**ŽILINSKÁ UNIVERZITA V ŽILINE**

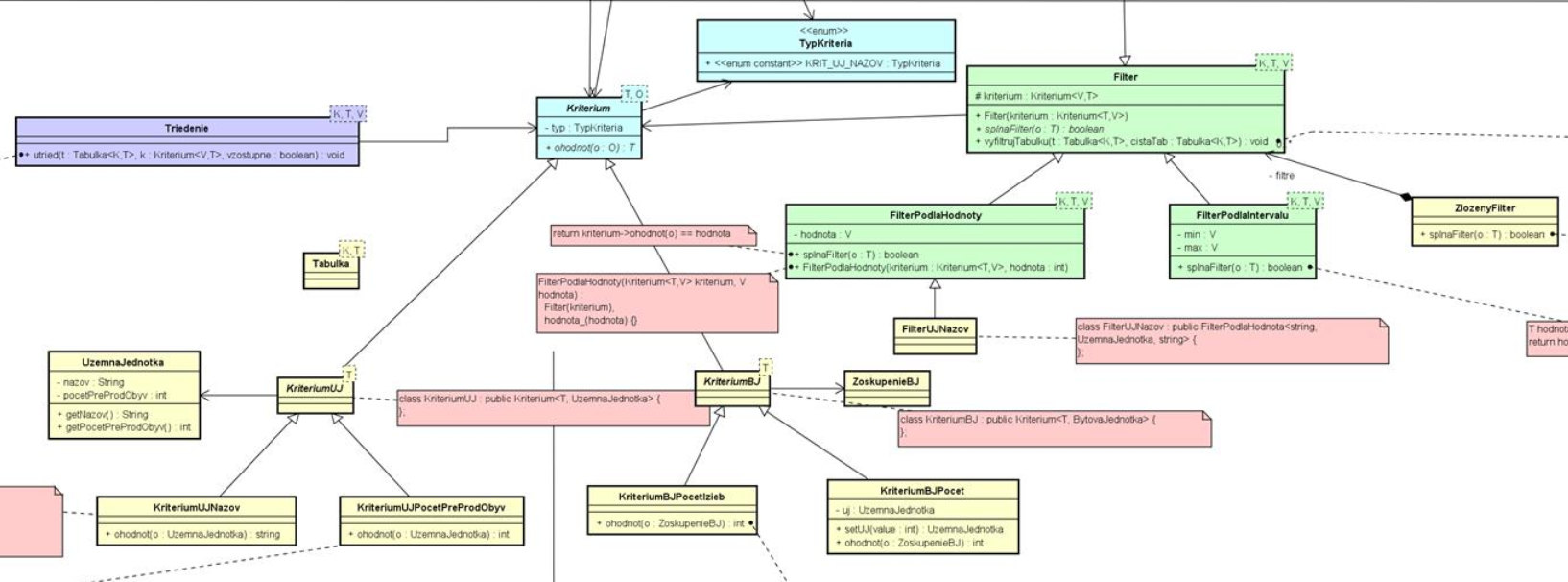
**FAKULTA RIADENIA A INFORMATIKY**

**Algoritmy a údajové štruktúry**

Semestrálna práca 2

Dávid Hrebičík

**2021**

**UML** **diagram**

Na vypracovanie aplikácie budú použité štruktúry – utriedená sekvenčná tabuľka, neutriedená sekvenčná tabuľka, pole. Utriedená sekvenčná tabuľka bude využitá na ukladanie údajov o územných jednotkách. Dôvod použitia utriedenej sekvenčnej tabuľky je pretože potrebujeme vyhľadávať územnej jednotky podľa kľúča so zložitosťou O(log n). Na triedenie sa používa algoritmus quick sort. Neutriedená sekvenčná tabuľka sa využíva na filtrovanie dát. Načítaniu dát predchádzalo odstránenie diakritiky vo všetkých CSV súboroch pomocou makra v Exceli.

Popis tried:  
Na načítavanie dát z CSV súborov je trieda Loader ktorá naloaduje dáta z CSV súborov načíta do triedy UzemnaJednotka a potom do Tabuľky. Trieda UzemnaJednotka slúži na reprezentáciu územných jednotiek s údajmi o nich a so zoznamom vyšších územných celkov. Ďalej si uchováva TypUzemnejJednotky čo je enum class na rozoznanie Obcí, Okresov, Krajov, Slovenska. Trieda Triedenie je určená na utriedenie tabuľky. Trieda ZlozenyFilter je trieda na kombináciu filtrov, obsahuje zoznam filtrov ktoré sa môžu/nemusia použiť to je zabezpečené vnútorným bool atribútom v triede Filter. Ďalej sú tam triedy Filter, FilterPodlaHodnoty, FilterPodlaIntervalu, Kriterium, KriteriumUJ, KriteriumBJ..., ktoré reprezentujú kritéria a filtre.

Popis výpočtu kritérií:

**KUJnazov** = metóda územnej jednotky – územnáJednotka.getNazov();

**KUJtyp** = metóda územnej jednotky – územnáJednotka.getTypUJ();

**KUJprislusnost** = metóda územnej jednotky -územnáJednotka.jeClenomUJ(parameter);

**KUJpocetPreproduktivnychObyvatelov** = metóda územnej jednotky – územnáJednotka.getPocetPreProObyv();

**KUJpocetProduktivnychObyvatelov** = metóda územnej jednotky – územnáJednotka.getPocetProduktObyv();

**KUJpocetPoproduktivnychObyvatelov** = metóda územnej jednotky – územnáJednotka.getPocetPoprodukObyv();

**KUJpocetObyvatelov** = súčet metód územnáJednotka.getPocetPoprodukObyv() + územnáJednotka.getPocetProduktObyv() +územnáJednotka.getPocetPreProObyv();

**KUJcelkovaVymera** = metóda územnej jednotky – územnáJednotka.getVymera();

**KUJzastavanaPlocha** = metóda územnej jednotky – územnáJednotka.getZastavPlocha();

**KUJzastavanost** = 100 \* územnáJednotka.getZastavPlocha()/ územnáJednotka.getVymera();

**KBJtyp** = metóda bytovej jednotky – bytováJednotka.getTyp();

**KBJpocetOsob** = metóda bytovej jednotky – bytováJednotka.getPocetOsob();

**KBJpocetIzieb** = metóda bytovej jednotky – bytováJednotka.getPocetIzieb();

**KBJpocet** = metóda bytovej jednotky – bytováJednotka.getPocetBytovychJednotiekVZoskupeni(územnáJednotka);

**KUJpocetBJ** = metóda bytovej jednotky – bytováJednotka.getPocetBytovychJednotiek (územnáJednotka);

Zložitosti jednotlivých vyhľadávacích kritérií:

3a – vyhľadanie a vypísanie územnej jednotky podľa názvu má zložitosť O(log n+k) pretože nájdenie záznamu má zložitosť O(log n) a údaje má územná jednotka v sebe uložené okrem zastavanosti ktorú treba vypočítať ale tam je zložitosť zanedbateľná takže vypísanie má zložitosť (k \* O(1)) kde k je počet údajov a n počet prvkov tabuľky

3b – vyhľadanie a vypísanie územnej jednotky podľa počtu obyvateľov má zložitosť O(log n+k) aj keď počet obyvateľov sa musí vypočítať pretože nájdenie záznamu má zložitosť O(log n) a údaje má územná jednotka v sebe uložené okrem zastavanosti a počtu obyvateľov ktoré treba vypočítať ale tam je zložitosť zanedbateľná takže vypísanie má zložitosť (k \* O(1)) kde k je počet údajov a n počet prvkov tabuľky

3c – vyhľadanie a vypísanie územnej jednotky podľa zastavanosti má zložitosť O(log n+k) aj keď zastavanosť sa musí vypočítať pretože nájdenie záznamu má zložitosť O(n) a údaje má územná jednotka v sebe uložené okrem zastavanosti a počtu obyvateľov ktoré treba vypočítať ale tam je zložitosť zanedbateľná takže vypísanie má zložitosť (k \* O(1)) kde k je počet údajov a n počet prvkov tabuľky

4a – zoradenie a vypísanie územných jednotiek podľa názvu má zložitosť O(n \* log n + n) pretože zoradenie jednotiek má zložitosť O(n \* log n) a údaje má územná jednotka v sebe uložené ale je tam aj zložitosť kontrolovania či územná jednotka spĺňa filtre tam je zložitosť (n \* O(1)) kde je n počet prvkov tabuľky

4b – zoradenie a vypísanie územných jednotiek podľa počtu obyvateľov má zložitosť O(n \* log n + n) pretože zoradenie jednotiek má zložitosť O(n \* log n) a údaje má územná jednotka v sebe uložené ale je tam aj zložitosť kontrolovania či územná jednotka spĺňa filtre tam je zložitosť (n \* O(1)) kde je n počet prvkov tabuľky

4c – zoradenie a vypísanie územných jednotiek podľa počtu obyvateľov má zložitosť O(n \* log n + n) pretože zoradenie jednotiek má zložitosť O(n \* log n) a údaje má územná jednotka v sebe uložené ale je tam aj zložitosť kontrolovania či územná jednotka spĺňa filtre tam je zložitosť (n \* O(1)) kde je n počet prvkov tabuľky

5 – vypísanie informácií o zoskupeniach bytových jednotiek vo zvolenej územnej jednotke má zložitosť O (n \* b) filtrovanie územných jednotiek má zložitosť O(n) vypísanie jednotlivých bytových jednotiek má zložitosť O(b) kde n je počet územných jednotiek a b je počet bytových jednotiek

6 – vypísanie sumárnych informácií o počtoch bytových jednotiek vo zvolených územných jednotkách má zložitosť O (n \* b) filtrovanie územných jednotiek má zložitosť O(n) filtrovanie jednotlivých bytových jednotiek má zložitosť O(b) pre každú územnú jednotku + výpis O(n), to znamená že celkovo O(n\*b + n) kde n je počet územných jednotiek a b je počet bytových jednotiek ostatné operácie sú zanedbateľné

7 – zoradenie územných jednotiek podľa sumárneho počtu zvolených bytových jednotiek - vyfiltrovanie územných jednotiek má zložitosť O(n) a pre každú územnú jednotku sa musia vyfiltrovať jej bytové jednotky čo nám dáva zložitosť O(f \* b) kde f je počet vyfiltrovaných územných jednotiek a b je počet bytových jednotiek. Ďalej musíme usporiadať jednotlivé územné jednotky so zložitosťou O((log f) \* f) a vypísať O(f). To nám dáva celkovú zložitosť O((log f )\*f +f \*b + f)

8 – vypísanie výberových kritérií pre zoskupenia bytových jednotiek vo zvolených územných jednotkách – vyfiltrovanie územných jednotiek má zložitosť O(n) , pre každú vyfiltrovanú územnú jednotku musíme nájsť minimum a maximum so zložitosťou O(b) a vypísať O(f) takže spolu máme zložitosť O(f\*b + f) kde f je počet vyfiltrovaných územných jednotiek

9 – vypísanie výberových kritéríi pre územné jednotky na základe zvolených zoskupení bytových jednotiek - vyfiltrovanie územných jednotiek má zložitosť O(n) vyfiltrovanie bytových O(b) pre každú vyfiltrovanú územnú jednotku dáva zložitosť O(f\*b). Ďalej musíme vyfiltrované bytové jednotky prehľadať a nájsť minimum a maximum so zložitosťou O(f\*b). Celková zložitosť je O(f\*b+ f\*b)

**Použivateľská príručka**

Na vstup a výstup sa používa konzola. Pri zadávaní filtrov je možnosť napísať “nedefinovany” čo zabezpeči že sa daný filter nebude aplikovať. Pri zadávaní čiselných filtrov sa miesto “nedefinovany” píše -1. Pri triedení je možnosť triediť zostupne alebo vzostupne taktiež na konzole pomocou príkazu “zostupne”. Implicitne sa triedi vzostupne. Pri zadávaní okresov sa píše “Okres” pred dané mesto, pre kraje sa píše iba začiatok napríklad “Zilinsky” alebo “Bratislavsky”.