**Žilinská univerzita v Žiline**

Fakulta Riadenia a Informatiky

Územné jednotky na Slovensku

Semestrálna práca

Filip Sudora

5ZYI23

Obsah

[UML diagram tried aplikácie 3](#_Toc72701542)

[Použité údajové štruktúry 5](#_Toc72701543)

[Sekvenčne utriedená tabuľka 5](#_Toc72701544)

[Sekvenčne neutriedená tabuľka 5](#_Toc72701545)

[Triedy použité pre prístup k dátam 5](#_Toc72701546)

[Územná jednotka 5](#_Toc72701547)

[Strom územných jednotiek 5](#_Toc72701548)

[Popis výpočtu kritérií 5](#_Toc72701549)

[Kritérium územná jednotka **počet obyvateľov** 5](#_Toc72701550)

[Kritérium územná jednotka **počet predproduktívnych obyvateľov** 5](#_Toc72701551)

[Kritérium územná jednotka **počet produktívnych obyvateľov** 5](#_Toc72701552)

[Kritérium územná jednotka **počet poproduktívnych obyvateľov** 6](#_Toc72701553)

[Kritérium územná jednotka **celková výmera** 6](#_Toc72701554)

[Kritérium územná jednotka **zastavaná plocha** 6](#_Toc72701555)

[Kritérium územná jednotka **zastavanosť** 6](#_Toc72701556)

[Kritérium územná jednotka **názov** 6](#_Toc72701557)

[Kritérium územná jednotka **typ** 6](#_Toc72701558)

[Kritérium územná jednotka **príslušnosť** 6](#_Toc72701559)

[Zložitosť operácií 6](#_Toc72701560)

[Zložitosť načítavania dát 6](#_Toc72701561)

[Funkcionalita č.**3** 7](#_Toc72701562)

[Funkcionalita č.**4** 7](#_Toc72701563)

[Používateľská príručka 7](#_Toc72701564)

# UML diagram tried aplikácie

Trieda **Triedenie** triedi zadanú tabuľku podľa zvoleného kritéria, triediacim algoritmov QuickSort. Trieda **Kritérium** je abstraktná trieda, od ktorej dedia všetky ostatné kritériá. **KriteriumUJ** je abstraktná trieda pre všetky kritériá územných jednotiek. Táto trieda má ďalších potomkov, títo reprezentujú jednotlivé kritériá viď Obrázok 1. Podobným spôsobom sú zostavené aj filtre. Budem vytvárať potomkov, ktorý budú mať konštruktory s hodnotou podľa ktorej sa budú filtrovať a s tabuľkou, ktorú majú vyfiltrovať viď Obrázok 3. Obidve tieto spôsoby zostavenia tried sú vytvorené za účelom zovšeobecnenia triedenia a filtrovania. Posledné zoskupenie tried si ukladá načítané dáta. Trieda **UzemnaJednotka** si drží údaje o: názve, type(štát, kraj, okres, obec), tabuľke nižších územných jednotiek a smerník na vyššiu územnú jednotku. Ak je daná územná jednotka obec uchovám si o nej zároveň aj dáta čo predstavuje smerník na inštanciu triedy **ObecData**. Vytváram triedy **UzemnaJednotka**, ktoré na seba odkazujú. Na začiatku si vytvorím územnú jednotku Slovenská republika, ktorá je koreňom stromu. Následne prechádzam súborom a do tabuľky nižších územných jednotiek Slovenskej republiky si postupne pridám všetky kraje, ak sa daný kraj v tabuľke ešte nenachádza. Ak nie, vytvorím územnú jednotku typu kraj a chronologicky opakujem postup až kým nenačítam celý súbor. Výsledné načítané dáta dávam do tabuliek v triede **StromUzemneJednotky**.

Obrázok, na ktorom je text, snímka obrazovky, vnútri

Automaticky generovaný popis

Obrázok 1 UML diagram tried triedenia a kritérií

Obrázok, na ktorom je text, snímka obrazovky, vnútri

Automaticky generovaný popis

Obrázok 2 UML diagram tried územnej jednotky

Obrázok, na ktorom je text, snímka obrazovky, vnútri, prenosný počítač

Automaticky generovaný popis

Obrázok 3 UML diagram tried filtrov

# Použité údajové štruktúry

## Sekvenčne utriedená tabuľka

Vybraný typ tabuľky používam na základe toho, že má nízku zložitosť pre vyhľadávanie prvku O(logN), čo môžem využiť pri hľadaní prvku v tabuľke podľa kľúča . Keďže tabuľku nebudem napĺňať príliš často tak zložitosť operácií Vlož/Zruš prvok nie je až tak dôležitá.

## Sekvenčne neutriedená tabuľka

Tento typ tabuľky používam kvôli možnosti utriedenia aj iným spôsobom ako len podľa kľúča, ako je to v prípade sekvenčne utriedenej tabuľky. Keďže potrebujem triediť aj podľa iných kritérií.

# Triedy použité pre prístup k dátam

## Územná jednotka

Predstavuje hocijakú územnú jednotku (obec/okres/kraj/štát). Ukladá si názov, typ, smerník ktorí ukazuje na vyššiu územnú jednotku, smerník na tabuľku nižších územných jednotiek a ak je územná jednotka typu obec ukladám smerník na dáta danej obce.

## Strom územných jednotiek

Pre jednoduchší prístup k územným jednotkám som si spravil vlastnú triedu. Celý „strom“ som zostavil zo 4 úrovní, implementovaných sekvenčne utriedenou tabuľkou. Koreň (prvá úroveň) stromu je tabuľka územných jednotiek, v ktorej je jediný prvok - územná jednotka Slovenská republika, ktorá má v sebe „potomkov“ (druhá úroveň) všetky kraje na Slovensku, t.j. Bratislavský, Trnavský, Žilinský,... Tieto kraje majú „potomkov“ (tretiu úroveň) okresy podľa toho do ktorého kraju patria. Nakoniec sú obce ako štvrtá úroveň „stromu“, zaradené pod svoj okres. Týmto spôsobom budem môcť efektívne pristupovať ku každej územnej jednotke.

# Popis výpočtu kritérií

Každá územná jednotka si pamätá tabuľku nižších územných jednotiek (okrem obcí, tie si pamätajú načítané dáta). Postupným prehľadávaním „stromu“ sa dostanem k dátam každej obce, ktorá patrí pod zadanú územnú jednotku.

## Kritérium územná jednotka **počet obyvateľov**

Pomocou metódy **getTyp()** triedy **UzemnaJednotka** si zistím typ územnej jednotky, ktorú budem filtrovať/zoraďovať. Na základe typu a tabuľky nižších územných jednotiek vypočítam počet obyvateľov danej územnej jednotky ako súčet predproduktívnych, produktívnych a poproduktívnych obyvateľov všetkých obcí danej územnej jednotky, tým spôsobom, že budem pristupovať ku dátam každej obce pomocou metódy **getPocetObyv()**.

## Kritérium územná jednotka **počet predproduktívnych obyvateľov**

Pomocou metódy **getTyp()** triedy **UzemnaJednotka** si zistím typ územnej jednotky, ktorú budem filtrovať/zoraďovať. Na základe typu a tabuľky nižších územných jednotiek vypočítam počet predproduktívnych obyvateľov danej územnej jednotky ako súčet predproduktívnych obyvateľov všetkých obcí danej územnej jednotky, tým spôsobom, že budem pristupovať ku dátam každej obce pomocou metódy **getPreProdObyv()**.

## Kritérium územná jednotka **počet produktívnych obyvateľov**

Pomocou metódy **getTyp()** triedy **UzemnaJednotka** si zistím typ územnej jednotky, ktorú budem filtrovať/zoraďovať. Na základe typu a tabuľky nižších územných jednotiek vypočítam počet produktívnych obyvateľov danej územnej jednotky ako súčet produktívnych obyvateľov všetkých obcí danej územnej jednotky, tým spôsobom, že budem pristupovať ku dátam každej obce pomocou metódy **getProdObyv()**.

## Kritérium územná jednotka **počet poproduktívnych obyvateľov**

Pomocou metódy **getTyp()** triedy **UzemnaJednotka** si zistím typ územnej jednotky, ktorú budem filtrovať/zoraďovať. Na základe typu a tabuľky nižších územných jednotiek vypočítam počet poproduktívnych obyvateľov danej územnej jednotky ako súčet poproduktívnych obyvateľov všetkých obcí danej územnej jednotky, tým spôsobom, že budem pristupovať ku dátam každej obce pomocou metódy **getPoProdObyv()**.

## Kritérium územná jednotka **celková výmera**

Pomocou metódy **getTyp()** triedy **UzemnaJednotka** si zistím typ územnej jednotky, ktorú budem filtrovať/zoraďovať. Na základe typu a tabuľky nižších územných jednotiek vypočítam celkovú výmeru danej územnej jednotky ako súčet výmeru všetkých obcí danej územnej jednotky, tým spôsobom, že budem pristupovať ku dátam každej obce pomocou metódy **getVymera()**.

## Kritérium územná jednotka **zastavaná plocha**

Pomocou metódy **getTyp()** triedy **UzemnaJednotka** si zistím typ územnej jednotky, ktorú budem filtrovať/zoraďovať. Na základe typu a tabuľky nižších územných jednotiek vypočítam zastavanú plochu danej územnej jednotky ako súčet zastavaných plôch všetkých obcí danej územnej jednotky, tým spôsobom, že budem pristupovať ku dátam každej obce pomocou metódy **getZastPlocha()**.

## Kritérium územná jednotka **zastavanosť**

Pomocou metódy **getTyp()** triedy **UzemnaJednotka** si zistím typ územnej jednotky, ktorú budem filtrovať/zoraďovať. Na základe typu a tabuľky nižších územných jednotiek vypočítam zastavanosť danej územnej jednotky ako súčet zastavaností(100\*(zastavaná plocha/celková výmera)) všetkých obcí danej územnej jednotky, tým spôsobom, že budem pristupovať ku dátam každej obce pomocou metód **getZastPlocha(), getVymera()**.

## Kritérium územná jednotka **názov**

Pomocou metódy **getNazov()** zistím názov územnej jednotky.

## Kritérium územná jednotka **typ**

Pomocou metódy **getTyp()** zistím typ územnej jednotky.

## Kritérium územná jednotka **príslušnosť**

Trieda **UzemnaJednotka** má atribút **vyssia**, ktorý je pointer na vyššiu územnú jednotku. Postupne budem prehľadávať všetky územné jednotky, ktoré sú nad ňou. Ak sa hodnota tohto atribútu rovná so zadaným vyšším celkom vrátim *true*, inak *false*.

# Zložitosť operácií

## Zložitosť načítavania dát

Najskôr si načítavam dáta všetkých obcí, do sekvenčne utriedenej tabuľky so zložitosťou O(N + k), kde N je počet všetkých obcí a „k“ je počet údajov pre jednu obec (k\*O(1)). Potom prejdem na načítavanie údajov o územných jednotkách, kde vkladám záznamy do troch sekvenčne utriedených tabuliek, čiže zložitosť tejto operácie je O([počet záznamov 1. tabuľky] + k) + O([počet záznamov 2. tabuľky] + k) + O([počet záznamov 3. tabuľky] + k), kde „k“ je počet údajov pre jednu územnú jednotku (k\*O(1)).

## Funkcionalita č.**3**

* Operácia 3A (filtrovanie územnej jednotky na základe jej názvu a vypísaniu údajov o nej) má zložitosť O(N + k), kde N je počet všetkých územných jednotiek zadaného typu spadajúcich pod zadanú príslušnosť, pretože najskôr je potrebné vyfiltrovať územnú jednotku podľa názvu (N operácií), a potom vypísať „k“ údajov (k\*O(1)).
* Operácia 3B (filtrovanie územnej jednotky na základe jej počtu obyvateľov a vypísaniu údajov o nej) má zložitosť O(N + k), kde N je počet všetkých územných jednotiek zadaného typu spadajúcich pod zadanú príslušnosť, pretože najskôr je potrebné vyfiltrovať územnú jednotku podľa počtu obyvateľov (N operácií), a potom vypísať „k“ údajov (k\*O(1)).
* Operácia 3C (filtrovanie územnej jednotky na základe jej zastavanosti a vypísaniu údajov o nej) má zložitosť O(N + k), kde N je počet všetkých územných jednotiek, pretože najskôr je potrebné vyfiltrovať územnú jednotku podľa zastavanosti (N operácií), a potom vypísať „k“ údajov (k\*O(1)).

## Funkcionalita č.**4**

* Operácia 4A (zoradenie územných jednotiek na základe ich názvu a vypísaniu údajov o nich) má zložitosť O( + k), kde N je počet všetkých územných jednotiek zadaného typu spadajúcu pod danú príslušnosť, pretože najskôr je potrebné zoradiť územné jednotky podľa názvu, triediacim algoritmom QuickSort ( operácií), a potom vypísať „k“ údajov (k\*O(1)).
* Operácia 4B (zoradenie územných jednotiek na základe ich počtu obyvateľov a vypísaniu údajov o nich) má zložitosť O( + k), kde N je počet všetkých územných jednotiek zadaného typu, pretože najskôr je potrebné zoradiť územné jednotky podľa počtu obyvateľov, triediacim algoritmom QuickSort ( operácií), a potom vypísať „k“ údajov (k\*O(1)).
* Operácia 4C (zoradenie územných jednotiek na základe ich počtu obyvateľov a vypísaniu údajov o nich) má zložitosť O( + k), kde N je počet všetkých územných jednotiek, pretože najskôr je potrebné zoradiť územné jednotky podľa zastavanosti, triediacim algoritmom QuickSort ( operácií), a potom vypísať „k“ údajov (k\*O(1)).

# Používateľská príručka

Používateľ komunikuje s aplikáciou prostredníctvom konzoly. Pomocou nej postupným vyberaním operácií určuje čo chce zobraziť. Pri vstupe do aplikácie sa používateľovi zobrazí úvodný text viď Obrázok 4. Používateľ si zvolí akú operáciu chce vykonávať a prejde na vypisovanie informácií viď Obrázok 5 alebo zoradenie územných jednotiek viď Obrázok 7. Následne po zadaní špecifikácií sa používateľovi zobrazí výstup filtrovania viď Obrázok 6 alebo zoraďovania viď Obrázok 8.

Obrázok, na ktorom je text

Automaticky generovaný popis

Obrázok 4 vstupný bod aplikácie

Obrázok, na ktorom je text

Automaticky generovaný popis

Obrázok 5 zadávanie špecifikácií používateľom pri filtrovaní

Obrázok, na ktorom je text

Automaticky generovaný popis

Obrázok 6 výpis dát po filtrovaní

Obrázok, na ktorom je text

Automaticky generovaný popisObrázok, na ktorom je text

Automaticky generovaný popis

Obrázok 8 výpis dát po zoraďovaní

Obrázok 7 zadávanie špecifikácií používateľom pri zoraďovaní