

**ŽILINSKÁ UNIVERZITA V ŽILINE**  
**FAKULTA RIADENIA A INFORMATIKY**

**ALGORITMY A ÚDAJOVÉ ŠTRUKTÚRY 1**

Semestrálna práca č. 2

Juraj Oberta

5ZY02A

2019

## Obsah

1. Návrh aplikácie.....	3
2. Popis implementácie, výpočtové zložitosti.....	5
3. Používateľská príručka .....	6
4. UML diagram programu (UML v plnej kvalite v súbore dokumentácie) .....	7

## 1. Návrh aplikácie

Okrem použitých údajových štruktúr ďalej moja aplikácia obsahuje triedy:

**Filter** – trieda obsahujúca „Filter“ a jeho potomkov „Filter\_fi“, „Filter\_Fi“ a „Filter\_Nazov“. Všetky filtre preberajú objekt a kritérium.

- Filter\_fi – pomocou metódy ohodnot' zistí či sa výsledok kritéria rovná zadanej hodnote alfa.
- Filter\_Fi – pomocou metódy ohodnot' zistí či sa výsledok kritéria nachádza v rozmedzí <Alfa, Beta>.
- Filter\_Názov – pomocou metódy ohodnot' zistí či sa kritérium názov obce/okresu/kraju podobá na zadaný string (stačí aj 1 rovnaký charakter)

**Kriterium** – trieda obsahujúca „Kriterium“ a jeho potomkov. Celá trieda je implementovaná pomocou generík takže môže spracovať všetky objekty ktoré majú v sebe dané metódy.

- KriteriumNazov – vráti hodnotu metódy GetNazov v objekte.
- KriteriumPrislusnost – vráti hodnotu metódy PatriDoVyssiehoCelku, preberá pointer na vyšší celok.
- KriteriumVolici – vráti počet voličov v obci/okrese/kraji, preberá hodnotu kolo.
- KriteriumVydaneObalky - vráti počet vydaných obálok jednotky podľa kola.
- KriteriumOdovzdaneObalky – vráti počet odovzdaných obálok jednotky podľa kola.
- KriteriumHlasyOdovzdane – vráti počet odovzdaných hlasov v danom kole.
- KriteriumUcast – vráti percento účasti podľa kola.

**Manazer** – Ovláda základnú funkcionality programu, filtrovanie obcí, okresov a krajov a taktiež načítava dáta z .csv súborov uložených v „/res/1Kolo“ a „/res/2Kolo“.

- Sorted Sequence Table – do SST sú ukladané všetky obce, okresy a kraje. Ako kľúč je použitý kód danej obce alebo územnej jednotky. Ako dáta obsahuje pointre na objekty územných jednotiek. Pre okresy a kraje je vytvorený ešte jeden SST ktorý obsahuje ako kľúč názvy týchto jednotiek (sú unikátne) a ako dáta rovnaké pointre ako hlavné SST.
- ArrayList – Do triedy MainForm ktorá sa stará o GUI sú posielané už vyfiltrované Listy s dátami.

**MainForm** – Trieda ktorá sa stará o GUI programu, prevažne využíva C# WinForm a jeho Garbage Collector (iba na GUI, štruktúry s dátami na výpis sú vytvorené zvlášť na Heap-e bez automatickej správy pamäte).

- ListView – štruktúra slúžiaca na výpis utriedených a filtrovaných dát do WinForms.
- ListViewItem – štruktúra ktorá obsahuje jeden riadok na výpis. Preberá dáta pomocou kritérií z ArrayList-u.
- ArrayList – slúžia na uloženie vyfiltrovaných dát z Manažéra, môže ešte prejsť cez quick sort algoritmus ktorý daný list utriedi podľa kritéria zadaného používateľom.

**QuickSort** – Utriedi ľubovoľného potomka štruktúry List podľa kritéria zadaného používateľom.

**Oblast** – Obsahuje informácie o územných jednotkách rozdelené podľa kola.

- Array – Array o veľkosti 2, ktorý drží integery. Na indexe 0 sú údaje o prvom kole, na indexe 1 sú údaje o druhom kole.

**Obec** – Obsahuje informácie o výsledkoch rozdelené podľa kola.

- Array – Array o veľkosti 2, ktorý drží integery. Na indexe 0 sú údaje o prvom kole, na indexe 1 sú údaje o druhom kole.

**ObertaJurajAUS2** – táto trieda obsahuje metódu main, vytvára inštancie manažéra a MainForm ktorej potom manažéra pomocou pointeru pošle.

## **2. Popis implementácie, výpočtové zložitosti**

**1. Vypísanie informácií o obciach,** pri jednom prechode zoznamom všetkých obcí  $O(n)$  skontroluje či sú zamietnuté alebo povolené podľa filtrov Názov, Voliči alebo Účasť. Ak sú povolené, priradí obec do nového ArrayListu  $O(1)$  a odošle tento list grafickému rozhraniu.

**Zložitosť:**  $O(n) + O(1)$

**2. Zoradenie obcí,** obce sú zoradené pomocou algoritmu quick sort podľa kritéria zadaného používateľom  $O(n \cdot \log n)$  a následne sú vypísane do ListView  $O(n)$ .

**Zložitosť:**  $O(n \cdot \log n) + O(n)$

**3. Vypísanie informácií o územných jednotkách,** rovnako ako pri obciach, prechádza SST všetkými územnými jednotkami a v prípade že jednotka spĺňa filtre Názov, Voliči alebo Účasť je priradená do nového ArrayListu  $O(1)$  ktorý je potom odoslaný grafickému rozhraniu na výpis  $O(n)$ .

**Zložitosť:**  $O(n) + O(1) + O(n)$

**4. Zoradenie územných jednotiek splňujúci daný filter,** podľa názvu sa vyhladá v SST pointer územnej jednotky  $O(\log n)$  a následne sa filtrujú obce podľa filtra Účasť a Príslušnosť obce  $O(n)$ . V prípade že jednotka splnila filtre, je pridaná do ArrayListu vyhovujúcich obcí  $O(1)$  a odoslaná grafickému rozhraniu. Následne sú zoradené podľa užívateľom zadaných kritérií  $O(n \cdot \log n)$ .

**Zložitosť:**  $O(\log n) + O(n) + O(1) + (n \cdot \log n)$

### 3. Používateľská príručka

Na ovládanie aplikácie sa využíva grafické užívateľské rozhranie ktoré vypadá nasledovne:

Názov	Názov Okresu	Názov Kraju	1. Počet Voličov	2. Počet Voličov	Spolu	1. Účasť	2. Účasť	1. Vydané obálky	2. Vydané obálky	1. Odovzd. obálky	2. Odovzd. obálky	1. Hlasy	2. Hlasy	Filter
Tvrdosín	Tvrdosín	Žilinský kraj	7 160	7 109	14 269	56%	45,49%	4 010	3 234	4 006	3 234	3 982	3 139	7 160
Liptovský Hrádok	Liptovský Hrádok	Žilinský kraj	6 206	6 138	12 344	55,25%	46,41%	3 429	2 849	3 428	2 847	3 401	2 788	6 206
Turzovka	Čadca	Žilinský kraj	6 160	6 133	12 293	45,73%	38,05%	2 817	2 334	2 817	2 334	2 792	2 283	6 160
Vrátky	Martin	Žilinský kraj	6 152	6 165	12 317	54,16%	46,06%	3 332	2 840	3 332	2 840	3 301	2 772	6 152
Námestovo	Námestovo	Žilinský kraj	6 078	6 102	12 180	54,5%	44,32%	3 313	2 705	3 313	2 705	3 293	2 616	6 078
Trstená	Tvrdosín	Žilinský kraj	5 733	5 683	11 416	56,07%	44,94%	3 215	2 554	3 214	2 553	3 180	2 450	5 733
Turčianske Teplice	Turčianske Teplice	Žilinský kraj	5 583	5 683	11 266	54,88%	48%	3 064	2 728	3 064	2 727	3 049	2 672	5 583
Kráľovo nad Kysou	Čadca	Žilinský kraj	5 390	5 396	10 786	49,59%	42,73%	2 673	2 306	2 672	2 306	2 658	2 250	5 390
Rajec	Čadca	Žilinský kraj	4 902	4 891	9 793	54,46%	45,18%	2 670	2 210	2 670	2 210	2 666	2 144	4 902
Oščadnica	Čadca	Žilinský kraj	4 666	4 641	9 307	46,14%	37,9%	2 153	1 759	2 152	1 759	2 130	1 735	4 666
Raková	Čadca	Žilinský kraj	4 453	4 441	8 894	48,57%	42,53%	2 163	1 889	2 163	1 888	2 157	1 859	4 453
Skalité	Čadca	Žilinský kraj	4 134	4 167	8 301	48,5%	39,88%	2 005	1 662	2 004	1 662	1 986	1 611	4 134

Program sa ovláda zadávaním hodnôt alebo rozmedzia pre Filtre a výberom územných jednotiek ktoré chce používateľ prehľadávať. Napríklad (ako na obrázku vyššie) výpis všetkých obcí ktoré majú v názve „a“, počet voličov v prvom kole obce bol v rozmedzí 4000 až 8000 ľudí a účasť obyvateľov v prvom kole bola medzi 40% až 70%. Obce sú zoradené zostupne podľa počtu voličov v prvom kole a ich nadradený územný celok je Žilinský kraj.

#### Filtre –

- Názov – kontroluje či územná jednotka obsahuje vo svojom názve daný charakter alebo slovo.
- Počet voličov – Kontroluje rozsah od, do počtu voličov v územnej jednotke.
- Účasť voličov – Kontroluje rozsah od, do účasti voličov v územnej jednotke. Účasť môže byť 0% až 100%.
- Kolo – výber podľa ktorého kola chce používateľ filtrovať. V prípade že chce používateľ filtrovať napr. podľa súčtu počtu voličov, je potrebné zaškrtnúť obe kolá.

#### Zoradenie –

- Kritérium – používateľ si môže vybrať kritérium podľa ktorého je možné jednotky zoradiť. Na výber je zoradenie podľa názvu, počtu voličov a účasti voličov v percentách.
- Zostupne – v prípade že chcete vypísať výsledok zoradenia zostupne, zaškrtnite toto políčko.

#### Výpis –

- Územná jednotka – dáva na výber aký typ územnej jednotky chceme vypisovať. Teda sú na výber obce, okresy alebo kraje Slovenskej Republiky.

#### Príslušnosť obce –

- Pole príslušnosť obce – používateľ môže zadať ľubovoľnú vyššiu územnú jednotku a do výpisu sa budú brať len obce ktoré sa nachádzajú v tejto vyššej územnej jednotke.

#### 4. UML diagram programu

