**ŽILINSKÁ UNIVERZITA V ŽILINE**

**FAKULTA RIADENIA A INFORMATIKY**

Algoritmy a údajové štruktúry 1

Semestrálna práca 2

Adam Parimucha

5ZYI27

2020/2021

# Popis použitých údajových štruktúr

Územné jednotky sú uložené v utriedených a neutriedených sekvenčných tabuľkách

**UnsortedSequenceTable –** tabuľka sa používa na uchovávanie územných jednotiek a ich triedenie, pretože sorted tabuľka sa nedá triediť.

**SortedSequenceTable –** tabuľka sa používa na uchovávanie územných jednotiek a pri ich bodovom vyhľadávaní, kvôli nízkej zložitosti vyhľadávania prvkov (O(log2n)).

# UML diagram tried

Územné jednotky

# 

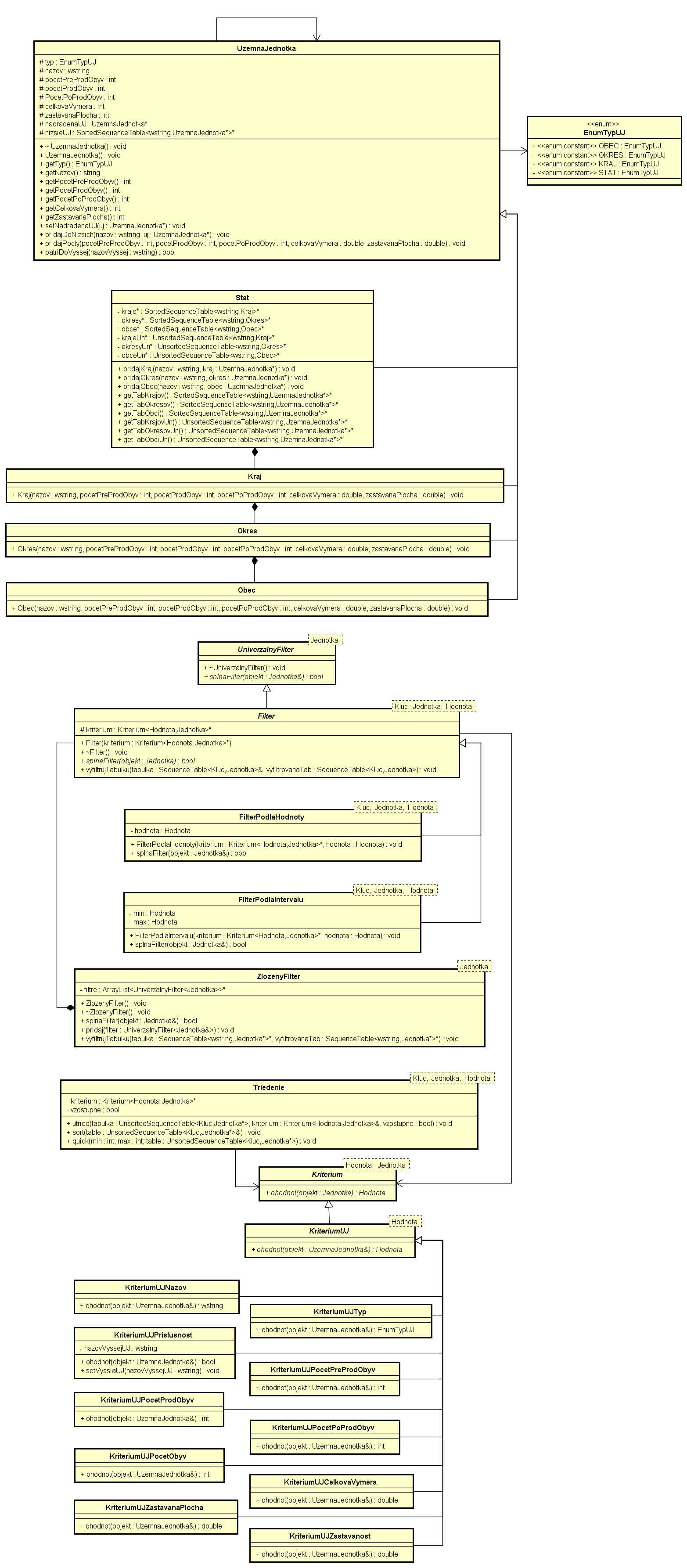
Trieda **UzemnaJednotka** má atribúty typ, nazov, pocetObyv, celkovaVymera atď., ako možno vidieť v diagrame a podľa týchto atribútov sa potom dá filtrovať.

Triedy **Stat, Kraj, Okres a Obec** dedia atribúty aj metódy z triedy **UzemnaJednotka**. Tvoria hierarchickú stromovú štruktúru takým spôsobom, že každá jednotka okrem štátu má atribút nadradenaUJ čo je smerník na predchodcu a tabuľku nizsieUJ, čo je tabuľka smerníkov na nižšie jednotky. Pomocou stromovej štruktúry potom pri vyhľadávaní budeme môcť vypísať informácie aj pre vyššie jednotky danej jednotky.

Trieda **Stat** má navyše sorted aj unsorted tabuľky všetkých krajov, okresov a obcí a tie tvoria ich „databázu“. Sorted tabuľky sú tam kvôli efektívnemu vyhľadávaniu a unsorted tabuľky kvôli tomu, aby sa jednotky dali triediť. Metóda nacitajZoSuboru v štáte načíta zo súborov Kraje,Okresy a obce a pre každú územnú jednotku vytvorí nový objekt, na ktorý sa ukazuje zo sorted a unsorted tabuliek v štáte a z nižších a vyšších územných jednotiek v každej územnej jednotke.

**EnumTypUJ** je enum, ktorý má typy OBEC,OKRES,STAT A KRAJ a slúži na to aby sa dal zistiť typ územnej jednotky.

Kritériá a filtre

**

**Kriterium** je abstraktná trieda používajúca template, kde Jednotka je objekt, ktorého hodnotu vlastnosti kritérium vracia a Hodnota je hodnota vlastnosti vstupného objektu, ktorú vráti metóda ohodnot. Z tejto triedy dedí **KriteriumUJ**, kde je za templatový parameter Jednotka dosadená UzemnaJednotka a je to takisto abstraktná trieda. Z triedy KriteriumUJ potom dedia triedy KriteriumUJNazov, KriteriumUJTyp, KriteriumUJPrislusnost atď. ako možno vidieť v UML diagrame. Každé z týchto kritérií vracia istú vlastnosť objektu napr. KriteriumUJNazov vráti názov územnej jednotky, KriteriumUJPocetObyv vráti počet obyvateľov v územnej jednotke a pod.

Trieda **Triedenie** využíva template, kde Kluc je kľúč tabuľky, Jednotka je objekt tabuľky a Hodnota je hodnota vlastnosti objektu, podľa ktorej sa má triediť. V metóde utried preberie kritériá, tabuľku, ktorá sa má utriediť a parameter, ktorý určuje či sa má triediť vzostupne. Následne triedi tabuľku tak, že porovnáva objekty podľa hodnoty vlastností, ktoré vracia kritérium. Na triedenie používam algoritmus QuickSort.

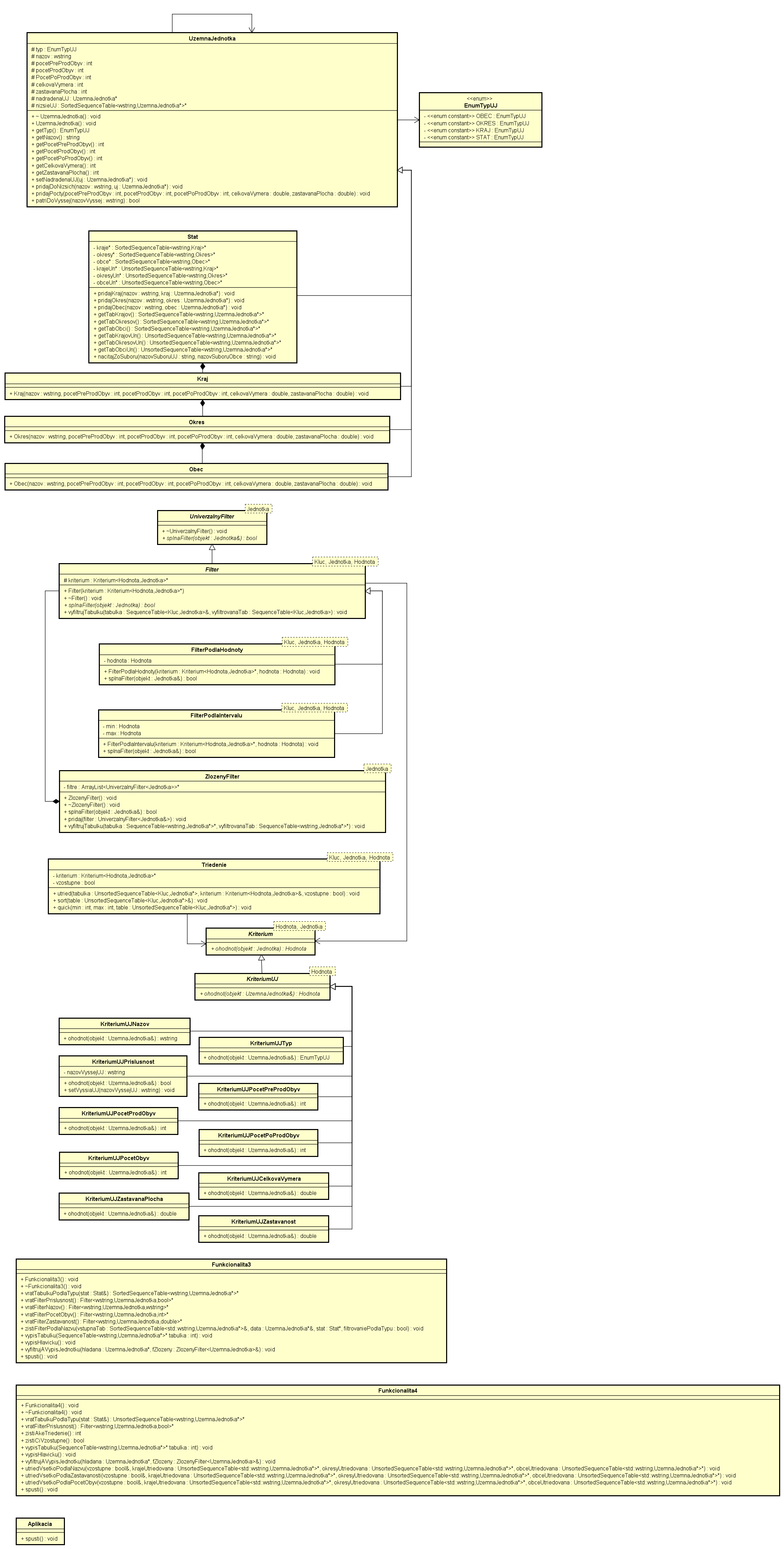
Trieda **UniverzalnyFilter** je abstraktná trieda kde Jednotka je objekt tabuľky.

Trieda **Filter** je potomok triedy UniverzalnyFilter a využíva template, kde Kluc je kľúč tabuľky, Jednotka je objekt tabuľky a Hodnota je hodnota vlastnosti objektu, podľa ktorej sa má filtrovať. Filter preberá kritérium, a v metóde vyfiltrujTabulku preberie tabuľku, ktorá sa má vyfiltrovať a tabuľku, do ktorej sa vložia vyfiltrované záznamy. Následne porovnáva hodnotu vlastnosti objektu so žiadanou vlastnosťou pre všetky položky v tabuľke a ak ju objekt spĺňa, zaradí ho do vyfiltrovanej tabuľky.

Z triedy Filter dedia triedy **FilterPodlaHodnoty** a **FilterPodlaIntervalu**. Jediný rozdiel medzi nimi je, že FilterPodlaHodnoty preberie hodnotu a porovnáva či sa hodnota vlastnosti objektu rovná zadanej hodnote a FilterPodlaIntervalu preberie minimum a maximum hodnoty a porovnáva či hodnota vlastnosti objektu je v intervale <minimum,maximum>.

Trieda **ZlozenyFilter** umožňuje kontrolovať viacero filtrov a to tak, že drží ArrayList univerzálnych filtrov, do ktorého sa pridávajú filtre metódou pridajFilter. V metóde splnaFilter sa pre každý filter v ArrayListe vyhodnocuje či ho objekt spĺňa a ak narazí na prvý filter, ktorý objekt nespĺňa, vráti false.

Funkcionality



Trieda **Funkcionalita3** vykonáva funkcionalitu tretej úrovne. Metóda vratTabulku podľa typu sa užívateľa spýta či chce filtrovať podľa typu a ak áno tak podľa akého a vráti tabuľku príslušného typu zo štátu. Ak užívateľ nechce filtrovať podľa typu, vráti nullptr. Metódy vratFilterPrislusnost, vratFilterNazov, vratFilterPocetObyv a vratFilterZastavanost sa užívateľa spýtajú či chce aplikovať daný filter, načítajú príslušné parametre a vytvoria a vrátia daný filter. Metóda zistiFilterPodlaNazvu sa užívateľa spýta či chce filtrovať podľa názvu a ak hodnota filtrovaniePodlaTypu je true, a teda sa filtruje podľa typu, prehľadá sa vstupná tabuľka a nájdená územná jednotka sa zapíše do vstupného parametra data. Ak sa nefiltruje podľa typu, tak prehľadá všetky tabuľky štátu, ktorý dostane vo vstupnom parametri a nájdenú jednotku zapíše do vstupného parametra data. Nakoniec vráti true alebo false podľa toho či užívateľ chcel alebo nechcel filtrovať podľa názvu. Metóda vyfiltrujAVypisJednotku berie územnú jednotku, na ktorú sa majú aplikovať filtre a zložený filter, ktorý sa na ňu má aplikovať. Na jednotku z parametra sa aplikuje zložený filter a ak ho spĺňa tak sa vypíše, inak sa vypíše že jednotka nespĺňa zadané parametre alebo ak je jednotka z parametra nullptr vypíše sa, že jednotka nebula nájdená. V metóde spusti sa vytvorí štát a načítajú sa doňho územné jednotky z CSV súborov. Následne sa zisťuje či chce užívateľ filtrovať podľa typu. Ak áno, zavolá sa metóda vrtTabulkuPodlaTypu, ktorá vráti tabuľku, ktorá sa bude filtrovať. Potom sa od užívateľa zisťuje, či chce aplikovať filtre podľa názvu, počtu obyvateľov a zastavanosti a tieto sa pridajú do zloženého filtra. Ďalej sa zisťuje či je zapnuté filtrovanie podľa typu. Ak áno tak sa ešte zistí či je zapnuté filtrovanie podľa názvu a ak je, tak za zavolá metóda vyfiltrujAVypisJednotku, a ak nie je tak sa filtruje celá tabuľka podľa zadaného typu a následne sa vypíše alebo ak je prázdna, vypíše sa, že zadaným parametrom neodpovedá žiadna jednotka. Ak nie je zapnuté filtrovanie podľa typu tak sa zisťuje či je zapnuté filtrovanie podľa názvu a ak áno, znova sa filtre aplikujú len na jednu jednotku a ak nie, tak sa od štátu vypýtajú postupne tabuľky krajov, okresov a obcí a tie sa vyfiltrujú a vypíšu.

Trieda **Funkcionalita4** vykonáva funkcionalitu štvrtej úrovne. Metóda vratTabulkuPodlaTypu je v podstate rovnaká ako pri funkcionalite 3, s tým rozdielom, že vracia neutriedenú tabuľku zo štátu. Metóda vratFilterPrislusnost sa užívateľa spýta či chce aplikovať daný filter, načíta príslušné parametre a vytvorí a vráti daný filter. Metóda zistiAkeTriedenie sa užívateľa spýta podľa čoho chce triediť a vráti integer 1 ak podľa názvu, 2 ak podľa počtu obyvateľov a 3 ak podľa zastavanosti. Metóda zistiCiVzostupne sa užívateľa spýta či chce filtrovať vzostupne a vráti true ak áno a false ak nie. Metódy utriedVsetkoPodlaNazvu, utriedVsetkoPodlaZastavanosti a utriedVsetkoPodlaPocetObyv berú tri tabuľky – krajov okresov a obcí a utriedia ich. V metóde spusti sa vytvorí štát a načítajú sa doň územné jednotky z CSV súborov. Následne sa vytvorí vstupná tabuľka a filtrovaná tabuľka a do vstupnej tabuľky sa priradí tabuľka, ktorú vráti metóda vratTabulkuPodlaTypu. Ďalej sa zistí, či sa filtruje podľa príslušnosti a ak áno, vytvorí sa taký filter. Potom sa zavolá metóda zistiAkeTriedene a výsledok sa zapíše do premennej moznostTriedenie. Následne ak je zapnuté filtrovanie podľa typu, tak sa vyfiltruje vstupná tabuľka podľa príslušnosti a potom sa vstupná tabuľka utriedi podľa hodnoty v premennej moznostTriedenia a nakoniec sa vypíše. Ak je vstupná tabuľka prázdna, vypíše sa, že zadaným parametrom neodpovedá žiadna jednotka. Ak nie je zapnuté filtrovanie podľa typu, tak sa vypýtajú postupne tabuľky krajov, okresov a obcí od štátu a ak filter podľa príslušnosti nie je nullptr, tak sa postupne vyfiltrujú podľa príslušnosti. Následne sa zavolá metóda utriedPodlaNazvu/PocetObyv/Zastavanosti podľa hodnoty v premennej moznostTriedenie a vypíšu sa zoradené tabuľky spolu s hodnotou kritéria podľa ktorého sa triedilo. Ak sú tabuľky prázdne, tak sa vypíše, že zadaným parametrom neodpovedá žiadna jednotka.

Trieda **Aplikácia** v metóde spusti sa spýta užívateľa, ktorú funkcionalitu chce spustiť a následne vytvorí inštanciu triedy danej funkcionality a spustí ju.

Popis výpočtu každého vyhľadávacieho, resp. triediaceho kritéria, nachádzajúceho sa v tretej úrovni

**KUJNázov,KUJTyp,KUJPočetPreproduktívnychObyvateľov,KUJPočetProduktívnychObyvateľov,KUJPočetPoproduktívnychObyvateľov,KUJCelkováVýmera,KUJZastavanáPlocha**-V týchto kritériách sa zavolá getter územnej jednotky, ktorý vráti hodnotu atribútu v hornom indexe kritéria

**KUJPríslušnosť** – zavolá na danú územnú jednotku metódu patriDoVyssej, ktorá rekurzívne zisťuje, či sa názov niektorej vyššej jednotky rovná názvu jednotky, do ktorej príslušnosť chceme zistiť a vráti návratovú hodnotu tejto metódy.

**KUJPočetObyvateľov –** spočíta hodnoty atribútov pocetPreProdObyv,PocetProbObyv a pocetPoProdObyv z územnej jednotky a vráti výsledok

**KUJZastavanosť-** vydelí atribúty zastavanaPlocha/celkovaVymera z územnej jednotky, vynásobí 100 a vráti výsledok

# Zložitosti jednotlivých vyhľadávacích kritérií

V oboch úrovniach sa využívajú 2 filtre - Filter FUJTyp a FUJPríslušnosť. FUJTyp má zložitosť O(1), pretože ide len o vybratie tabuľky zadaného typu zo štátu. FUJPríslušnosť má zložitosť nanajvýš O(3), pretože ide o rekurzívne pristupovanie na vyššiu jednotku, ktoré má zložitosť O(1) a v najhoršom prípade, teda ak sa začína od obce sa pristúpi k vyššej jednotke 3 krát (obec->okres->kraj->štát).

Načítavanie územných jednotiek má zložitosť O(N\*log­­2N+N+5\*log2N), kde N je počet všetkých územných pretože každá jednotka sa pridáva do utriedených tabuliek (O(N\*log2N)), potom do neutriedených tabuliek (O(N\*1)), okresy a kraje sa vyhľadávajú v utriedených tabuľkách či tam už nie sú (O(2\*log2N)) a následne sa pridajú do nižších jednotiek v štáte, kraji a okrese (3\*O(log2N))

## Úroveň 3

**Operácia 3 a)** (vyhľadanie územnej jednotky na základe jej názvu a vypísanie údajov o nej) - zložitosť O(log2N+k+p), kde N je počet územných jednotiek daného typu, lebo najskôr je potrebné vyhľadať územnú jednotku podľa názvu v utriedenej tabuľke, potom vypísať k údajov o jednotke(zložitosť k\*O(1)) a ešte vypísať názvy p vyšších jednotiek, ku ktorým sa pristupuje cez pointer(p\*O(1)).

**Operácia 3 b)** (vyfiltrovanie územných jednotiek na základe ich počtu obyvateľov a vypísanie údajov o nich) - zložitosť O(N+N\*(k+p)), kde N je počet územných jednotiek daného typu, lebo najskôr je potrebné vyfiltrovať N územných jednotiek, ktorých počet obyvateľov je v žiadanom intervale v neutriedenej tabuľke, potom vypísať pre vyfiltrované jednotky, ktorých bude v najhoršom prípade N k údajov (k\*O(1)) a ešte vypísať názvy p vyšších jednotiek, ku ktorým sa pristupuje cez pointer(p\*O(1)).

**Operácia 3 c)** (vyfiltrovanie územných jednotiek na základe ich zastavanosti a vypísanie údajov o nich) - zložitosť O(N+N\*(k+p)), kde N je počet územných jednotiek daného typu, lebo najskôr je potrebné vyfiltrovať N územných jednotiek, ktorých zastavanosť je v žiadanom intervale v neutriedenej tabuľke, potom vypísať pre vyfiltrované jednotky, ktorých bude v najhoršom prípade N k údajov (k\*O(1)) a ešte vypísať názvy p vyšších jednotiek, ku ktorým sa pristupuje cez pointer(p\*O(1)).

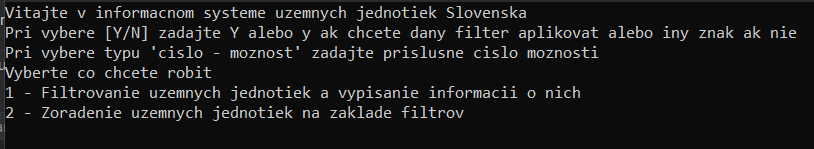
## Úroveň 4

**Operácia 4 a)** (utriedenie tabuľky na základe názvu a vypísanie údajov) – zložitosť O(N\*log­2N+N), kde N je počet jednotiek v tabuľke, pretože zložitosť triedenia quicksortom je N\*log­2N, a potom sa vypíše o N jednotkách 1 údaj, a to jej názov (O(1)).

**Operácia 4 b)** (utriedenie tabuľky na základe počtu obyvateľov a vypísanie údajov) – zložitosť O(N\*log­2N+N\*2), kde N je počet jednotiek v tabuľke, pretože zložitosť triedenia quicksortom je N\*log­2N, a potom sa vypíšu o N jednotkách 2 údaje (názov a počet obyvateľov) (O(2)).

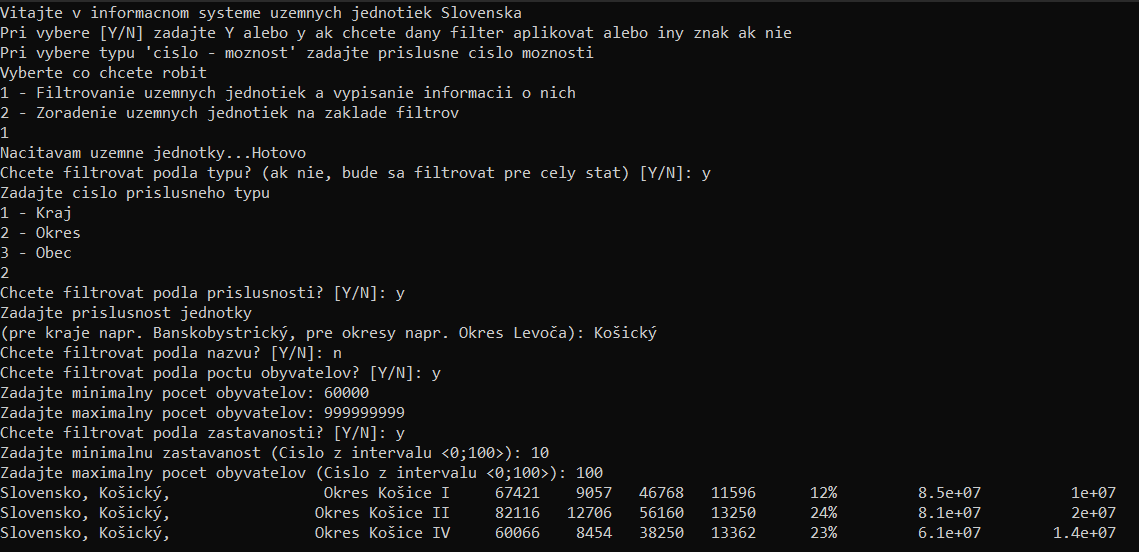
**Operácia 4 c)** (utriedenie tabuľky na základe zastavanosti a vypísanie údajov) – zložitosť O(N\*log­2N+N\*2), kde N je počet jednotiek v tabuľke, pretože zložitosť triedenia quicksortom je N\*log­2N, a potom sa vypíšu o N jednotkách 2 údaje (názov a zastavanosť) (O(2)).

# Používateľská príručka



Na začiatku program privíta používateľa a vypíšu sa inštrukcie k výberu možností. Používateľ si môže vybrať či chce získať informácie o územných jednotkách na základe zvolených filtrov alebo zoradiť jednotky spĺňajúce zadaný filter. Po zadaní zlého čísla aplikácia užívateľa upozorní a vyzve ho na zadanie platnej možnosti, ale je nutné zadávať len čísla, inak sa aplikácia môže správať nekorektne. Pri otázkach ďalej v aplikácii, kde je na konci otázky [Y/N] používateľ zadá Y alebo y, ak chce daný filter aplikovať alebo n alebo iný znak okrem prázdneho znaku ak nie. Používateľ si vyberie zadaním príslušnej možnosti a potvrdením stlačením tlačidla Enter.

Filtrovanie územných jednotiek a vypísanie informácií o nich



Po výbere tejto možnosti sa zo súboru načíta databáza územných jednotiek, čo môže chvíľu trvať a následne si užívateľ vyberá podľa inštrukcií na obrazovke.

Filtrovanie podľa typu

Používateľ si môže vybrať či chce filtrovať podľa typu, to znamená či chce špecificky len Kraje, okresy alebo obce. Ak používateľ neaplikuje tento filter budú sa ďalej filtrovať všetky nižšie jednotky v štáte t.j. kraje, okresy aj obce.

Filtrovanie podľa príslušnosti

Používateľ si môže zvoliť či chce filtrovať podľa príslušnosti, a teda že chce len tie jednotky ktoré patria pod jednotku, ktorej názov zadá v ďalšom kroku. Je nutné kraje zadávať ako prídavné meno napr. ako „Banskobystrický“ alebo „Košický“, teda bez slova kraj a samozrejme aj bez úvodzoviek a okresy ako Okres [Názov] napríklad „Okres Žilina“ alebo „Okres Spišská Nová Ves“.

Filtrovanie podľa názvu

Ak si používateľ vyberie, že chce filtrovať podľa názvu, aplikácia ho vyzve aby zadal názov územnej jednotky, ktorú chce hľadať. Názvy krajov a okresov sa zadávajú rovnako ako vo filtri podľa príslušnosti a názvy obcí sa zadávajú ako názov obce napr. „Zlaté Moravce“ alebo „Leopoldov“ a pod.

Filtrovanie podľa počtu obyvateľov

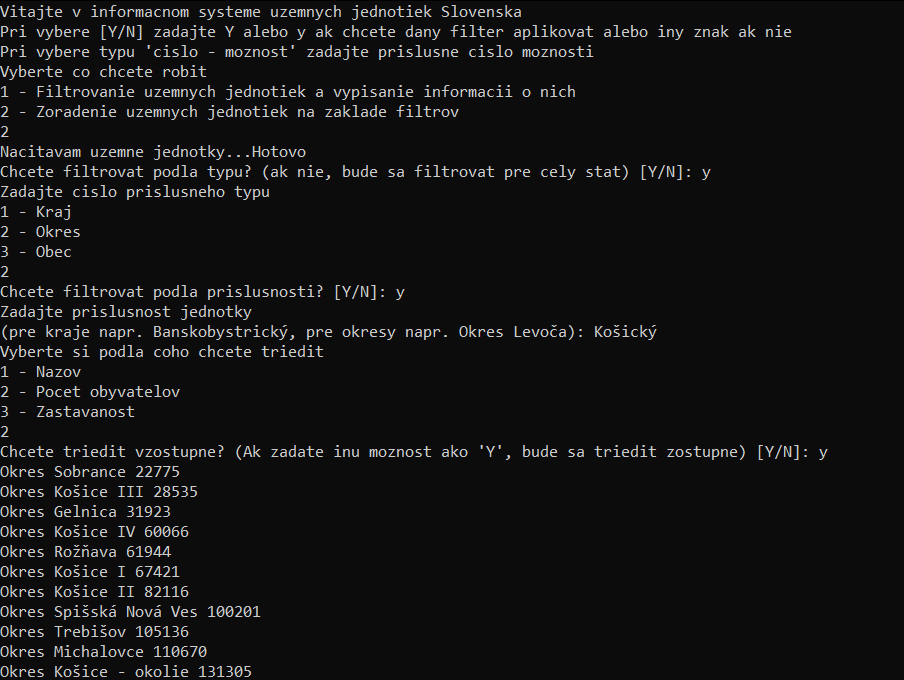
Ak si používateľ vyberie filtrovanie podľa počtu obyvateľov, aplikácia vyzve používateľa na minima a maxima intervalu, podľa ktorého chce filtrovať. Ak sa používateľ pomýli a zadá maximum menšie ako minimum alebo minimum menšie ako 0, aplikácia ho vyzve aby ich zadal znova. Je nutné zadať aj maximum aj minimum, preto keď používateľ nechce žiadnu dolnú hranicu mal by zadať 0 a ak nechce žiadnu hornú hranicu, mal by zadať nejaké veľké číslo, najviac však 2.147.483.647. Je nutné zadávať celé čísla inak sa aplikácia môže správať nekorektne.

Filtrovanie podľa zastavanosti

Ak si používateľ vyberie filtrovanie podľa počtu obyvateľov, aplikácia vyzve používateľa na minima a maxima intervalu, podľa ktorého chce filtrovať. Minimum aj maximum musia byť z intervalu <0;100>,pretože ide o percentá. Ak sa používateľ pomýli a zadá maximum menšie ako minimum alebo nezadá hodnotu z intervalu <0;100>, aplikácia ho vyzve aby ich zadal znova. Je nutné zadať aj maximum aj minimum, preto keď používateľ nechce žiadnu dolnú hranicu mal by zadať 0 a ak nechce žiadnu hornú hranicu, mal by zadať 100. Je nutné zadávať celé čísla inak sa aplikácia môže správať nekorektne.

Po aplikovaní všetkých filtrov aplikácia vypíše územné jednotky, ktoré tieto filtre spĺňajú a všetky jej nadradené územné jednotky, alebo vypíše, že žiadna územná jednotka nespĺňa aplikované filtre. Za názvom územnej jednotky označujú stĺpce veci v tomto poradí: Počet obyvateľov, počet predproduktívnych obyvateľov, počet produktívnych obyvateľov, počet poproduktívnych obyvateľov, zastavanosť, celkovú výmeru a zastavanú plochu územnej jednotky.

Zoradenie územných jednotiek na základe filtrov



Najprv sa zo súboru načíta databáza územných jednotiek, čo môže chvíľu trvať a následne si užívateľ vyberá podľa inštrukcií na obrazovke.

Na začiatku môže používateľ filtrovať na základe typu a príslušnosti (viď. vyššie). Po zvolení filtrov sa aplikácia používateľa spýta či chce filtrovať podľa názvu, počtu obyvateľov alebo zastavanosti. Dá sa vybrať vždy len jedna možnosť. Následne sa aplikácia užívateľa spýta, či chce územné jednotky zoradiť vzostupne alebo zostupne. Po vybratí možnosti aplikácia vypíše územné jednotky zoradené podľa výberu používateľa a hodnotu kritéria podľa ktorého sa zoradzovalo.

Pre iné filtrovanie alebo triedenie je nutné aplikáciu spustiť znovu.