ďalej rozvíjať a zdokonaľovať.

Od nášho programu AirStat sme si ako úvodný cieľ stanovili aby zvládol vypočítať základné popisné charakteristiky poskytnutého súboru štatistických údajov, ktoré predstavujú v tejto časti vývoja programu len prierezové a nie časové údaje. Na tom si vieme overiť, že AirStat je schopný vykonávať štatistické výpočty a je pripravený na rozšírenie o pokročilejšie štatistické výpočty a analýzy pri jeho ďalšom vývoji. Medzi

základné popisné charakteristiky, ktoré sme chceli aby AirStat vedel vypočítať patria priemer, modus, medián, dolný a horný kvartil, šikmosť a špicatosť. Takisto sme do výpočtov zahrnuli miery variability absolútne aj relatívne ako napr. štandardná odchýlka či variačný koeficient. [3] Pre vizualizáciu uchovaných a vypočítaných údajov sme v programe chceli zobrazit' stĺpcový graf a box plot.

Na rozdiel od súčasne dostupných programov na štatistické výpočty sme chceli aby bolo možné s naším programom zasahovať do zdrojových údajov a okamžite tak vidieť prejavené zmeny v popisných charakteristikách, stĺpcovom grafe, či box plote.

AirStat sme od začiatku tvorili s myšlienkou zachovať štruktúru MVC, aby bolo možné jednotlivé komponenty na výpočet štatistických údajov jednoducho prispôbiť, rozšíriť o ďalšiu funkcionality, alebo premiestniť do úplne iného projektu. Program teda funguje samostatne na jednoduchú prezentáciu dosiahnutých výsledkov našej práce, ale jeho knižnice, ktoré obsahujú metódy na spracovanie vstupných údajov a ďalších štatistických výpočtov, je možné jednoducho použiť aj v inom projekte.

AirStat sme sa rozhodli primárne publikovať ako desktopovú aplikáciu, nakoľko používateľ je zvyknutý spracovávať štatistické údaje v desktopových aplikáciách. Program je možné s použitím toho istého zdrojového kódu publikovať aj pre webové prehliadače, ale niektorým používateľom by mohlo vadit' pri používaní programu posielat' citlivé štatistické údaje na internet. Ďalšou nevýhodou programu pre webový prehliadač je jeho nedostupnosť v prípade problémov so stránkou, kde je umiestnený, alebo ak používateľ nie je pripojený na internet.

3 Metodika práce a metody skúmania

Pri vývoji nášho programu sme si na začiatok museli odpovedať hneď na niekoľko otázok počnúc zvolením vhodnej platformy a programovacieho jazyka cez princíp fungovania akým bude program prijímať a spracúvať údaje až po konkrétne metódy a postupy použité v programe.

3.1 Platforma

Pre vývoj programu AirStat sme sa rozhodli použiť platformu *Adobe Flash* s použitím programovacieho jazyka *ActionScript 3.0* a s využitím štruktúry *Flex 4.12.0*. Jazyk *ActionScript* je založený na skriptovacom jazyku *ECMAScript*, na ktorom je založený aj programovací jazyk *JavaScript* využívaný na moderných webstránkach. Preto sú na pohľad tieto dva jazyky veľmi podobné. Rozhodnutie použiť práve túto platformu a programovací jazyk ovplyvnili hlavne naše dlhoročné skúsenosti s prácou s týmto programovacím jazykom. Vedeli sme čoho všetkého je táto platforma schopná a aké sú jej obmedzenia.

Adobe Flash je multimediálna platforma pracujúca s vektorovými grafikami na tvorbu animácií, hier, alebo pokročilých internetových aplikácií. Vektorová grafika využíva na kreslenie jednoduché tvary ako body, priamky, krivky a polygóny, pričom všetky tieto tvary sú vyjadriteľné matematickým výrazom. Toto má za následok, že súbory s vektorovými obrázkami, resp. animáciami zaberajú menej miesta na pevnom disku počítača ako štandardné bitmapové obrázky a vektorové obrázky je možné približovať bez straty ich kvality, pričom pri bitmapových obrázkoch dochádza pri približovaní k ich skresleniu, takzvanej pixelácii. Ich nevýhodou je, že nimi nie je možné zakresliť detailné tvary zachytené napríklad vo fotografii.



Pôvodná aplikácia ešte pod názvom *SmartSketch* bola vyvinutá Jonathanom Grayom a publikovaná spoločnosťou *FutureWave Software* ako kresliaca aplikácia pre perové počítače používajúce operačný systém *PenPointOS*. *PenPoint* sa však na trhu nepresadil a aplikácia *SmartSketch* bola pretvorená pre prácu pod operačnými systémami *Windows*



a Mac OS. S narastajúcou popularitou internetu do programu pribudli nové možnosti ako vektorové animácie a spoločnosť FutureWave vydala program FutureSplash Animator. Spoločnosť FutureWave v roku 1995 ponúkla na predaj svoj program FutureSplash spoločnosti Adobe, táto však v tom čase ponuku odmietla. V roku 1996 získala tento program spoločnosť *Macromedia* a vydala ho pod názvom Flash. V súčasnosti program Flash vydáva spoločnosť Adobe, ktorá ho získala v roku 2005. [4]

Adobe Flash sa využíva na tvorbu vektorových animácií, čo môžu byť aj krátke animované filmy, nakoľko je schopný k animáciám doplniť zvuky, prípadne v nich použiť reálne obrázky alebo videá. Ďalšie široké využitie platformy Adobe Flash je pri tvorbe obrovského množstva prehliadačových hier, ktoré je možné hrať na rôznych internetových portáloch. Pravdepodobne najčastejšie využitie Flash animácií, s ktorými sa používateľ stretne na internete, je vo forme animovaných a často krát až otravných reklamných bannerov. Aj preto sa Flash stáva terčom kritiky pri použití na internete. No okrem animácií a hier má Flash široké uplatnenie ako platforma pre pokročilé internetové aplikácie. Za zmienku stojí napríklad jeho bezkonkurenčná vlastnosť prehrávať prúdy videí zo služieb ako napríklad YouTube, ktoré aj napriek jeho kritike ho preferujú, lebo rôzne iné prehrávače videí založené napríklad na HTML5 zatiaľ neposkytujú plnohodnotnú náhradu Flashu. S využitím štruktúry Flex sa ľahko tvoria komplexné aplikácie, ako napríklad súčasný Internet banking Tatra Banky. [5] V neposlednom rade sa Adobe Flash používa aj pri E-Learningu ako nástroj na tvorbu interaktívnych animácií, ktoré majú pre užívateľa vzdelávací charakter.



Jedným z dôvodov ktorý mal významný vplyv na naše rozhodnutie použiť platformu Adobe Flash je možnosť vytvoriť samoinštalčný balík pre AIR (Adobe Integrated Runtime). AIR je možné bezplatne získať od spoločnosti Adobe a nainštalovať ho na Windows, Mac OS, ako aj na mobilné operačné systémy Android a iOS. To znamená, že program vytvorený pre Adobe AIR je možné nainštalovať a spustiť na každom z týchto operačných systémov s použitím jedného zdrojového kódu, teda bez potreby prispôbovať zdrojový kód pre každý operačný systém zvlášť. Tým je veľmi jednoducho zabezpečená multiplatformovosť programu AirStat.



3.2 Prostriedky

Pre tvorbu programov a animácií na báze technológie Flash vývojári používajú grafické vývojové prostredie Adobe Flash Professional. Toto prostredie je však predovšetkým určené na tvorbu animácií a nie je úplne vhodné pre písanie programu, aj keď to nie je úplne nemožné. V tomto prostredí môže grafik použiť sadu nástrojov na kreslenie jednoduchých tvarov, ktoré je možné následne podľa jeho vôle jednoducho animovať pomocou animačných nástrojov. Tiež je možné doň importovať obrázky, zvuky a videá a následne ich použiť v animáciách.

Pre vývoj programu AirStat sme sa preto rozhodli použiť vývojové prostredie *Adobe Flash Builder 4.7* založený na vývojovom prostredí Eclipse. Flash Builder je programové vývojové prostredie plne zamerané na vývoj softvéru. Nie je možné v ňom kresliť tvary ako v prostredí Flash Professional ani jednoducho animovať pomocou nástrojov. Ako skúsení programátori sme však vedeli v prípade potreby kresliť či animovať priamo pomocou príkazov v zdrojovom kóde. Vývojári komplexných interaktívnych animácií, ako napríklad hier však využívajú obe tieto vývojové prostredia. Vo Flash Professional nakreslia objekty a vytvoria potrebné animácie a vo Flash Builder spravia všetok aplikačný softvér.



Pre AirStat sme sa rozhodli využiť hlavne štruktúru Flex 4.12.0. Flex je sada programových nástrojov, ktoré značne rozširujú štandardné programové možnosti jazyka ActionScript hlavne pri tvorbe aplikácií a to poskytnutím nástrojov na preberanie dát z internetu, zobrazovanie dát prostredníctvom tabuliek a grafov a mnoho ďalších funkcií, ktoré sme využili v projekte a bez ktorých by sa vývoj programu značne predĺžil a skomplikoval.

3.2.1 Flex

Flex bol pôvodne vyvinutý spoločnosťou Adobe ako sada nástrojov, narádií, knižníc a doplnkov pre uľahčenie vývoja Flash aplikácií. Štandardné Flash aplikácie je možné písať aj v grafickom vývojovom prostredí

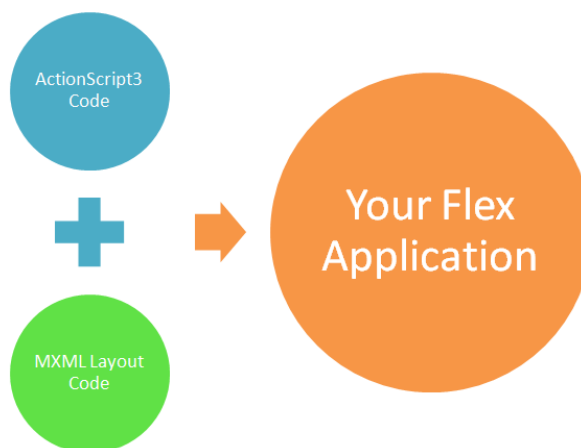


Flash Professional, no Flex je možné použiť len v prostredí Flash Builder. Flex vývojárom uľahčuje vývojový proces, nakoľko redukuje veľa repetitívnych úloh, ktoré by vývojár musel inak robiť manuálne sám. Spoločnosť Adobe prestala v roku 2011 ďalej vyvíjať Flex a venovala ho nadácii Apache ešte vo verzii 4.6.0. Apache je dobrovoľná komunita programátorov, vyvíjajúcich aj iné užitočné nástroje a po získaní knižníc Flex, pokračuje aktívne ďalej v jeho vývoji. Keď sme začali pracovať na tomto projekte bol Flex aktuálne vo verzii 4.10.0 a v čase dokončenia projektu sme už pracovali s verziou 4.12.0. V každej ďalšej verzii komunita Apache opravovala existujúce chyby a rozširovala funkčnosť Flexu najmä o jeho podporu na mobilných zariadeniach.

Flex umožňuje vývoj aplikácií pre Android, iOS, BlackBerry, ako aj tradičných prehliadačových a desktopových aplikácií s použitím jedného zdrojového kódu, programovacieho modelu a nástrojov. Flex funguje samostatne a na publikáciu projektov nepotrebuje Flash Professional. Flex bol vždy open-source a teda publikovanie aplikácií vytvorených vo Flexe je bezplatné. Na druhej strane Flash Professional aj Flash Builder ostávajú vo vlastníctve spoločnosti Adobe a ako komerčné aplikácie je ich použitie spoplatnené.

3.2.2 ActionScript + XML

Flex pre tvorbu aplikácií kombinuje zdrojový kód ActionScript a rozloženie objektov na scéne v XML štruktúre. Kód ActionScript riadi činnosť programu a XML štruktúra sa stará o rozdelenie objektov na scéne. Pomocou XML štruktúry je teda možné do programu vložiť a podľa potreby rozmiestniť objekty na scéne ako sú tlačidlá, tabuľky, textové polia, grafy a mnohé ďalšie. ActionScript kód potom určuje ako sa majú jednotlivé objekty na scéne správať a ako má program reagovať na podnety používateľa. [6]



Obr. 1 Ilustrácia spojenia ActionScript+XML
Zdroj: [6]

3.2.3 Štruktúra MVC

Objektovo orientované programovanie je forma programovania, kde sa na jednotlivé celky programu pozerá ako na Objekty. Býva zvykom, že priamo v programovacom jazyku sú jednotlivé Objekty pomenované začiatočným veľkým písmenom. Programátor tak hneď vie, že sa jedná o Objekt v programe a nie napríklad o premennú, alebo o metódu. Pre zdôraznenie Objektov budeme aj v našej bakalárskej práci označovať programové Objekty začiatočným veľkým písmenom.

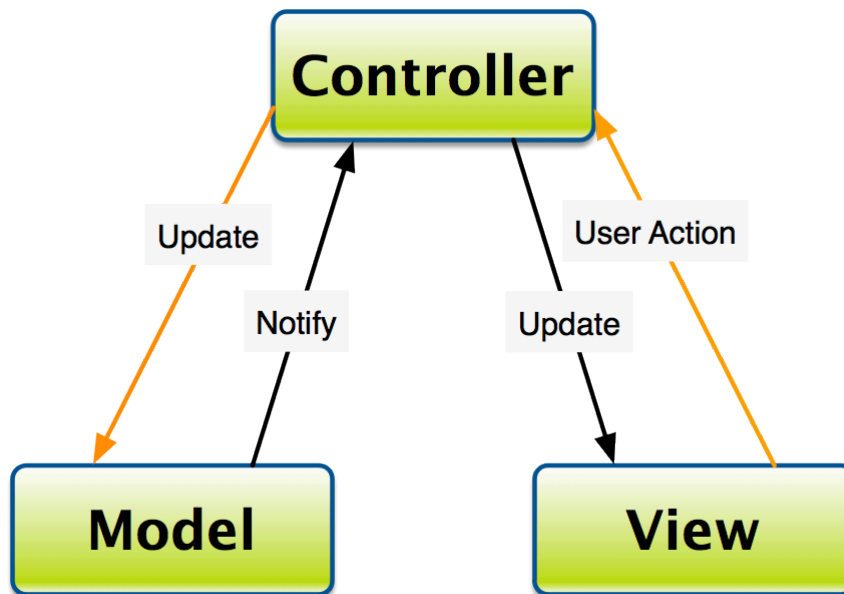
Každý Objekt má svoje vlastnosti a metódy. Vlastnosti, alebo atribúty, resp. premenné, definujú v akom stave sa daný Objekt nachádza. Vlastnosťou Objektu môže byť každá premenná, napríklad číslo alebo text. Metódy alebo funkcie sú súbory príkazov alebo aj iných funkcií, ktoré sa vykonajú po spustení danej metódy. Metóda môže napríklad spracovať vstupné údaje a poskytnúť ich výsledok podľa použitých postupov.

Jednotlivé Objekty sú zapuzdrené a z vonkajšieho pohľadu nie je pre programátora dôležité vedieť, čo sa nachádza v ich vnútri. Dôležité je vedieť, ako ich použiť. Používateľ napríklad nemusí vedieť ako funguje televízor na to, aby ho vedel ovládať diaľkovým ovládačom a televízor pre neho predstavuje zapuzdrený Objekt.

Podľa toho ako si v programe oddelíme jednotlivé Objekty, dostaneme určitú štruktúru, ktorá nám oddeľuje Objekty do logických celkov - komponentov. Oddelovanie

kódu do týchto komponentov robíme preto, aby bol zdrojový kód prehľadnejší a použiteľný aj v iných projektoch, nakoľko jednotlivé Objekty môžeme z projektu zobrať a jednoducho ich použiť v inom projekte.

Jedna z najpoužívanejších metód oddelenia Objektov zdrojového kódu je metóda MVC – Model View Controller. Touto metódou oddeľujeme Objekty zdrojového kódu do troch komponentov.



Obr. 2 Typická spolupráca MVC komponentov
Zdroj: [7]

Komponent Model obsahuje Objekty, ktoré uchovávajú údaje. Tieto Objekty nijako údaje nespracovávajú, len ich udržujú v pamäti a poskytujú pre ďalšie časti programu podľa požiadaviek.

Komponent View obsahuje Objekty, ktoré slúžia na priame zobrazenie údajov užívateľovi. Je to všetko to, čo vidíme na monitore počítača. Všetky tlačidlá, textové polia, tabuľky, grafy a podobne. Takisto do tejto skupiny patria Objekty, ktoré prijímajú informácie od užívateľa na ďalšie spracovanie programom. V programe AirStat, ktorý využíva štruktúru Flex, sa o vykreslenie a zobrazenie Objektov typu View stará XML rozloženie.

Komponent Controller obsahuje Objekty, ktoré sa starajú o logické správanie programu a prepojenie podskupín Model a View. Všetka logika a všetky výpočty sú

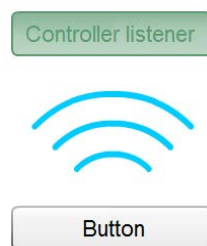
vykonávané práve v Objektoch skupiny Controller. Táto skupina ale žiadne údaje neuchováva, ale ich ukladá do skupiny Model.

Jednotlivé komponenty tejto štruktúry, Model, View a Controller, sa z programového hľadiska nazývajú Objekty, alebo aj Triedy a zo súborového hľadiska bývajú uložené v samostatných súboroch. Tieto súbory môžeme potom v prípade potreby jednoducho prekopírovať do iného projektu a použiť ich tam.

3.2.4 Data binding

Veľmi užitočnou vlastnosťou, ktorú nám poskytuje štruktúra Flex a ktorá nám značne zjednodušuje prácu, je takzvané previazanie dát, data binding.

ActionScript, ako objektovo orientovaný programovací jazyk využíva štandardne systém hlásenia udalostí. Keď niektorý Objekt v programe zaznamená udalosť, ktorú je povinný hlásiť, vyšle do systému o tejto udalosti hlásenie a ak iný Objekt počúva toto hlásenie, vie pri danej udalosti patrične reagovať. Napríklad pre tlačidlo na scéne je dôležitá udalosť, ak užívateľ na tlačidlo klikne. Ak sa tak stane, toto tlačidlo vyšle do systému hlásenie o udalosti ktorým oznamuje, že bolo naň kliknuté. Iný Objekt v programe, napríklad niektorá riadiaca trieda Controller, môže „počúvať“ hlásenia tohto konkrétneho tlačidla a keď zachytí hlásenie o kliknutí, bude podľa nastavení reagovať, napríklad zobrazí pre používateľa nejaké menu.



Ako vývojári programu musíme toto všetko manuálne nastaviť. Tlačidlo ako komponent má už v sebe zahrnuté hlásenia o kliknutí, prejdení kurzorom myši, opustením kurzora myši a mnoho ďalších, z pohľadu používateľa nezaujímavých udalostí, ako napríklad, hlásenie umiestnenia na scénu, odstránenie zo scény, získanie zamerania programu a podobne. Aby niektorá kontrolná trieda mohla počúvať udalosti tlačidla, musíme tomuto tlačidlu pri vývoji programu nastaviť „počúvanie“ a reakciu naň. Pri písaní rozsiahlejšieho programu sa tento prístup opakuje veľa krát pre mnoho rôznych tlačidiel a ďalších Objektov na scéne a stáva sa pre nás, ako programátorov, monotónny až otravný. Navyše pri optimalizácii zdrojového kódu, ktorou sa snažíme efektívne využívať

výpočtovú kapacitu procesora počítača, je potrebné počúvanie na udalosť vypnúť, keď už nie je potrebné, napríklad pri odstránení nepotrebného tlačidla zo scény.

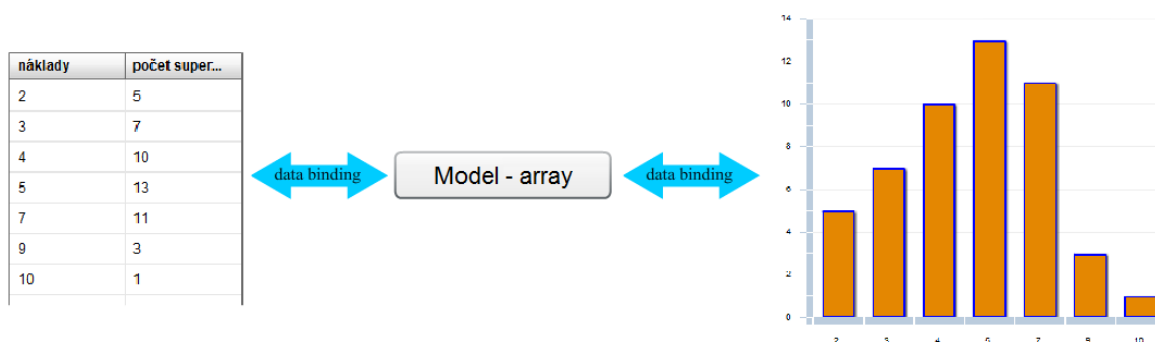
Štruktúra Flex tento monotónny postup automatizuje pomocou takzvaného previazania dát, *data binding*. Danému tlačidlu na scéne je možné, pri písaní zdrojového kódu v XML štruktúre, jednoducho priradiť funkciu, ktorá sa zavolá po jeho kliknutí, resp. inej udalosti, ktorú chceme počúvať a pri odstránení tlačidla zo scény sa všetky nastavené počúvania na udalosť automaticky vypnú.

Avšak zaujímavejším použitím previazania dát vo Flexe je ich zobrazovanie z Modelu. V programe AirStat máme uložené veľké množstvo dát z tabuliek a vypočítané popisné charakteristiky štatistických údajov. Bez previazania dát by sme museli práčne každý jeden údaj z Modelu príslušným príkazom zobrazit' na scéne v tabuľkách aj vo výsledkoch výpočtov. Pomocou previazania dát nám však stačilo jednoducho danému textovému poľu nastaviť na ktorý údaj v Modeli sa má naviazať a Flex už zabezpečí, aby každá zmena údaje v Modeli bola hneď zobrazená na scéne.

Najdôležitejšie uplatnenie tejto vlastnosti Flexu v našom projekte je zobrazenie údajov do tabuliek. V Modeli sú tabuľkové údaje uložené ako sled čísiel v poli. Flex zoberie toto pole a automaticky ho zobrazí v tabuľke. Bez tejto možnosti by sme museli manuálne brať každý jeden údaj z tabuľky a vykresliť ho v tabuľke. Každá zmena, ktorú užívateľ vykoná v tabuľke je automaticky uložená do dátového Modelu. A naopak každá zmena v Modeli sa prejaví v tabuľke.

Obdobne používame previazanie dát, *data binding*, pri zobrazení údajov v stĺpcovom grafe. Zaujímavosťou je, že Flex využije rovnakú skupinu údajov z poľa v Modeli a zobrazí ich ako graf. Ak teda užívateľ vykoná nejakú zmenu v tabuľke, Flex pomocou previazania dát túto zmenu zaznamená a uloží do Modelu. Zároveň, keďže došlo ku zmene údajov v Modeli, tieto zmeny sa prejavia okamžite aj na tvare stĺpcového grafu.

Z pohľadu štruktúry MVC teda vo Flexe nie je potrebné na prepojenie dát z Modelu a ich zobrazením vo View manuálne použiť Controller na nastavenie počúvania zmien v tabuľke, ale pomocou previazania dát *data binding* je tento medzikrok vynechaný a prepojenie je automatické. Samozrejme, že o toto prepojenie sa vo Flexe stará určitá trieda Controller, ale pre nás ako programátorov je táto trieda skrytá a nemusíme na ňu pri tvorbe programu myslieť.



Obr. 3 Zobrazenie prepojenia tabuľky a grafu s použitím previazania dát

Zdroj: Vlastné spracovanie

Tým sa pre nás zvyšuje rýchlosť vývoja programu, nakoľko nie sme zaťažovaní monotónnymi úlohami a môžeme sústrediť náš čas a úsilie podstatnejším aspektom programu AirStat.

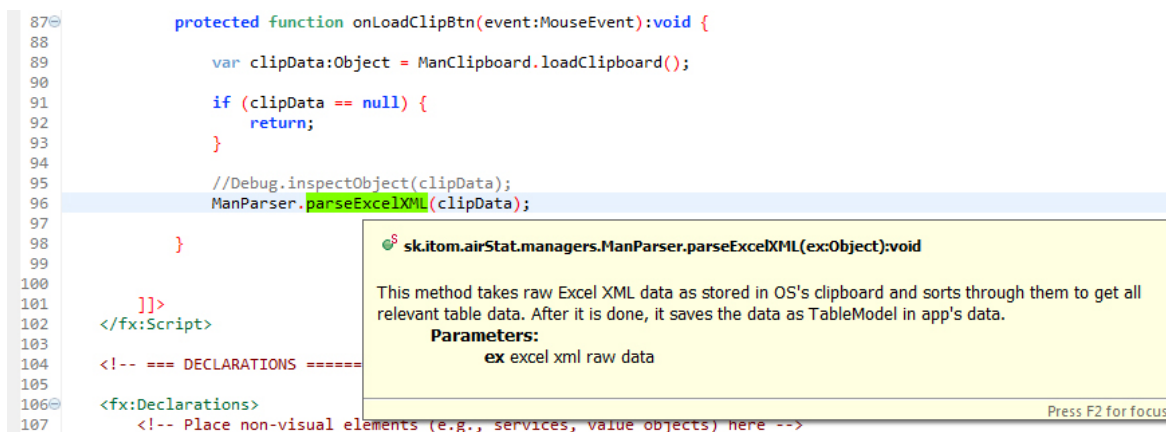
3.2.5 ASDoc

V každom programovacom jazyku programátor používa komentáre, ktoré mu umožňujú ľahšie sa orientovať vo svojom alebo cudzom zdrojovom kóde. Komentáre sú bloky zdrojového kódu označené špecifickými značkami a program ich jednoducho ignoruje. Sú určené len pre programátorov a vysvetľujú im činnosť jednotlivých častí zdrojového kódu. Uľahčujú tak programátorom pochopenie činnosti zdrojového kódu bez potreby práce študovať funkčnosť použitých príkazov.

Pri vývoji programu môžeme jednotlivé príkazy v zdrojovom kóde znefunkčniť aplikovaním komentárov bez toho, aby sme ich museli vymazať a pozorovať tak, ako sa zmení správanie programu.

Špeciálnou verziou komentárov sú dynamické komentáre charakteristické pre niektoré programovacie jazyky. Pre programovací jazyk ActionScript je takýmto komentárom ASDoc. Je to špeciálny komentár, ktorý sa používa na zdokumentovanie zdrojového kódu. Vo všeobecnosti môžeme ako programátori tento komentár priradiť k jednotlivým metódam a premenným našich Objektov. Komentár ASDoc bude vysvetľovať čo konkrétna metóda vykonáva a aké parametre prijíma na svojom vstupe, prípadne aké hodnoty môžeme očakávať na jej výstupe.

Podstatnou prednosťou komentárov typu ASDoc oproti štandardným komentárom v zdrojovom kóde je, že k nim môžeme pristupovať aj bez potreby otvoriť konkrétny súbor, kde sú napísané. Ak napríklad z komponentu View chceme nastaviť reakciu na kliknutie tlačidlom myši a zavolať konkrétnu metódu z Controllera, nemusíme otvárať Triedu s týmto Controllerom, aby sme si prečítali, čo táto metóda vykonáva, ale vo vývojovom prostredí Flash Builder sa nám zobrazí priamo pri tejto metóde jej popis, ktorý sme si pred tým k tejto metóde napísali.



Obr. 4 ukážka zobrazenia komentára ASDoc vo vývojovom prostredí Flash Builder
Zdroj: Vlastné spracovanie

Viac ako pre nás je tento komentár užitočný pre iných programátorov, ktorí by tento projekt alebo Triedy z neho prevzali po nás, resp. ak by sme my prebrali Triedy od iných programátorov a chceli ich použiť v našom projekte. Preto sme aj my v našom projekte napísali všetky ASDoc komentáre v angličtine.

Komentáre ASDoc nie sú tvorené len nami v zdrojovom kóde, ale sú k dispozícii aj oficiálne od spoločnosti Adobe, respektíve od komunity Apache pre metódy a premenné nimi poskytovaných Tried.

3.3 Postupy

Vstavané prostriedky a metódy v štruktúre Flex, ako aj naše vlastné metódy a postupy sme použili pri riešení konkrétnych postupov v našom programe. Od prijatia údajov, cez ich spracovanie a uloženie nasledované výpočtami popisných charakteristík až po zobrazenie výsledkov spracovania, tabuliek a stĺpcového grafu na scéne.

3.3.1 Údaje

Program AirStat sme konštruovali tak, aby mal používateľ možnosť vytvoriť si novú tabuľku a manuálne do nej vkladať údaje. Tento spôsob je však prácny, hlavne keď má používateľ väčšinou svoje zdrojové štatistické údaje uložené najčastejšie v tabuľkách programu Excel.

Pre jednoduchosť sme preto do programu AirStat implementovali možnosť vložiť zdrojové štatistické údaje, priamo z tabuliek programu Excel, pre používateľa čo najjednoduchším možným spôsobom a to kopírovaním a vložením zvolenej tabuľky z programu Excel. Programy založené na platforme Adobe Flash majú obmedzenú schopnosť použitia pravého tlačidla myši a preto vloženie údajov skopírovaných z Excelu do nášho programu nie je možné s použitím kontextovej ponuky vyvolanej kliknutím pravého tlačidla myši. Na vloženie týchto údajov sa preto v programe AirStat nachádza tlačidlo load clipboard. Okrem toho pravé tlačidlo myši nie je použiteľné pri mobilných zariadeniach ovládaných len dotykmi prstov.

Takto prenesené údaje sa do programu AirStat dostanú v HTML formáte. Tu sme pri písaní zdrojového kódu programu narazili na jeden z prvých problémov a to nájsť v tomto HTML formáte relevantné údaje a uložiť ich ako pole čísel. Jedná sa v podstate o formu data miningu, keď program hľadá v HTML formáte relevantné údaje.



Obr. 5 prenos údajov z programu Excel cez HTML štruktúru do programu AirStat
Zdroj: Vlastné spracovanie

Náš program AirStat očakáva prijaté údaje uložené v stĺpcoch. Dokáže rozpoznať, či vložená tabuľka v HTML formáte obsahuje jeden, dva, alebo viac stĺpcov. Ak má jeden stĺpec, program ju považuje za obyčajný súbor hodnôt. V prípade, že má dva stĺpce, tak prvý stĺpec považuje program za číselnú hodnotu a druhý stĺpec za jej početnosť. Program AirStat pracuje so štatistickým súborom vo formáte dvoch stĺpcov, teda ako hodnoty

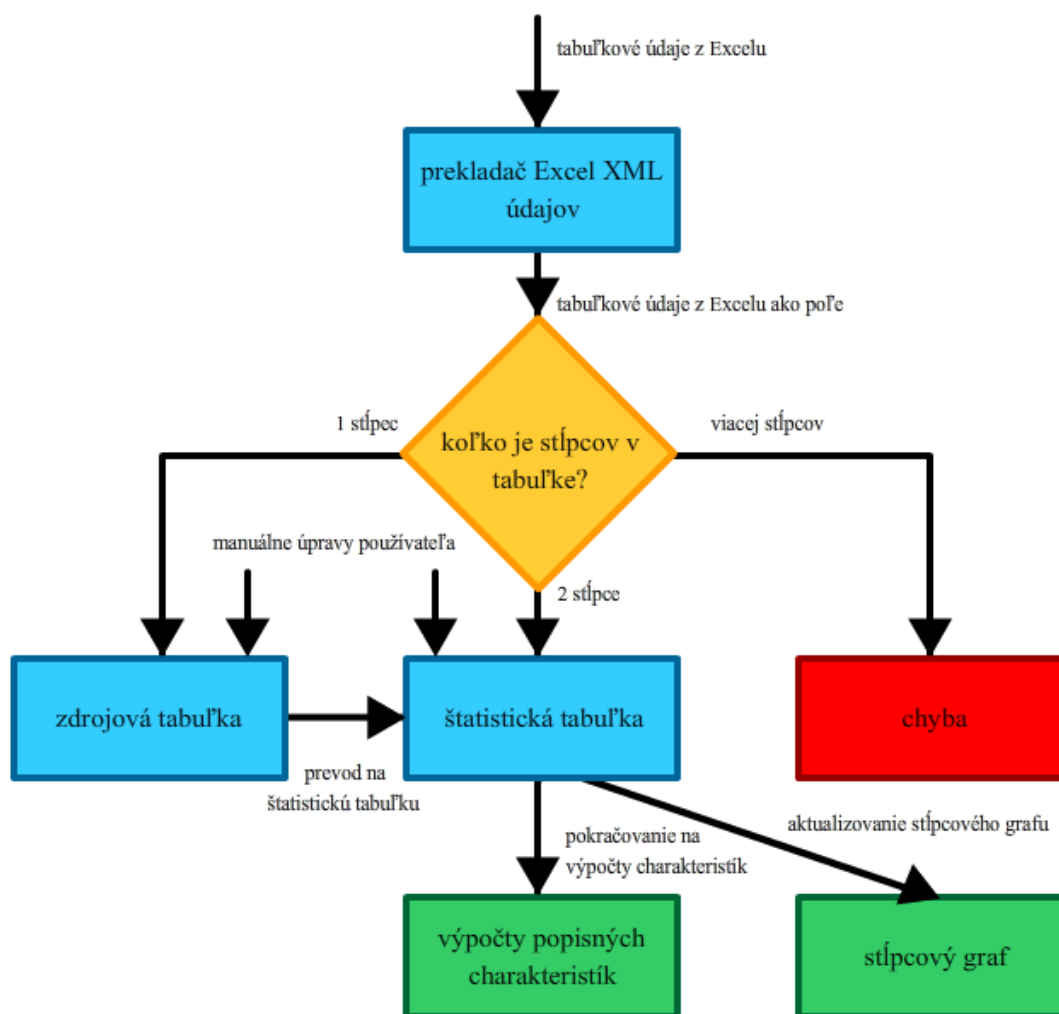
a početnosti. Pri vložení zdrojového štatistického súboru s jedným stĺpcom, pretvorí tento zdrojový súbor na formát dvoch stĺpcov a každej hodnote priradí jej početnosť.

Samotné hodnoty prijaté z programu Excel môžu obsahovať aj desatinné čísla, ale program AirStat ich správne rozpozná len vtedy, keď je ako oddeľovač desatinných miest použitá bodka a nie čiarka, ako to býva všeobecne pri väčšine programovacích jazykov. Ide o nedokončený program a toto je jedna z vecí, ktorú môžeme pri jeho ďalšom vývoji vyriešiť.

Ak by prvý riadok vložených tabuliek z programu Excel obsahoval údaje iné ako číselné, program AirStat použije tieto údaje ako názvy stĺpcov tabuľky. Ak programu AirStat dodáme údaje vo viac ako dvoch stĺkoch, program nebude schopný tieto údaje spracovať a vypíše na monitore chybové hlásenie. Obdobne vypíše chybové hlásenie, ak nenájde pri kopírovaní žiadne údaje.

3.3.2 Spracovanie

Po vložení štatistických údajov do novej tabuľky v programe AirStat, alebo po vložení údajov kopírovaním z existujúcej tabuľky programu Excel, sa tieto uložia ako pole v Modeli programu a spustí sa proces spracovania prijatých dát.



Obr. 6 Spracovanie a rozdelenie prijatých štatistických údajov

Zdroj: Vlastné spracovanie

Pole údajov prijaté z programu Excel sa musí v programe AirStat spracovať na tabuľkový formát. Ak ide o pole údajov s jedným stĺpcom, uloží sa v pamäti programu ako zdrojová tabuľka s jedným stĺpcom a táto sa pomocou previazania dát data binding ihneď zobrazí na scéne v zdrojovej tabuľke. Následne dôjde k jej konverzii na štatistickú tabuľku, teda na tabuľku s dvomi stĺpcami, kde v prvom sú hodnoty a v druhom početnosť. Ak základné pole údajov získané z Excelu má dva stĺpce, ihneď sa chápe ako štatistická tabuľka a tak sa aj uloží. Následne dôjde vďaka data bindingu k jej zobrazeniu na scéne v tabuľke so štatistickými údajmi a súčasne k zobrazeniu stĺpcového grafu, ktorý používa spoločný zdroj údajov. Ak by základné pole údajov z Excelu obsahovalo viac ako dva stĺpce, program AirStat túto tabuľku nespracuje a vyhlási chybu pri spracovaní údajov.

Používateľ je súčasne schopný meniť údaje v oboch tabuľkách. Zmena, ktorú vykoná v zdrojovej tabuľke sa následne prenesie do štatistickej tabuľky. Zmena

v štatistickej tabuľke nebude mať v súčasnom stave vývoja programu AirStat žiaden vplyv na zdrojovú tabuľku.

Po vložení tabuľky z programu Excel, resp. po každom zásahu užívateľa do dát zdrojovej či štatistickej tabuľky, program spustí sériu výpočtov pre získanie základných popisných charakteristík štatistického súboru.

3.3.3 Výpočty popisných charakteristík

Okrem vstavanej Flex funkcie previazania dát sme na zdrojovú aj štatistickú tabuľku s dátami nastavili vlastnú funkciu na počúvanie udalostí zmien v týchto tabuľkách, ktorá spôsobí, že pri zmene údajov v tabuľke program zavolá nami vytvorené metódy, ktoré tieto zmeny ďalej spracujú.

Ak používateľ vykoná zmenu v údajoch štatistickej tabuľky, program spustí metódy na získanie štatistických popisných charakteristík. Ak používateľ vykoná zmenu v údajoch zdrojovej tabuľky, dôjde najprv ku konverzii zdrojovej na štatistickú tabuľku. Tým sa kompletne prepíše existujúca štatistická tabuľka zdrojovou tabuľkou pretvorenou na štatistickú, čo program eventuálne tiež zaznamená ako zmenu v údajoch štatistickej tabuľky a následne spustí metódy na získanie štatistických popisných charakteristík. Tieto výpočty sme v našom programe rozdelili do desiatich po sebe idúcich krokov.

Prvým krokom pred samotnými výpočtami je vypnutie nami nastaveného počúvania na udalosti zmien údajov v štatistickej tabuľke. To preto, lebo ďalšie dva kroky budú priamo zasahovať do týchto údajov a modifikovať ich. Keby počúvanie ostalo zapnuté, došlo by k nekonečnému cyklu, kedy by sa neustále volali metódy na vykonanie výpočtov popisných charakteristík.

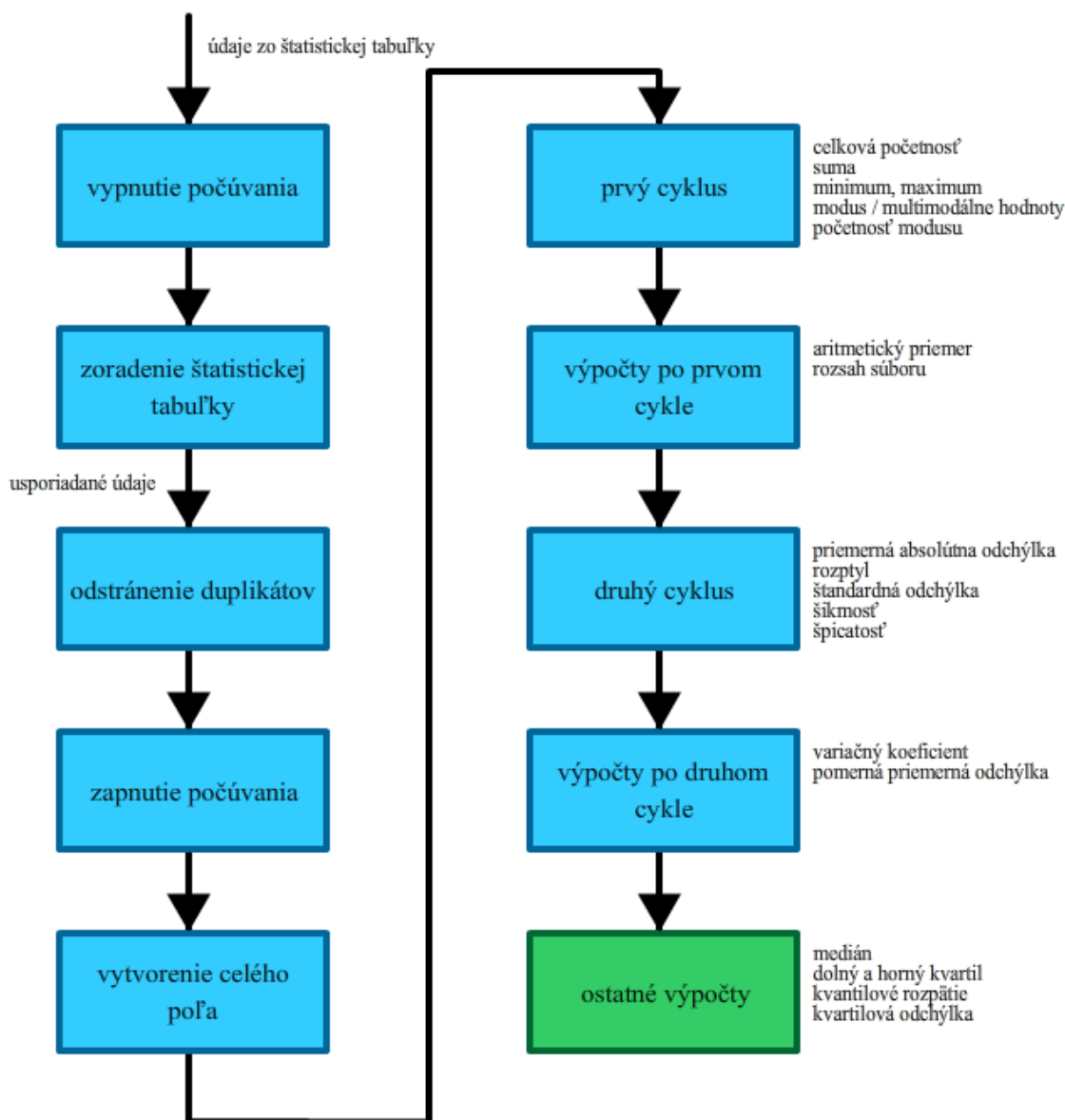
Druhý krok je zoradenie tabuľky. Nová vložená tabuľka z programu Excel, alebo tabuľka modifikovaná používateľom nemusí mať všetky prvky usporiadané v poradí od najmenšieho po najväčšie. A toto poradie potrebujeme pre ďalšie štatistické výpočty. Na samotné zoradenie údajov tabuľky sme použili vstavanú funkciu na radenie dát podľa prvého stĺpca, teda podľa stĺpca s údajmi o hodnotách. O samotné zoradenie dát sa už postaral Flex, ktorý má na to vstavané metódy. Ak by ich nemal, museli by sme prácne

použiť niektorú z algoritmických metód na radenie dát, napríklad Bubble Sort alebo Merge Sort. Tu sa znova ukázalo, ako Flex zjednodušuje vývoj programu.

Tretím krokom je odstránenie duplikátov. Vo vloženej tabuľke z programu Excel, alebo aj v údajoch vložených manuálne užívateľom sa môžu vyskytovať údaje rovnakej hodnoty. Pri štatistickej tabuľke však program pracuje s hodnotami a ich početnosťami, preto keď objaví viac krát sa opakujúce hodnoty, tak ponechá iba jednu, ktorej nastaví rovnakú početnosť ako je množstvo duplikátov a duplikáty odstráni. Vzhľadom na to, že súbor údajov je už usporiadaný, tak všetky duplikáty sa nachádzajú v rade za sebou, čo odbúrava potrebu prehľadávať celý štatistický súbor pre výskyt ďalších duplikátov jednej hodnoty. Keďže pracujeme so špecifickým súborom dát, teda hodnotami a početnosťami, Flex nemá vstavané metódy na odstránenie duplikátov a preto sme túto metódu museli vypracovať manuálne.

Predchádzajúce dva kroky priamo zasahovali do údajov štatistickej tabuľky a keby ostala zapnutá nami nastavená funkcia na počúvanie zmien v štatistickej tabuľke, v programe by to spôsobovalo nekonečné volanie metód na výpočty popisných charakteristík. Ďalšie kroky na výpočet popisných charakteristík však už nezasahujú do údajov štatistickej tabuľky a preto v štvrtom kroku môžeme znova zapnúť v programe počúvanie zmien údajov v tejto tabuľke. To že sme vypli nami nastavené počúvanie zmien nemalo vplyv na vstavanú funkciu Flexu previazania dát, data binding, a s každou úpravou v štatistickej tabuľke sa tieto zmeny prejavili aj v zobrazení štatistickej tabuľky na scéne ako aj v stĺpcovom grafe. Tieto zmeny však prebehli tak rýchlo, že ich používateľ nestihol postrehnúť.

Piaty krok predstavuje vybranie údajov z tabuľky do jedného poľa. V Modeli obsahujúcom tabuľku sú totiž údaje uložené v poli po jednotlivých riadkoch. Keďže štatistická tabuľka obsahuje dva stĺpce, jeden riadok poľa údajov obsahuje dva záznamy: hodnotu a početnosť. Pre niektoré ďalšie výpočty, ako napríklad rôzne kvantily, je však vhodnejšie použiť zoradené pole čísiel a preto týmto krokom program vyberie všetky hodnoty z tabuľky a uloží ich do poľa vzostupne za sebou, každú toľko krát, koľko je jej početnosť. Toto pole používateľ nevidí na scéne a je len pre interné použitie v programe na ďalšie výpočty, aby sa program nemusel neustále odkazovať na zdrojové údaje tabuľky a prerátavať ich početnosť.



Obr. 7 Postup spracovania údajov pri výpočte popisných charakteristík
Zdroj: Vlastné spracovanie

V šiestom kroku sa spustí prvý cyklus cez všetky údaje. Program AirStat v tomto kroku prechádza postupne cez všetky údaje v štatistickej tabuľke a vykonáva hneď niekoľko matematických operácií, ktoré je možné v prvom cykle získať. Ide konkrétne o:

- celkovú početnosť, ktorú získa ako súčet jednotlivých početností všetkých údajov,
- sumu, ktorú dostane sčítaním násobkov hodnôt a ich početností,
- minimálnu a maximálnu hodnotu, ktoré získa pri prvej hodnote tabuľky uložením tejto hodnoty ako minimum aj ako maximum a pri každej ďalšej hodnote sa uložené minimum a maximum porovnáva s hodnotou tabuľky a uloží sa, ak je väčšia ako uložená pre maximum, alebo menšia ako uložená pre minimum a

- o modus, pri prvej hodnote si program uloží aj túto hodnotu a početnosť ako hodnotu a početnosť modusu a pri každej ďalšej hodnote porovnáva jej početnosť s uloženým modulusom, pričom ak je početnosť väčšia, prepíše uložený modus a početnosť a ak je rovnaká, môže ísť o multimodálny štatistický súbor a program vtedy uloží obe hodnoty.

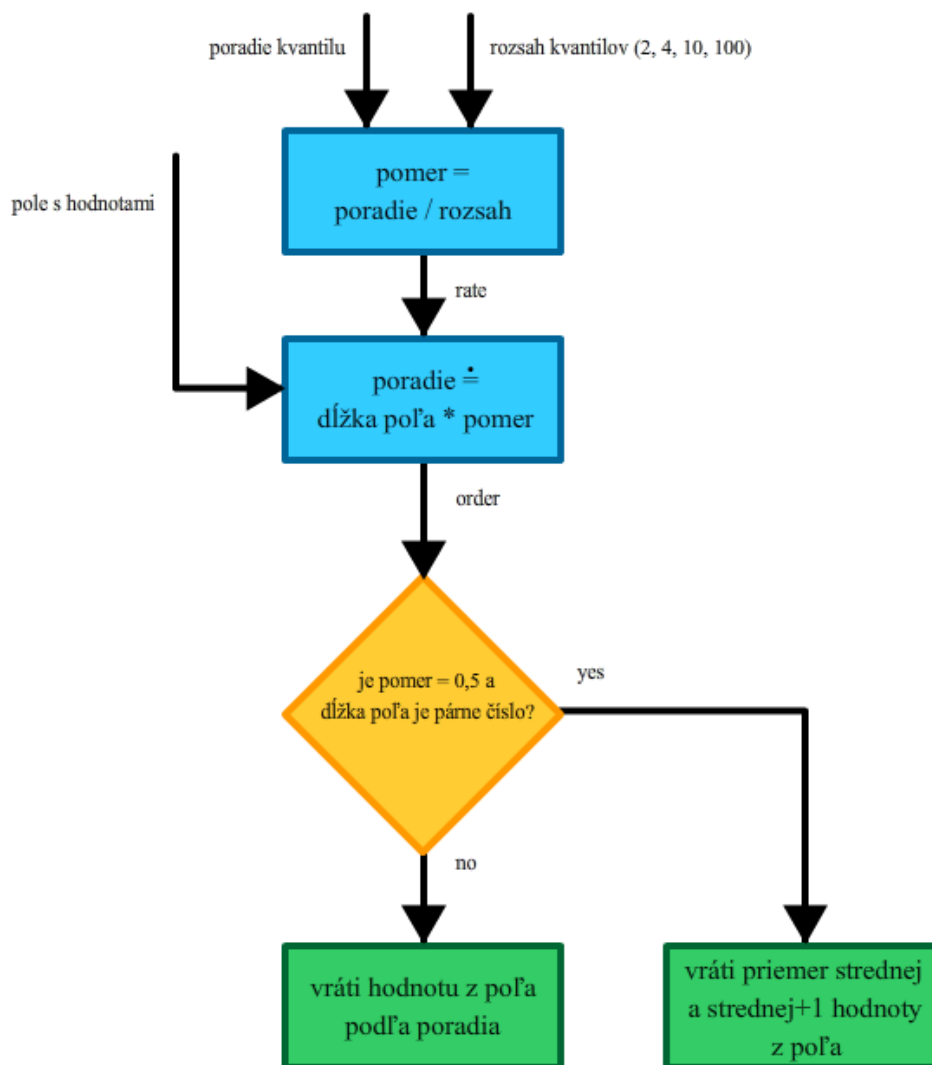
Po prvom cykle sa v siedmom kroku vykonajú matematické operácie, na ktoré sme v programe potrebovali poznať údaje získané z prvého cyklu. Ide o aritmetický priemer, vypočítaný ako podiel sumy a počtu údajov a rozsah súboru získaný ako rozdiel maximálnej a minimálnej hodnoty.

V ôsmom kroku sa spustí druhý cyklus, kedy program musí znova prejsť cez všetky údaje štatistickej tabuľky, aby vykonal ďalšie výpočty s použitím hodnôt, ktoré mohol získať len tak, že najprv už cez všetky hodnoty už raz prešiel, ako napríklad aritmetický priemer. Výpočty vykonávané v samotnom cykle sú relatívne jednoduché. Najprv sa získa číselná hodnota ako násobok premennej a početnosti v jednom riadku tabuľky od ktorej sa odpočíta aritmetický priemer, teda operácia $x_i - \bar{x}$. Tejto hodnote následne priradíme absolútnu hodnotu a exponenty 2, 3 a 4, čo program dočasne uloží ako 4 nezávislé hodnoty $\sum|x_i - \bar{x}|$, $\sum(x_i - \bar{x})^2$, $\sum(x_i - \bar{x})^3$ a $\sum(x_i - \bar{x})^4$, ktoré program použije na výpočet

- o priemernej absolútnej odchýlky, ako podiel absolútnej sumy hodnôt a celkovej početnosti $\frac{\sum|x_i - \bar{x}|}{n}$,
- o rozptylu, ako podiel sumy druhých mocnín hodnôt a celkovej početnosti $\frac{\sum(x_i - \bar{x})^2}{n}$,
- o štandardnej odchýlky, ako hodnotu druhej odmocniny rozptylu,
- o šikmosti, ako podiel sumy tretej mocniny hodnoty a početnosti a tretej mocniny štandardnej odchýlky $\frac{\sum(x_i - \bar{x})^3}{\frac{n}{s^3}}$ a
- o špicatosti, ako podiel sumy štvrtej mocniny hodnoty a početnosti a štvrtej mocniny štandardnej odchýlky $\frac{\sum(x_i - \bar{x})^4}{\frac{n}{s^4}}$. [3]

V deviatom kroku sa obdobne ako po prvom cykle vykonajú výpočty, ku ktorým sú potrebné hodnoty z druhého cyklu. Ide o variačný koeficient a pomernú priemernú odchýlku, ktoré sme dostali zo štandardnej odchýlky a z priemernej absolútnej odchýlky.

V poslednom kroku sa vykonávajú ostatné výpočty. V súčasnom stave programu sú to hlavne kvantily. Pre výpočet kvantilov sme si v programe vytvorili vlastnú univerzálnu metódu, ktorá prijíma dva parametre na výpočet a to poradie kvantilu, ktorý chceme vypočítať a celkové množstvo kvantilov. Na základe týchto údajov vie táto metóda vypočítať rôzne kvantily, či už ide o medián, kvartily, decily alebo percentily.



Obr. 8 Programový princíp výpočtu kvantilov
Zdroj: Vlastné spracovanie

V tejto metóde program najprv vypočíta pomer hľadaného kvantilu. Pre medián je to napríklad $\frac{1}{2} = 0,5$. Pri dolnom kvartile je to $\frac{1}{4} = 0,25$ a podobne. Pomer sa potom použije na výpočet poradia prvku, ktorý hľadáme v súbore prvkov a to ako zaokrúhlená hodnota násobku dĺžky štatistického súboru a pomeru. Pri mediáne ide o špeciálny prípad, kde ak je párny počet hodnôt v štatistickom súbore, výsledná hodnota je priemer

prostrednej a prvej nasledujúcej hodnoty. Ak metóda nehľadá medián, alebo ak je počet hodnôt v štatistickom súbore nepárny, metóda vráti hodnotu poradia štatistického súboru.

Desiatym krokom s použitím tejto metódy teda získame popisné charakteristiky ako medián, dolný kvartil, horný kvartil, kvantilové rozpätie a kvartilovú odchýlku.

Týmito desiatimi krokmi program AirStat vypočítal všetky v ňom nastavené popisné charakteristiky a uložil ich do príslušného pamäťového miesta v Modeli. Postupným ukladáním týchto údajov do Modelu došlo vďaka dátovému prepojeniu Flexu data binding k automatickému zobrazeniu údajov na scéne v tabuľke popisných charakteristík. Tieto výpočty však prebiehali veľmi rýchlo a tak sa zdanlivo všetky popisné charakteristiky v tabuľke objavili súčasne a nie postupne. Pri ďalšom vývoji programu v budúcnosti, keď by sme program obohatili o ďalšie popisné charakteristiky, môžeme metódy na ich výpočet doplniť na príslušné miesta v tomto desať - krokovom postupe na príslušné miesta podľa toho či na svoj výpočet potrebujú údaje získané z prvého alebo druhého cyklu.

Programom AirStat je možné zobrazit' na scéne aj ďalšie doplňujúce informácie o danej popisnej charakteristike. Napríklad by sa ukázaním myšou na danú popisnú charakteristiku mohli zobrazit' podrobnejšie údaje o nej, najmä to, ako ju program vypočítal zobrazením konkrétneho vzorca s doplnenými číselnými údajmi. V súčasnom stave vývoja programu sme však túto funkciu nevytvorili, no na ukážku sme v programe pre každú popisnú charakteristiku vytvorili tooltip, ktorý sa zobrazí po ukázaní myšou naň. Tu môže nastať drobný problém s implementáciou na mobilných zariadeniach, keďže tieto nemajú možnosť prejsť kurzorom myši na niektorý objekt na scéne.

Akýkoľvek ďalší zásah používateľom do zdrojových alebo štatistických tabuliek, či už manuálnou úpravou alebo vložením novej tabuľky z programu Excel, bude mať za následok opätovné spustenie metód na získanie popisných charakteristík štatistického súboru ako aj automatickú aktualizáciu zobrazenia tabuľky a stĺpcového grafu cez data binding. Existujúce popisné charakteristiky sú nahradené novými, ktoré program musí znova vypočítať.

4 Výsledky práce

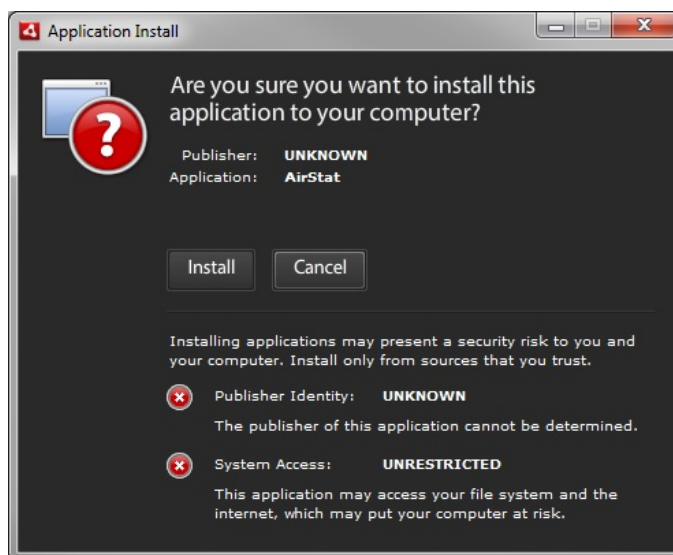
Výsledkom našej práce je samoinštalateľný balík AirStat.air pre platformu Adobe AIR, ktorý na pevnom disku zaberá menej ako 1MB miesta. Samoinštalateľný balík je možné nainštalovať na operačné systémy Windows, alebo Mac OS, na ktorých je už nainštalovaný zdarma dostupný program Adobe AIR.



V čase dosiahnutia súčasnej fázy vývoja programu AirStat nie sú jeho zdrojové súbory dostupné na verejnom portáli. Ich zverejnenie by sme chceli ešte podmieniť opravením drobných kozmetických chýb, ktoré sa v programe stále nachádzajú.

4.1 Inštalácia

Pre potreby prezentácie bol inštalateľný balík vytvorený s vlastným a teda certifikačnou autoritou nepotvrdeným certifikátom a pri inštalácii môže vyhlásiť varovanie pred inštalovaním potenciálne nebezpečného softvéru.



Obr. 9 Varovanie pred inštalovaním potenciálne nebezpečného softvéru
Zdroj: Vlastné spracovanie

Tento inštalateľný balík je určený pre inštaláciu na desktopoch a teda na operačných systémoch Windows a Mac OS. Aby sme získali inštalateľný balík pre mobilné operačné systémy Android a iOS, museli by sme vo vývojovom prostredí Flash Builder vytvoriť

nový mobilný Flex projekt, ktorého výstupom by boli aplikačné balíky pre spomínané mobilné operačné systémy. Projekt pre mobilné aplikácie má vo vývojovom prostredí iný programový "obal", ale jadro ostáva rovnaké, teda môžeme použiť jeden a ten istý zdrojový kód pre oba projekty. Pri vývoji mobilnej verzie programu však musíme brať do úvahy obmedzenia mobilnej platformy, akými sú napríklad nemožnosť kliknutia pravým tlačidlom myši, či prejdienie tlačidlom myši cez nejaký objekt na scéne.

Inštalácia programu na počítači prebieha rýchlo a je veľmi jednoduchá. Používateľ si môže zvoliť, kam chce program nainštalovať a či chce mať jeho ikonu programu v ponuke menu Štart a na pracovnej ploche.

Po nainštalovaní programu si tento na mieste, kde bol nainštalovaný, vytvorí hlavný AirStat.swf súbor a ďalšie podporné súbory z inštalačného balíka. Koncovka .swf je štandardná koncovka ShockWave Flash súborov používaných na internete, no AIR dokáže tento súbor spustiť lokálne na počítači používateľa v okne. Ostatné súbory z inštalačného balíka predstavujú rôzne certifikáty a podporné informácie, ktoré potrebuje AIR na správne spustenie programu.

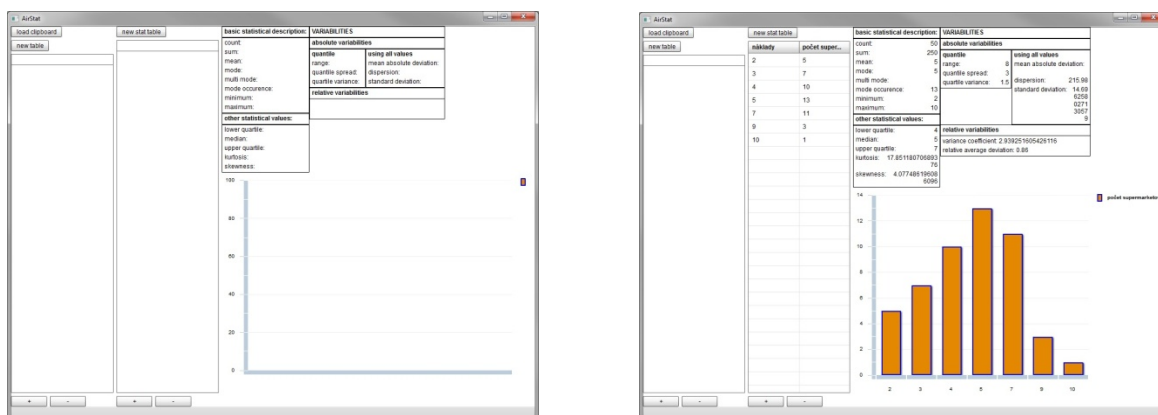
Program sa bude nachádzať aj v zozname nainštalovaných programov v nastaveniach operačného systému Windows, ako každý iný bežne inštalovateľný program. Odtiaľ ho používateľ, v prípade že ho už nebude nepotrebovať, môže jednoducho odinštalovať.

V prípade budúceho vývoja programu AirStat bude pri inštalácii aktualizácie používateľ upozornený na existenciu programu AirStat v jeho systéme a bude mu ponúknutá možnosť aktualizácie na novšiu verziu.

4.2 Používanie

Používateľ môže program AirStat spustiť ikonou na ploche alebo v ponuke menu Štart systému Windows. Spustením programu sa používateľovi zobrazí okno s dvomi prázdnyimi tabuľkami, prázdnyimi popisnými charakteristikami a prázdnyim stĺpcovým grafom. Vytvorením novej tabuľky, alebo prekopírovaním existujúcej tabuľky z programu

Excel prostredníctvom funkcie kopírovať a vložiť, sa v programe AirStat naplnia prázdne údaje v tabuľke, popisných charakteristikách a v stĺpcovom grafe.



Obr. 10 Zobrazenie programu AirStat bez údajov a naplneného údajmi
Zdroj: Vlastné spracovanie

4.2.1 Tabuľkové údaje

Tlačidlami v ľavom hornom rohu môže používateľ vytvoriť novú, alebo vložiť existujúcu tabuľku z programu Excel. V ľavej tabuľke sú zdrojové údaje a táto tabuľka sa automaticky vyplní, ak používateľ preniesie z programu Excel tabuľku s jedným stĺpcom. Druhá tabuľka v programe AirStat je tabuľka so štatistickými údajmi a vyplní sa ak používateľ preniesie z programu Excel tabuľku s dvomi stĺpcami, pričom prvý je automaticky považovaný za hodnoty a druhý za ich početnosti. Štatistická tabuľka sa takisto naplní údajmi zo zdrojovej tabuľky, ak v nej dôjde k zmene vyvolanej či už manuálnym zásahom používateľa, alebo vložení novej tabuľky z programu Excel s jedným stĺpcom. Zmenou zdrojovej tabuľky sa prepíše štatistická tabuľka bez ohľadu na to, aké údaje sa v nej nachádzajú. Vytvorením novej zdrojovej tabuľky sa na scéne objaví prvý riadok tabuľky s hodnotou 0. Okamžite na to sa v štatistickej tabuľke objaví jeden riadok s hodnotou 0 a početnosťou 1.

Pod obidvomi tabuľkami sa nachádzajú tlačidlá + a -. Týmto tlačidlami môže používateľ do existujúcich tabuliek v programe AirStat pridávať, alebo z nich odoberať jednotlivé riadky. Pridaním nového riadka do zdrojovej tabuľky sa na konci tabuľky objaví nová hodnota s číslom 0. Toľko krát, koľko používateľ zatlačí tlačidlo + sa v tabuľke objaví nový riadok s hodnotou 0. Týmto samozrejme dochádza aj k modifikácii údajov

štatistickej tabuľky, kde sa s každým pridaním nového riadka zvýši početnosť hodnoty 0. Stlačením tlačidla - pod zdrojovou tabuľkou z nej odstráni posledná hodnota a nový stav zdrojovej tabuľky sa preniesie na štatistickú tabuľku. Pridaním nového riadka do štatistickej tabuľky v nej pribudne hodnota 0 s početnosťou 0. Tento údaj môže používateľ upraviť na inú hodnotu a početnosť. Pridaním nového riadka dôjde k zmene údajov v tabuľke a to spôsobí spustenie metód na výpočet štatistických popisných charakteristík, čo bude mať okrem iného za následok aj to, že nový riadok sa v štatistickej tabuľke zobrazí na svojom vzostupne usporiadanom mieste. Ak už v tabuľke existuje riadok s hodnotou 0 a ľubovoľnou početnosťou, pridaním nového riadka sa zdanlivo nič nezmení a to pre to, že znova nastane prepočítanie štatistického súboru a jedným z krokov je odstránenie duplikátov. A keďže už existuje riadok tabuľky s hodnotou 0, pridaním nového riadka s hodnotou 0 a početnosťou 0 dôjde k ich zjednoteniu a početnosť sa nezmení. Tlačidlom - sa odstráni prvý riadok v stĺpci. Pridávanie a odstraňovanie riadkov s hodnotami takýmto spôsobom nie je moc praktické. Pri ďalšom vývoji programu by sme ho mohli nastaviť tak, aby napríklad vedel pridať viacero riadkov súčasne a aby pri pridávaní nedochádzalo k usporadúvaniu hodnôt počas ich vytvárania, aby používateľ nebol zaskočený náhlou zmenou poradia riadkov. Obdobne by bolo vhodné nastaviť program tak, aby odstránil vybraný riadok namiesto posledného, čo by aj používateľovi dávalo väčší zmysel.

Pre zmenu údajov v niektorej z tabuliek môže používateľ jednoducho kliknúť na zvolené číslo, či už hodnotu alebo početnosť, ak mení údaje v štatistickej tabuľke a pomocou klávesnice napísať ním požadovanú hodnotu. Pre jeho pohodlie je však k dispozícii aj možnosť preklikať sa k požadovanej hodnote pomocou šípok, ktoré sa zobrazia pri každom čísle v tabuľke po jeho zakliknutí. Toto je štandardný komponent štruktúry Flex, ktorý sa nazýva Numeric Stepper. Pre jeho správne použitie sme museli v programe definovať maximálnu a minimálnu hodnotu, ktorú môže tento Numeric Stepper zobrazovať. Maximálna hodnota bola nastavená na maximálnu hodnotu 32 bitovej číselnej premennej integer, čo zodpovedá číslu 2'147'483'647 a minimálna hodnota bola obdobne nastavená na minimálnu hodnotu 32-bitovej premennej integer, čo zodpovedá číslu -2'147'483'648. Pri dosiahnutí krajnej hodnoty už používateľ nebude môcť v programe kliknutím na šípky prekročiť krajné hodnoty a manuálnym prepísaním čísla na vyššiu hodnotu program automaticky zmení zapísanú hodnotu na krajnú povolenú hodnotu. Pre demonštračné účely programu AirStat ako aj pri bežnom použití

predpokladáme, že použité hodnoty neprekročia tieto hranice, nakoľko ich tabuľka jednoducho nebude schopná kvôli obmedzeniam programu zobrazit’.

4.2.2 Výsledkové údaje

Popisné charakteristiky štatistického súboru sa nachádzajú v pravej hornej časti scény programu. Po spustení programu sú nevyplnené, no vytvorením alebo vložením tabuľky program tieto údaje vyplní číselnými hodnotami získanými z výsledkov metód popisných charakteristík. V súčasnom stave vývoja programu si môže používateľ ukázaním na konkrétnu popisnú charakteristiku zobrazit’ jej popis, čím sa zobrazí vyskakovacie okno tooltip, kde je stručne vysvetlené čo daná charakteristika predstavuje.

Stĺpcový graf v pravom dolnom rohu scény programu je vstavaný komponent Flexu a pre program AirStat nebol nijako upravovaný. Graf má všeobecné číselné hodnoty na svojej vodorovnej a zvislej osi a ukázaním kurzora myši na jednotlivé stĺpce v grafe si používateľ môže zobrazit’ konkrétne číselné hodnoty daného stĺpca. Graf automaticky prispôsobuje svoju veľkosť aj šírku jednotlivých stĺpcov rozsahu údajov zo štatistickej tabuľky a ich množstvu a poskytnutému priestoru na scéne.

4.2.3 Ostatné údaje

Všetky texty v programe AirStat sú napísané v angličtine pre jednoduché zabezpečenie jazykovej univerzálnosti programu. V súčasnom stave vývoja programu sú všetky textové popisy vložené priamo v zdrojovom kóde v triedach typu View, čo nie je úplne v súlade s myšlienkou oddelenia zdrojového kódu do štruktúry MVC. No tieto texty môžeme jednoducho zobrať z miest kde sú v programe použité a preniesť ich do jazykových súborov. V týchto jazykových súboroch je možné uložiť aj preklady do iných jazykov. AIR aplikácia si po spustení vie zistiť jazyk operačného systému a vie tak vybrať vhodný jazykový balík. Jeden program by teda mohol svoj zobrazovaný jazyk prispôbiť jazyku operačného systému. Ak by preklad do niektorého jazyka nebol úplný, program

použije jazykové súbory z východzieho jazyka, čo je angličtina. Obdobne môžeme nastaviť program AirStat tak, aby používateľovi ponúkol možnosť vybrať si ním preferovaný jazyk.

Nakoľko je program AirStat ešte stále vo vývoji, môže v ňom počas používania nastať niekoľko nepredvídateľných správaní, ktoré je potrebné pred samotným vypustením programu do bežného používania ošetriť. Atypické hodnoty v popisných charakteristikách môžu napríklad spôsobiť zväčšenie tabuľky s popisnými charakteristikami natoľko, že táto vytlačí stĺpcový graf a spodný okraj tabuliek mimo hranice scény.

Záver

Z prezentácie funkčnosti programu AirStat vidíme, že javí všetky známky funkčnosti spoľahlivého programu na vykonávanie jednoduchých aj zložitejších štatistických výpočtov. Program ako taký nie je ani zďaleka vo svojej finálnej verzii a pripravený pre koncového používateľa pre plnohodnotné spracovanie štatistických údajov, ale poskytuje solídny základ pre jeho ďalší vývoj a zdokonaľovanie. Jeho knižnice, ktoré sú z programového hľadiska plne funkčné a schopné spracovávať poskytnuté štatistické údaje je možné jednoducho zobrať a v prípade záujmu použiť pri vývoji iného programu na štatistické výpočty.

Vzhľadom na to, že na trhu sú už k dispozícii rôzne softvérové riešenia na spracovanie štatistických údajov, rozhodli sme sa ponechať zdrojový kód prístupný verejnosti ako open-source. Nie je naším cieľom konkurovať existujúcim softvérovým riešeniam, ale obohatiť už existujúce riešenia, prípadne navrhnúť nové, keď napríklad použité metódy v našom zdrojovom kóde vnuknú niekomu nápad na zlepšenie iného programu, alebo vo svojom programe rovno použije celé naše knižnice.

Vývoj sofistikovaného softvéru je náročná záležitosť na prácu aj na čas. Problémy, na ktoré sme počas vývoja narazili nám predĺžili čas vývoja a oddialili termín vyhotovenia programu. Niektoré funkčnosti programu AirStat boli implementované narýchlo a nerešpektujú celkom architektúru MVC, ako napríklad jazykové súbory, ktoré sú priamo súčasťou komponentov View a niektoré iné funkčnosti programu sme nestihli implementovať vôbec, ako napríklad štvorcový graf box plot, ktorý bol súčasťou plánu. Sofistikované programy vyvíjajú celé tímy ľudí, kde každý má zadelené svoje úlohy a ich práca je koordinovaná a trvá niekoľko mesiacov až rokov. Nami vyvíjaný projekt mal jedného vývojára, ktorý vypracoval návrh, napísal a testoval program a napísal k nemu dokumentáciu. Vzhľadom na to je výsledok práce adekvátny dostupným zdrojom.

Prvým významnejším problémom ešte v úvode programu bol data mining dát z Excelu, kedy bolo treba vybrať z dodanej HTML štruktúry relevantné dáta. Tento problém sme vyriešili prehľadávaním poskytnutej HTML štruktúry a hľadaním príslušných značiek označujúcich riadky a stĺpce v tabuľke. Znaky, ktoré nasledovali za týmito značkami boli údaje tabuľky, ktoré sme hľadali.

Ďalší zaujímavý problém bolo vytvorenie univerzálnej metódy na nájdenie ľubovoľného kvantilu z daného súboru hodnôt. Podarilo sa nám vytvoriť univerzálnu metódu postačujúcu pre naše výpočty, ktorá je uvedená v časti Výpočty popisných charakteristík.

Jeden z najzaujímavejších doplnkov, ktorý v programe AirStat chýba a ktorý bol stanovený ako jeden z cieľov vývoja programu je štvorcový graf box plot. Flex nie je primárne tvorený na štatistické výpočty a tak nemá vstavaný komponent na zobrazenie box plotu, ako má napríklad štandardné a stĺpcové, či riadkové grafy. Naším plánom bolo teda vytvoriť vlastný vizuálny komponent, ktorý by po prijatí dát cez data binding vedel vykresliť na scéne box plot so všetkými potrebnými informáciami. Bohužiaľ časová tieseň nám nedovolila vývoj tohto komponentu uskutočniť a tak v programe nie je implementovaný. V budúcom vývoji však nie je problém tento komponent jednoducho doplniť.

V ďalšom vývoji programu v budúcnosti by sme sa mali v prvom rade sústrediť na opravu existujúcich chýb, ktoré program obsahuje, ako napríklad v súčasnosti najzávažnejší problém a to posunutie stĺpcového grafu mimo scény pri atypických hodnotách vstupných údajov. Neskôr môžeme do programu pridať nové doplnky, ako napríklad možnosť importovať celé súbory typu Excel, bez potreby otvoriť program Excel a manuálne kopírovať jednotlivé tabuľky do programu AirStat. Ďalej by sme mohli doplniť do programu spomínaný box plot. Zaujímavým doplnkom by mohla byť možnosť ukladať rozrobenú prácu do súboru a následne ju zo súboru načítať a pokračovať v práci ďalej. Koncového používateľa by isto potešila možnosť vytlačiť svoje údaje na tlačiarňu, alebo do súboru PDF. Program vytvorený pre platformu AIR je tohto všetkého schopná. Určite by bolo dobré, ak by program AirStat vedel pracovať nie len s prierezovými ale aj s časovými údajmi. Mohol by mať schopnosť vykresliť časové údaje v podobe grafu na scéne a vedel by predvídať vývoj poskytnutých údajov v budúcnosti.

Komentáre ASDoc použité v našom programe pri popise funkčnosti rôznych metód a premenných značne uľahčia ďalší vývoj, keď sa k zdrojovým súborom vrátime po dlhšom čase, alebo keď zdrojové súbory po nás preberie niekto iný. Súčasťou štruktúry Flex je aj jeden šikovný program s rovnomenným názvom ASDoc.bat, ktorý pri správnom nastavení dokáže prehľadať všetky zdrojové súbory nášho programu a zo všetkých použitých metód a premenných zo všetkých Tried dokáže vytvoriť komplexnú

dokumentáciu vo forme HTML súborov, ktoré následne uloží na nami zvolené miesto na disku. Túto dokumentáciu potom môžeme použiť ako nezávislý zdroj informácií o použitých metódach a zverejniť ho napríklad na webstránke ako oficiálnu dokumentáciu k zdrojovému kódu programu AirStat. Spoločnosť Adobe tiež používa presne takúto štruktúru HTML stránok na poskytnutie oficiálnej dokumentácie k svojim metódam a premenným, ktoré sú už vstavané v štruktúre Flex a v jazyku ActionScript na svojich webstránkach.

Toto všetko si však vyžaduje ďalší čas a prácu investovanú do vývoja, čo nebolo možné stihnúť v rámci poskytnutého priestoru pre vypracovanie projektu. Pokračovať vo vývoji programu v budúcnosti môže mať prínos pre nás ako aj pre ostatných potenciálnych vývojárov, ktorí by sa na projekte mohli spolupodieľať, čím by získali neoceniteľné skúsenosti pri manipulácii, uchovávaní a spracovaní štatistických údajov v tomto alebo iných projektoch.

Zoznam použitej literatúry

- [1] *What is R?* [online]. The R Project for Statistical Computing [cit. 2014.03.14].
Dostupné na internete: <<http://www.r-project.org>>.
- [2] *Gnu Regression, Econometrics and Time-series Library* [online]. Free Software Foundation, 2013.11.21 [cit. 2014.03.14]. Dostupné na internete:
<<http://gretl.sourceforge.net/>>.
- [3] PACÁKOVÁ, V. a kol. 2003. *Štatistika pre ekonómov*. Bratislava : IURA EDITION, 2003. 358 s. ISBN 80-89047-74-2.
- [4] WALDRON R. 2000. The Flash history [online]. flashmagazine.com, 2000.11.20 [cit. 2014.03.15]. Dostupné na internete:
<http://www.flashmagazine.com/news/detail/the_flash_history/>.
- [5] *Internet banking Tatra banky bude vo Flashi* [online]. DSL.sk, 2010.09.11. [cit. 2014.03.15]. Dostupné na internete: <<http://www.dsl.sk/article.php?article=9711>>.
- [6] *About Apache Flex* [online]. Apache [cit. 2014.03.18]. Dostupné na internete:
<<http://flex.apache.org/about-what-is.html>>.
- [7] FEIFAN Z. 2011. Design Patterns: Model-View-Controller [online]. 2011.08.13 [cit. 2014.04.16]. Dostupné na internete:
<<http://cupsofcocoa.com/2011/08/13/design-patterns-model-view-controller/>>.