

Fakulta riadenia a informatiky  
Informatika



SEMESTRÁLNA PRÁCA II. – ALGORITMY A ÚDAJOVÉ ŠTRUKTÚRY I.

Výsledky volieb

## Návrh aplikácie

Aplikáciu som rozvrhol do nasledujúcich tried: App, Area (zahŕňa všetkých potomkov triedy Filter (zahŕňa všetky potrebné triedy špecifických filtrov ako potomkov), Kriterium (zahŕňa všetky potrebné triedy špecifických kritérií ako potomkov), Nacitanie, VolebneKolo., MyForm.

**Trieda App** zabezpečuje chod aplikácie, mienené tak, že aplikácia by sa dá ľahko prerobiť tak, že bude fungovať aj s iným GUI prostredím, prípadne len ako consolová aplikácia.

**Trieda Area** má potomkov triedy Kraj, Okres, Obec. Okres, ktoré vedia vrátiť typ ako string, kvôli vypisovaniu do GUI. Trieda Okres si ešte navyše oproti Arei ukladá názov kraja v ktorom sa nachádza, trieda Obec si ešte navyše oproti Arei ukladá názov kraja a názov okresu v ktorom sa nachádza. Každá Area taktiež obsahuje pole výsledkov volebných kôl. Na 0-tej pozícii sa nachádzajú sumárne výsledky, na 1. výsledky prvého a na 2. výsledky druhého kola.

**Trieda Filter** obsahuje potomka Filter\_FI a Filter\_fi, kedy Filter\_FI má definovaný rozsah, zatiaľ čo Filter\_fi len porovnáva hodnoty. Taktiež sa tu nachádzajú konkrétne filtre, ktoré boli v zadaní, konkrétne FilterHasName, FilterVolici, FilterUcast, FilterPrislusnostObce.

**Trieda Kriterium** obsahuje všetky potrebné kritéria, tak ako boli opísané v zadaní napríklad. KriteriumNazov, KriteriumPrislusnostObce...

**Trieda Nacitanie** sa stará o vytváranie štruktúr, načítanie z csv súboru a zaradenie do správnych štruktúr. Taktiež zabezpečuje prepočítavanie sumárnych štatistík za obidve kolá.

**Trieda VolebneKolo** obsahuje všetky dáta z volebného kola, ako napríklad počet voličov, účasť voličov a pod.

**Trieda MyForm** zastupujúca GUI rozhranie.

## Popis implemetácie údajových štruktúr

Všetky hlavné štruktúry sa vytvárajú už pri načítavaní dát. Používam binárny vyhľadávací strom( BinarySearchTree), kvôli jeho rýchlosti načítania a taktiež ideálnej zložitosti pristupovania prvkov pre moje potreby.

Ideálny prípad Vlož prvok ->  $O(\log N)$  Zruš prvok ->  $O(\log N)$  Nájdi prvok ->  $O(\log N)$

Najhorší prípad Vlož prvok ->  $O(N)$  Zruš prvok ->  $O(N)$  Nájdi prvok ->  $O(N)$

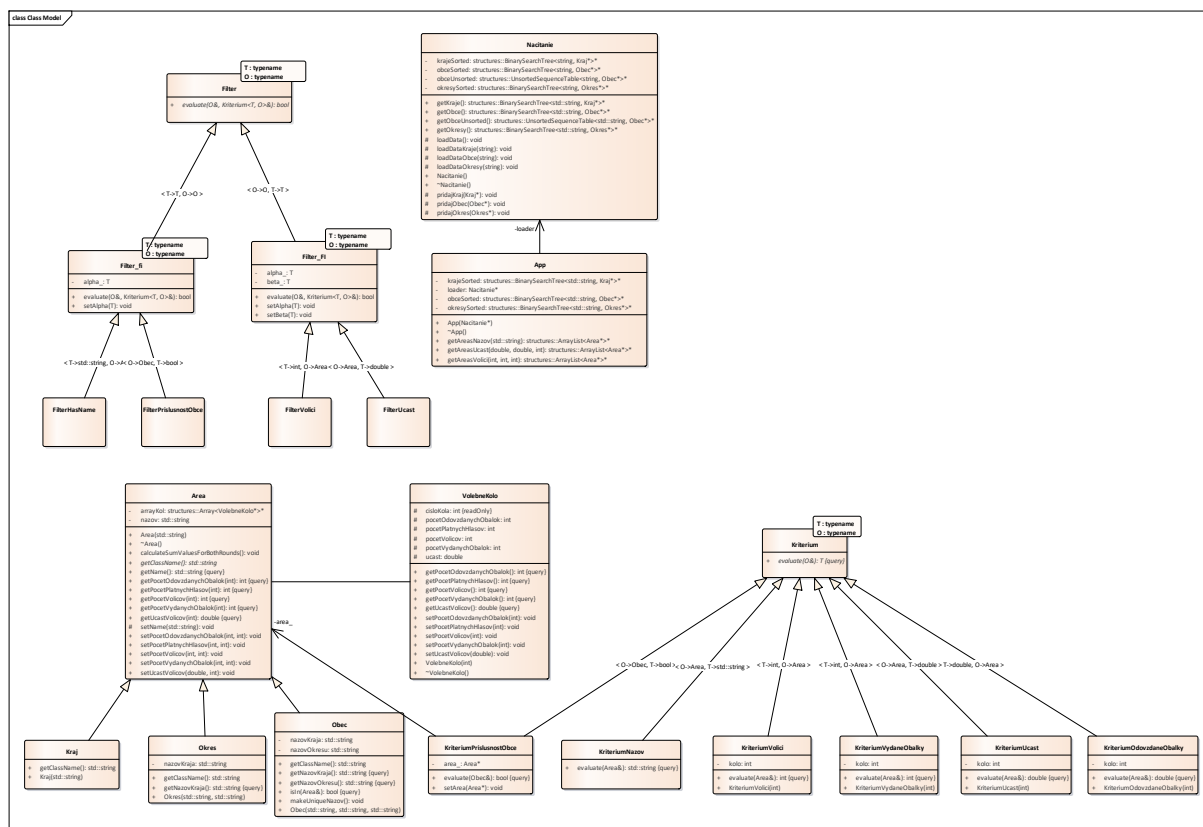
Taktiež si vytváram jednu neusporiadanú tabuľku (UnsortedSequenceTable) do ktorej si ukladám obce, so zložitostou  $O(1)$  a následne ich viem potom aj sortovať, vhodným algoritmom.

V aplikácii si potom vytváram pomocné ArrayListy 3 -> ktoré posielam go GUI, na to aby som ich vedel vypísať na obrazovku do rohrania.

## Zložitosti jednotlivých operácií

1. Vypíšte informácie o obciach podľa filtra názov -> vyhľadanie obce na základe jej vypísanie údajov o obci) má zložitosť  $O(\log N)$ , kde  $N$  je počet obcí.
2. Vypíšte informácie o obciach podľa filtra voliči -> má zložitosť  $O(n)$ , kde  $N$  je počet obcí, pretože je potrebné prejsť všetky prvky tabuľky, kvôli userovej voľbe.
3. Vypíšte informácie o obciach podľa filtra účasť -> má zložitosť  $O(n)$ , kde  $N$  je počet obcí, pretože je potrebné prejsť všetky prvky tabuľky, kvôli userovej voľbe.
4. Zoradíte vzostupne aj zostupne všetky obce podľa nasledujúcich názov ->  $O(N \cdot \log N)$
5. Zoradíte vzostupne aj zostupne všetky obce podľa nasledujúcich voliči ->  $O(N \cdot \log N)$
6. Zoradíte vzostupne aj zostupne všetky obce podľa nasledujúcich účasť ->  $O(N \cdot \log N)$
7. Vypíšte informácie o územných jednotkách<sub>1</sub> podľa filtrov názov ->  $N \cdot \log N$
8. Vypíšte informácie o územných jednotkách<sub>1</sub> podľa filtrov voliči ->  $O(n)$
9. Vypíšte informácie o územných jednotkách<sub>1</sub> podľa filtrov účasť ->  $O(n)$
10. Zoradíte vzostupne aj zostupne všetky obce<sub>2</sub>, ktoré spĺňajú filter  $F_{\text{Účasť}} \cap F_{\text{Prislušnosť Obce}}$ , podľa nasledujúcich kritérií názov ->  $O(N \cdot \log N) * O(N)$
11. Zoradíte vzostupne aj zostupne všetky obce<sub>2</sub>, ktoré spĺňajú filter  $F_{\text{Účasť}} \cap F_{\text{Prislušnosť Obce}}$ , podľa nasledujúcich kritérií voliči ->  $O(N \cdot \log N) * O(N)$
12. Zoradíte vzostupne aj zostupne všetky obce<sub>2</sub>, ktoré spĺňajú filter  $F_{\text{Účasť}} \cap F_{\text{Prislušnosť Obce}}$ , podľa nasledujúcich kritérií účasť ->  $O(N \cdot \log N) * O(N)$

## Diagram Tried



## Používateľská príručka

Program obsahuje jednoduché grafické užívateľské rozhranie.