



ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia a informatiky

Semestrálna práca 2

Algoritmy a údajové štruktúry 1 5UI124

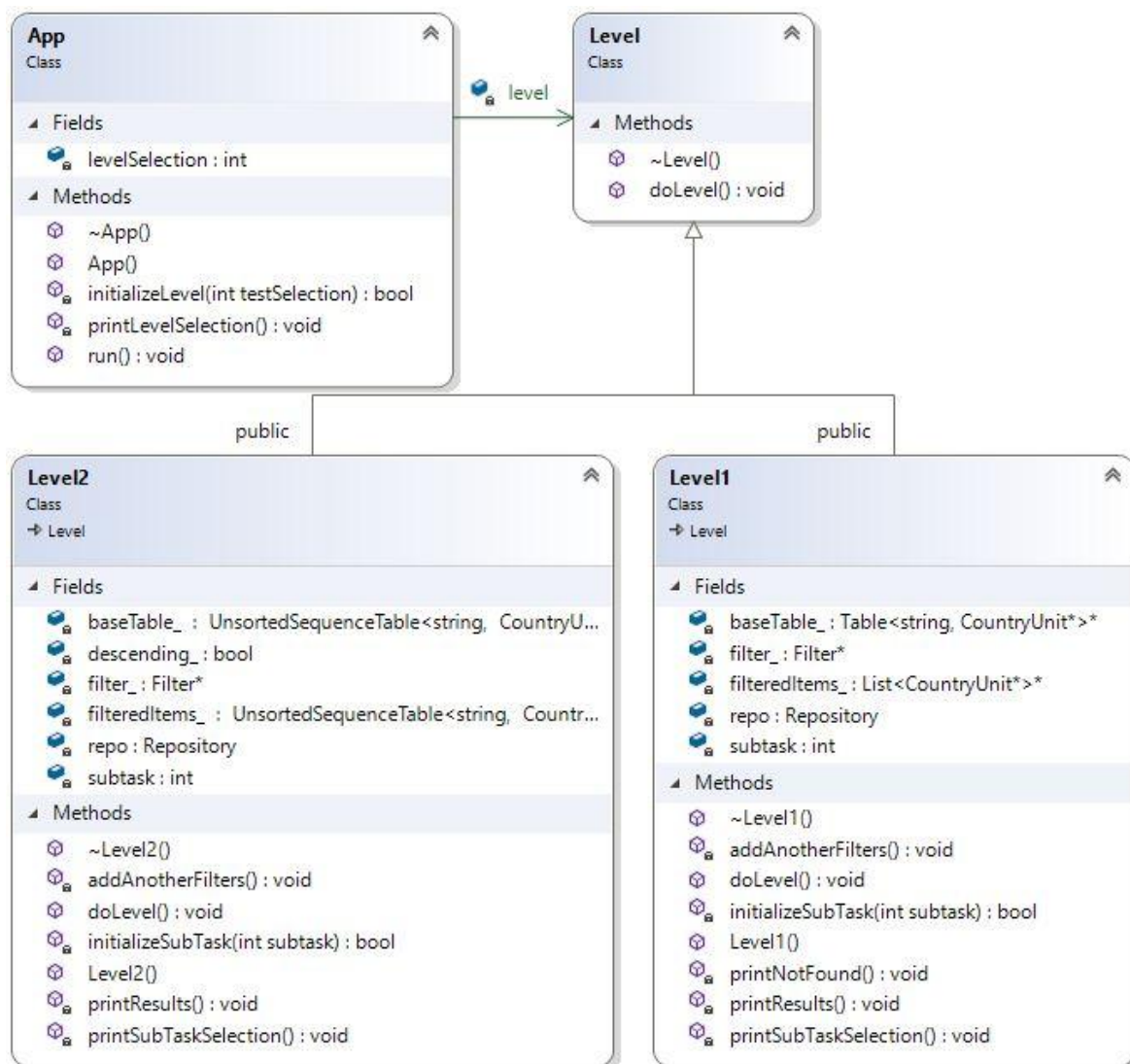
Aplikácia	3
Úrovne aplikácie	3
Kritériá	4
Filtre	5
Formátovač filtra	6
Územná jednotka	6
Triedenie	7
Ostatné triedy	7
Úroveň 1, 3	8
Načítanie dát	8
Podúloha a) Filter názov	8
Podúloha b) Filter počet obyvateľov	8
Podúloha c) Filter zastavanosť	9
Úroveň 2, 4	9
Načítanie dát	9
Podúloha a) Kritérium názov	9
Podúloha a) Kritérium počet obyvateľov	10
Podúloha a) Kritérium zastavanosť	10
Zložitosti jednotlivých filtrov	10
FilterName	10
FilterUpperCu	10
FilterCuType	10
FilterBuiltPercentage	10
FilterTotalCitizens	10
Používateľská príručka	11

Aplikácia

Táto semestrálna práca je tvorená konzolovou aplikáciou, ktorá umožňuje používateľovi vyhľadávanie a triedenie jednotlivých územných jednotiek na území Slovenskej republiky na základe výberových kritérií..

Úrovně aplikácie

Aplikácia pracuje s určitými úrovňami reprezentovanými triedou Level, ktoré sú prispôsobené buď na vyhľadávanie podľa rôznych kritérií alebo prispôsobené pre zoradenie daných územných jednotiek. Každá jednotlivá implementácia Level triedy obsahuje ukazovateľ na Filter, ktorý sa vytvára na základe používateľského vstupu. Taktiež obsahuje tabuľku, ktorá obsahuje prvotný zoznam územných jednotiek spolu s druhou štruktúrou v ktorej sú uložené územné jednotky vyhovujúce zadaným kritériám.



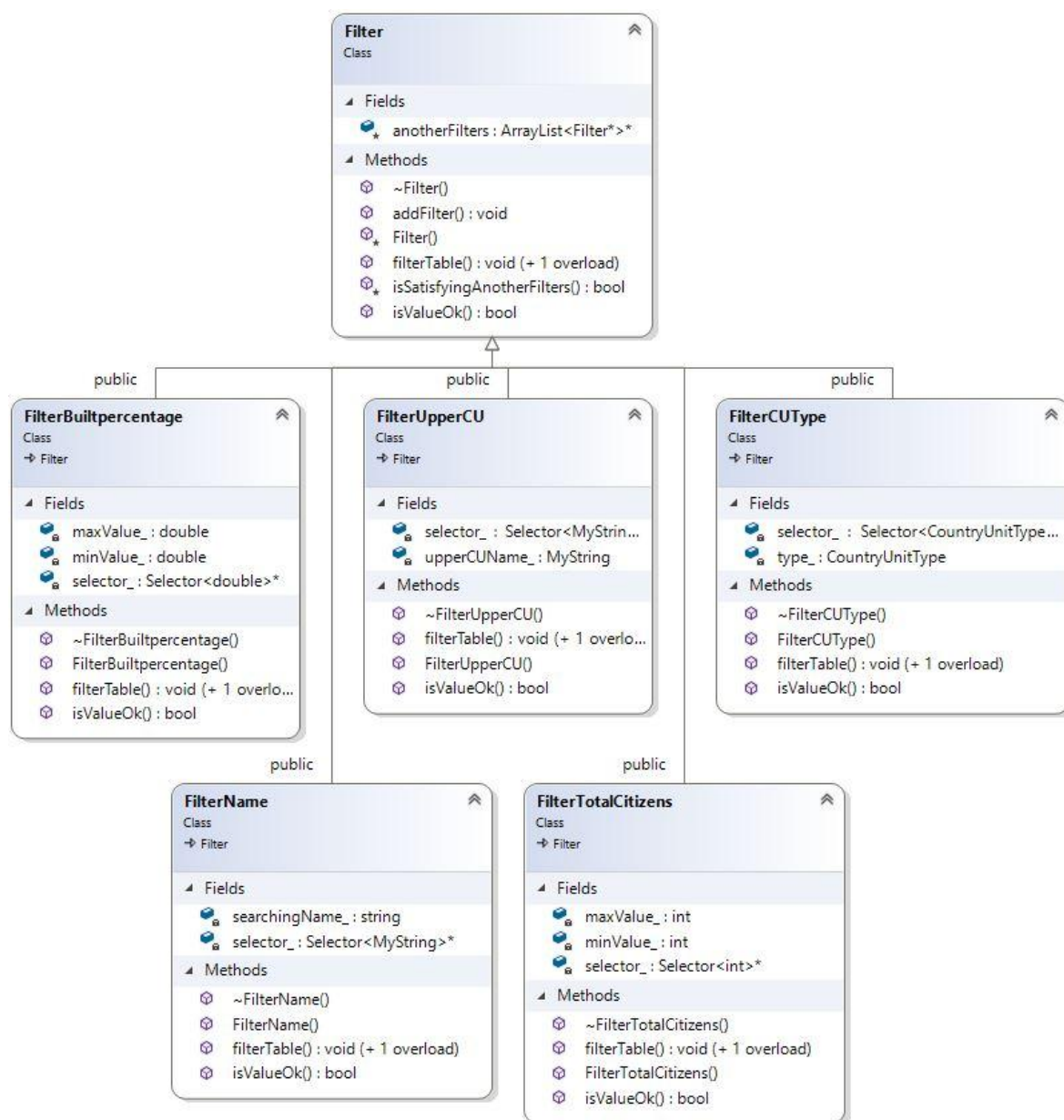
Kritériá

Pre umožnenie ucelenosti vyhľadávania alebo triedenia tieto algoritmy pracujú s kritériami reprezentovanými triedou `Selector<T>`. Táto trieda poskytuje rozhranie pre jednotlivé implementácie triedy a tiež poskytuje šablónu pre typ, ktorý metóda `getValue()` vracia. Do táto metóda berie ako parameter odkaz na `CountryUnit` triedu reprezentujúcu územnú jednotku. Pre danú implementáciu triedy `Selector` táto metóda vracia hodnotu územnej jednotky prislúchajúcu jednému zadanému kritériu.



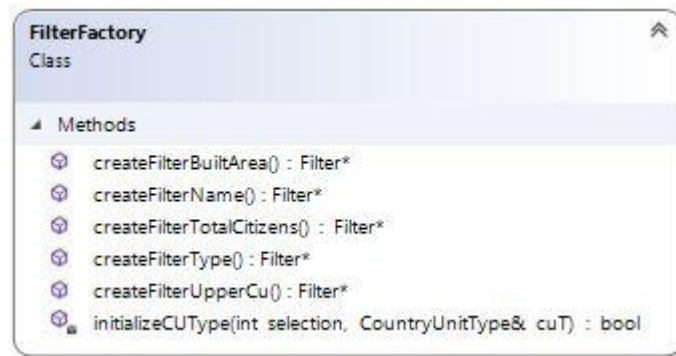
Filtre

Filtre umožňujú prehľadať danú množinu územných jednotiek a filtrovať ju na základe zadaných kritérií. Filter je v aplikácii reprezentovaný triedou `Filter`. Každý filter potrebuje vstupný parameter konštruktora, ktorý zadáva akú hodnotu má daný filter brať pri prechádzaní množinu UJ do úvahy a podľa nej vyberať vhodné UJ. Tento parameter konštruktora môže byť buď len jedna hodnota ako napríklad názov typu `MyString` alebo to môžu byť dve hodnoty predstavujúce maximálnu hranicu a minimálnu hranicu. Každý filter obsahuje tiež metódu pre pridanie ďalšieho filtra ku nemu samotnému. To má za následok, že pokiaľ je pridaný ďalší filter do triedy tak po nájdení príslušnej UJ filtrom sa navyše ešte skontroluje táto UJ pridanými filtermi či UJ vyhovuje.



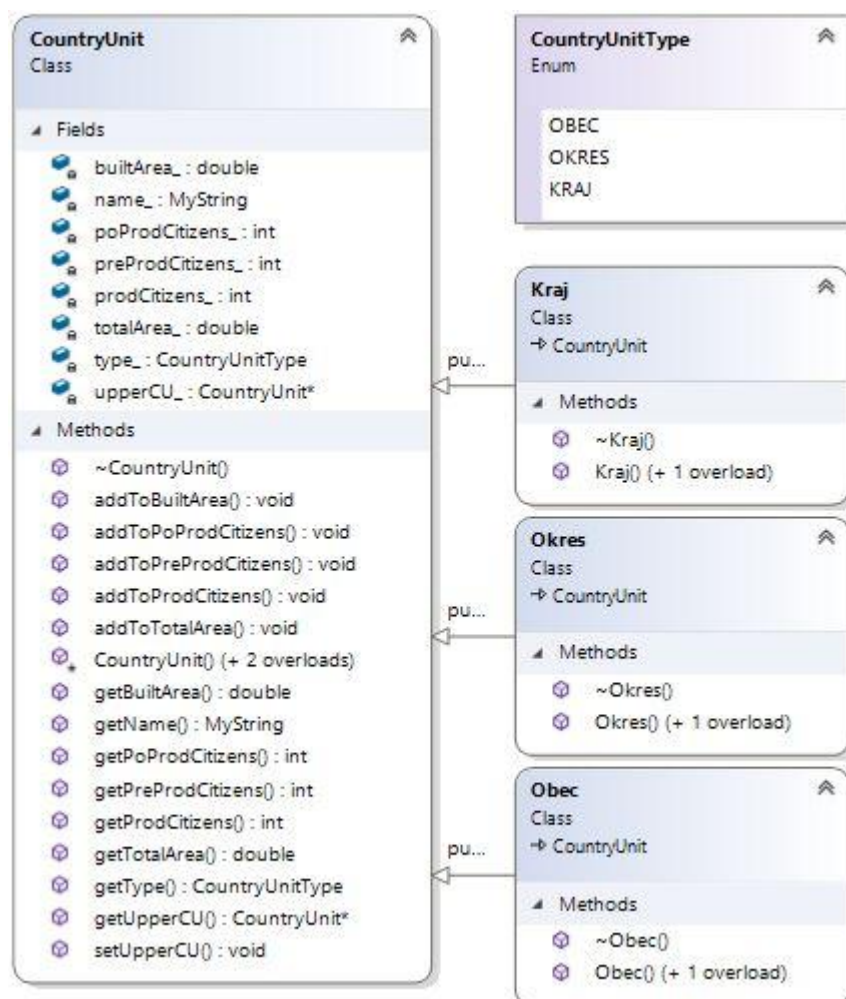
Formátovač filtra

Každý filter musí byť vytvorené špecificky pretože každý filter má iné atribúty filtrovania. Pre tento účel je vytvorená trieda `FilterFactory`, ktorá interagovaním s používateľom pomocou konzolového vstupu na obrazovku vytvorí objekt príslušného filtra a vráti ho ukazovateľom.



Územná jednotka

Aplikácia pracuje s územnými jednotkami, ktoré reprezentuje trieda CountryUnit. Jednotlivé implementácie tejto triedy predstavujú územné jednotky kraj, okres a samotnú obec. Rozlišovanie týchto tried je podľa atribútu typu územnej jednotky, ktorý predstavuje enum.



Tento typ UJ sa predáva ako parameter pri vytváraní základnej triedy CountryUnit. Každá implementácia triedy nastavuje svoj príslušný typ automaticky.

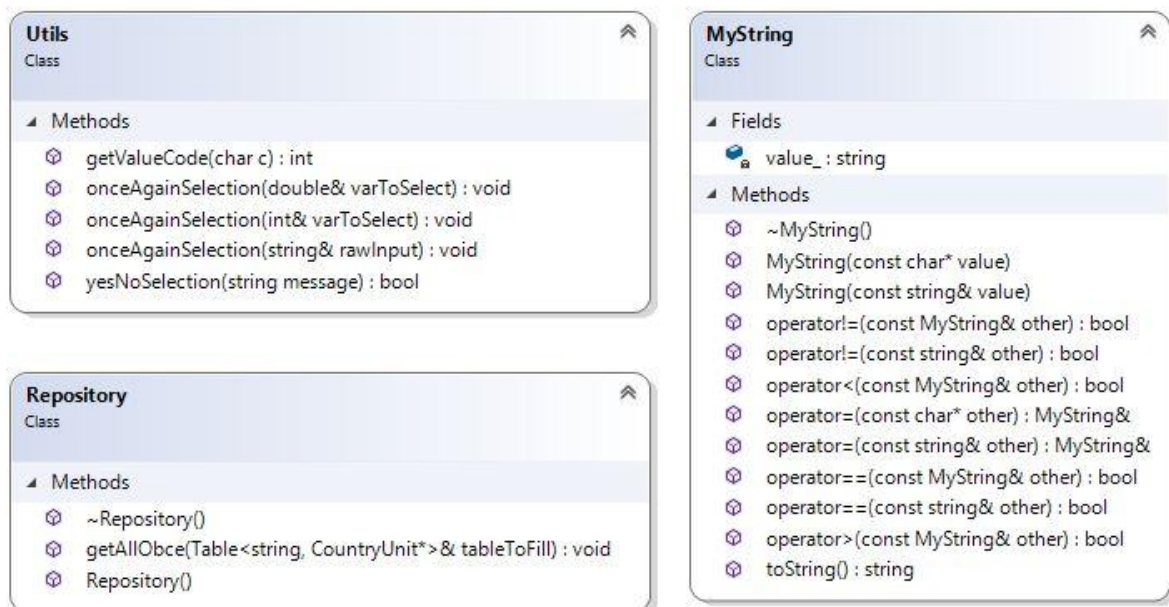
Triedenie

O triedenie územných jednotiek sa stará trieda Sort. Obsahuje jednu statickú metódu sort(), kde ako parameter je predávaná sekvenčná neutriedená tabuľka spolu s triedou pre daný Selector<T> a prepínačom pre zostupné triedenie.



Ostatné triedy

Aplikácia navyše využíva aj iné triedy, ktoré predstavujú tkz. pomocné triedy. Jedná sa o triedu pre pomocné funkcie Utils, triedu MyString, ktorá ma preťažené operátory pre prácu so stringom, ktorý obsahuje diakritiku a tiež Repository triedu pre načítanie údajov do štruktúr.



Úroveň 1, 3

Tieto úrovne aplikácie predstavujú vyhľadávanie medzi územnými jednotkami na základe výberových kritérií. Tretia úroveň sa od prvej líši len v tom, že používateľ pred spustením vyhľadávania zadá, že filtre pre typ a príslušnosť danej UJ sa majú preskočiť. Táto úroveň je v aplikácii reprezentovaná triedou Level1. V tejto úrovni je použitá pre prvotné uloženie dát o územných jednotkách **utriedená sekvenčná tabuľka**. Táto tabuľka je použitá pre jednoduchosť nasledujúceho vyhľadávania a tiež pri načítaní dát, ktoré sú už abecedne utriedené to nie je zložité. Pre uloženie územných jednotiek ako výsledok filtrovania je použitý arraylist. Pri konečnom výpise poskytuje jednotkovú zložitosť prístupnosti k dátam.

Načítanie dát

Po zvolení danej úrovne používateľom aplikácia nainicializuje príslušnú triedu Level1. Sekvenčne utriedená tabuľka je predaná ako parameter pri volaní metódy getAllObce() nad objektom Repository. Táto metóda načíta nielen obce, ale aj kraje a okresy do príslušnej tabuľky. Výsledná zložitosť načítania jednej obce pri tejto úrovni teda je $O(n)$, kde n je počet už uložených územných jednotiek v tabuľke. Súčasne spolu s načítaním dát o obci z CSV súboru Obce SR.csv sa načítava aj daná vyššie UJ, t.j. okres alebo kraj pre danú obec. Po načítaní názvu tejto UJ jednotky zo súboru Územné členenie SR.csv musí skontrolovať či už v zozname nie je. Tento zoznam je tvorený Arraylistom. Po pridaní všetkých obcí do tabuľky sa tieto dva arraylisty s kraji a okresmi tiež pridávajú do tabuľky. Výsledná zložitosť načítania kraja alebo okresu je teda $O(n + k)$, kde n je počet všetkých územných jednotiek v tabuľke a k je počet uložených okresov (v prípade načítania kraja krajov) v arrayliste.

Podúloha a) Filter názov

V tejto podúlohe používateľ je schopný vyhľadať danú územnú jednotku zadáním príslušného názvu UJ. Používateľ môže zapnúť a nastaviť aj filtre pre filtrovanie len určitého typu UJ alebo tiež filtrovanie UJ len pre danú príslušnosť do vyššej UJ. Nakoniec je vytvorený filter FilterName, kde sa prípadne nastaví aj ďalšie filtre ako FilterCuType alebo FilterUpperCu, ktoré pracujú s príslušnými triedami Selector. V tejto podúlohe je výsledná zložitosť operácie $O(\log(n) + k)$. n reprezentuje počet územných jednotiek v tabuľke zatiaľ čo k reprezentuje počet vyfiltrovaných UJ pri výpise kde zložitosť je $k \cdot O(1)$.

Podúloha b) Filter počet obyvateľov

Podobne ako pri predchádzajúcej podúlohe si používateľ môže nastaviť aký typ UJ chce filtrovať a tiež aj s akou príslušnosťou majú byť. Nakoniec sa vytvorí filter FilterTotalCitizens, kde sa prípadne nastaví aj ďalšie filtre ako FilterCuType alebo FilterUpperCu, ktoré pracujú s príslušnými triedami Selector. Výsledná zložitosť pri tejto podúlohe je $O(n + k)$. n reprezentuje počet územných jednotiek v tabuľke zatiaľ čo k reprezentuje počet vyfiltrovaných UJ pri výpise kde zložitosť je $k \cdot O(1)$.

Podúloha c) Filter zastavanost'

Podobne ako pri predchádzajúcej podúlohe si používateľ môže nastaviť aký typ UJ chce filtrovať a tiež aj s akou príslušnosťou majú byť. Nakoniec sa vytvorí filter `FilterTotalBuiltPercentage`, kde sa prípadne nastaví aj ďalšie filtre ako `FilterCuType` alebo `FilterUpperCu`, ktoré pracujú s príslušnými triedami `Selector`. Výsledná zložitosť pri tejto podúlohe je $O(n + k)$. n reprezentuje počet územných jednotiek v tabuľke zatiaľ čo k reprezentuje počet vyfiltrovaných UJ pri výpise kde zložitosť je $k \cdot O(1)$

Úroveň 2, 4

Tieto úrovne aplikácie predstavujú triedenie územných jednotiek na základe určitých atribútov. Štvrtá úroveň sa od druhej líši len v tom, že používateľ pred spustením triedenia zadá, že filtre pre typ a príslušnosť danej UJ sa majú preskočiť. Táto úroveň je v aplikácii reprezentovaná triedou `Level2`. V tejto úrovni je použitá pre prvé uloženie dát územných jednotiek **neutriedená sekvenčná tabuľka**. Ak používateľ zadá, že chce nastaviť určitý filter pred triedením ako napríklad na typ alebo príslušnosť triedených UJ pred samotným triedením sa urobí filtrovanie a následne sa triedenie uskutoční nad vyfiltrovanými dátami. Preto v tejto úrovni sú dve neutriedené tabuľky. Pred samotným začatím triedenia aplikácia vyzve používateľa pre zadanie toho či chce triediť vzostupne alebo zostupne. Tento prepínač je potom predávaný ako parameter.

Načítanie dát

Po zvolení danej úrovne používateľom aplikácia inicializuje príslušnú triedu `Level2`. Sekvenčne neutriedená tabuľka je predaná ako parameter pri volaní metódy `getAllObce()` nad objektom `Repository`. Táto metóda načítá nielen obce, ale aj kraje a okresy do príslušnej tabuľky. Výsledná zložitosť načítania jednej obce pri tejto úrovni teda je $O(1)$, z dôvodu vypnutie kontroly jedinečnosti vkladaneho kľúča. Súčasne spolu s načítaním dát o obci z CSV súboru `Obce SR.csv` sa načítava aj daná vyššie UJ, t.j. okres alebo kraj pre danú obec. Po načítaní názvu tejto UJ jednotky zo súboru `Územné členenie SR.csv` musí skontrolovať či už v zozname nie je. Tento zoznam je tvorený `Arraylistom`. Po pridaní všetkých obcí do tabuľky sa tieto dva `arraylisty` s kraji a okresmi tiež pridávajú do tabuľky. Výsledná zložitosť načítania kraja alebo okresu je teda $O(k)$ pri vypnutej kontrole jedinečnosti vkladaneho kľúča a k je počet uložených okresov (v prípade načítania kraja krajov) v `arrayliste`.

Podúloha a) Kritérium názov

V podúlohe používateľ môže zadať, ktorý typ UJ alebo územné jednotky s danou príslušnosťou chce triediť na začiatku tejto podúlohy. Aplikácia vytvorí `Filter` a nastaví do neho aj ďalší `Filter` (v prípade, že používateľ chce filtrovať oboma filterami). Aplikácia následne prefiltruje všetky územné jednotky a uloží ich do druhej neutriedenej tabuľky so zložitosťou filtrovania $O(n)$, kde n je počet územných jednotiek v tabuľke. Po odfiltrovaní správnych UJ aplikácia spustí triedenie typu **heap sort**, ktoré poskytuje trieda `Sort`. Ako parameter tejto metódy je predaná neutriedená tabuľka s odfiltrovanými územnými jednotkami a tiež `Selector` pre názov UJ. Toto triedenie má zložitosť $O(n \cdot \log_2(n))$, kde n

predstavuje počet prvkov v neutriedenej tabuľke. Po utriedení tabuľky sa údaje vypíšu so zložitou $O(n)$, kde n je p. prvkov. Teda ako výslednú zložitost' tejto operácie môžeme brať $O(k + n * \log_2(n) + n)$, kde n je p. prvkov v neutriedenej tabuľke po odfiltrovaní vyhovujúcich UJ a k predstavuje celkový počet UJ pre filtrovaním.

Podúloha a) Kritérium počet obyvateľov

Podobne ako v predchádzajúcej podúlohe používateľ môže zadať typ a príslušnosť s ktorou sa budú triediť územné jednotky. Zložitost' tohoto filtrovania sa nemení a ostáva $O(n)$, kde n je počet všetkých územných jednotiek. Samotné filtrovanie sa deje na základe triedy SelectorTotalCitizens. Výsledná zložitost' je teda tiež ako pri predchádzajúcej podúlohe $O(k + n * \log_2(n) + n)$, kde n je p. prvkov v neutriedenej tabuľke po odfiltrovaní vyhovujúcich UJ a k predstavuje celkový počet UJ pre filtrovaním.

Podúloha a) Kritérium zastavanost'

Podobne ako v predchádzajúcej podúlohe používateľ môže zadať typ a príslušnosť s ktorou sa budú triediť územné jednotky. Zložitost' tohoto filtrovania sa nemení a ostáva $O(n)$, kde n je počet všetkých územných jednotiek. Samotné filtrovanie sa deje na základe triedy SelectorBuiltPercentage. Výsledná zložitost' je teda tiež ako pri predchádzajúcej podúlohe $O(k + n * \log_2(n) + n)$, kde n je p. prvkov v neutriedenej tabuľke po odfiltrovaní vyhovujúcich UJ a k predstavuje celkový počet UJ pre filtrovaním.

Zložitosti jednotlivých filtrov

FilterName

Celková zložitost' samotného filtra je $O(\log(n))$, kde n je p. prvkov v tabuľke.

FilterUpperCu

Celková zložitost' samotného filtra je $O(n)$, kde n je p. prvkov v tabuľke.

FilterCuType

Celková zložitost' samotného filtra je $O(n)$, kde n je p. prvkov v tabuľke.

FilterBuiltPercentage

Celková zložitost' samotného filtra je $O(n)$, kde n je p. prvkov v tabuľke.

FilterTotalCitizens

Celková zložitost' samotného filtra je $O(n)$, kde n je p. prvkov v tabuľke.

Používateľská príručka

Zadávaný vstup do aplikácie sa realizuje pomocou vstupu do konzoly a to buď výberom číselných možností alebo zadáním danej hodnoty v závislosti od výzvy aplikácie.

Po spustení aplikácie je zobrazený výber jednotlivých úrovní. Výber sa realizuje pomocou číselného vstupu 1 alebo 2. Po zadaní tohto výberu a po jeho potvrdení aplikácia načíta dáta pre danú úroveň. Po načítaní dát je zas vypísaný výber jednotlivých podúrovní aplikácie. Tento výber je na základe číselného vstupu 1, 2 alebo 3.

Po vybratí podúrovne príde narad či chce používateľ aplikovať filter pre typ UJ. Toto je možnosť áno/nie - výber 1 alebo 2. Ak áno je vypísané nastavenie samotného filtra ak nie ide na rad výber filtra pre príslušnosť UJ. Tiež je to možnosť áno/nie. Po prejdení týchto výberov príde na rad samotná akcia pre danú podúroveň.

Nakoniec tejto akcie je výsledok vypísaný na konzolu a aplikácia končí.