Fakulta riadenia a informatiky

Informatika

SEMESTRÁLNA PRÁCA II. – ALGORITMY A ÚDAJOVÉ ŠTRUKTÚRY I.

# Výsledky volieb

Ing. Marek Kvet, PhD. – PIA 12:00

2018/2019 Dominik Bullo 5ZY029

## Návrh aplikácie

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

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

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 nacházda. 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 potrebne 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 rýhlosti 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 zložitosťou O(1) a následne ich viem potom aj sortovať, bhodným algoritmom.

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

## Zložitosti jednotlivých operácii

1. Vypíšte informácie o obciach podľa filtra názov -> vyhľadanie obce na základe jej vypísanie údajov o obci) má zložitosť O(logN), 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. Zoraďte vzostupne aj zostupne všetky obce podľa nasledujúcich názov-> keďže používam usporiadú tabuľku (respektívne binárny vyhľadávací strom) je už pri načítaní, pro zostupnom usporiadavaní je zložitosť O(n), taktiež výpis na GUI má zložitosť O(n).
5. Zoraďte vzostupne aj zostupne všetky obce podľa nasledujúcich voliči -> O(N\*logN)
6. Zoraďte vzostupne aj zostupne všetky obce podľa nasledujúcich účasť -> O(N\*logN)
7. Vypíšte informácie o územných jednotkách podľa filtrov názov-> N\*logN
8. Vypíšte informácie o územných jednotkách podľa filtrov voliči -> O(n)
9. Vypíšte informácie o územných jednotkách podľa filtrov účasť-> O(n)
10. Zoraďte vzostupne aj zostupne všetky obce, ktoré spĺňajú filter 𝐹Úč𝑎𝑠ť∩𝐹𝑃𝑟í𝑠𝑙𝑢š𝑛𝑜𝑠ť𝑂𝑏𝑐𝑒, podľa nasledujúcich kritérií názov-> O(N\*logN) + O(n) na výpis do GUI.
11. Zoraďte vzostupne aj zostupne všetky obce, ktoré spĺňajú filter 𝐹Úč𝑎𝑠ť∩𝐹𝑃𝑟í𝑠𝑙𝑢š𝑛𝑜𝑠ť𝑂𝑏𝑐𝑒, podľa nasledujúcich kritérií voliči-> O(N\*logN) + O(n) na výpis do GUI.
12. Zoraďte vzostupne aj zostupne všetky obce, ktoré spĺňajú filter 𝐹Úč𝑎𝑠ť∩𝐹𝑃𝑟í𝑠𝑙𝑢š𝑛𝑜𝑠ť𝑂𝑏𝑐𝑒, podľa nasledujúcich kritérií účasť -> O(N\*logN)+ O(n) na výpis do GUI.

## Diagram Tried



## Používateľská príručka

Program obsahuje jednoduché grafické užívateľské rozhranie.