Fakulta riadenia a informatiky

Informatika

Semestrálna práca S1

# Systém pre geodetov

# 2023

Ing. **Peter Jankovič,** PhD.   
PONDELOK 10, 11 Maroš Gorný, 5ZIB11

2023/2024

Obsah

[Aplikácia na spracovanie údajov zo Sčítania obyvateľov domov a bytov z roku 2021 1](#_Toc105394738)

[Návrh aplikácie z pohľadu použitých údajových štruktúr 5](#_Toc105394739)

[Uloženie územných jednotiek 5](#_Toc105394740)

[Diagram tried a ich popis 6](#_Toc105394741)

[Data 6](#_Toc105394742)

[UzemnaJednotka – Balíček 7](#_Toc105394743)

[Kriterium – Balíček 8](#_Toc105394744)

[Filter – Balíček 11](#_Toc105394745)

[Triedenie – Balíček 13](#_Toc105394746)

[VyberoveKriteria – Balíček 15](#_Toc105394747)

[15](#_Toc105394748)

[Enumy 16](#_Toc105394749)

[Popis výpočtu kritérií 17](#_Toc105394750)

[Zložitosť operácií 19](#_Toc105394751)

[Načítanie dát 19](#_Toc105394752)

[Kritéria 19](#_Toc105394753)

[Filtre 21](#_Toc105394754)

[Triedenie 21](#_Toc105394755)

[Vyhľadávanie najlepších 21](#_Toc105394756)

[Vypísanie údajov 21](#_Toc105394757)

[Používateľská príručka 21](#_Toc105394758)

## Návrh aplikácie z pohľadu použitých údajových štruktúr

### Uloženie parciel a nehnutelností

Údajové štuktúry ktoré som použil boli quad stromy, binárný vyhľadaváci strom a taktiež už implementované listy a zásobniky v programovacom jazyku C#. Vzhľadom na to, že sme potrebovali unikátnosť záznamov, vytvorili sme objekty v strome, ktoré niesli dvojicu   
kľúč – dáta. Parcely a nehnuteľnosti sa do stromu pridávali na základe dvoch pozicí a to najviac ľavý dolný roh a najviac horný pravý roh, ktore presne vymedzili priestor parcely alebo nehnuteľnosti. V strome teda vieme vyhľadávať pomocou daných súradníc.

#### Quad strom

V aplikácií sme urobili tri quad stromy, jeden pre parcely, jeden pre nehnuteľnosti a jeden spoločný, kde sme pridávali aj parcely aj nehnuteľnosti. Vďaka tomuto rozdeleniu, sme mohli efektívnejšie vyhľadávať na základe typu objektu, buď parcely alebo nehnutelnosti.

Spoločný strom nám pomáha pri vyhľadávani objektu podľa pozície, keď nám nezáleží na jeho type.

Tieto výhody však priniesli aj mierne zhoršenie komplexity pri vymazávaní, pretože vždy musíme objekt vymazat z dvoch stromov. Dopredu však vieme, že operácia vymazať sa bude vykonávať minimálne, preto sme boli ochotní podstúpiť toto mierne zhoršenie.

#### Binárny vyhľadávací strom

Pre potreby obmedzenia duplicitnosť klúčov, si quad strom pamätá všetky kľúče ktoré boli vložené do stromu a to práve v nami implementovanom binárnom vyhľadávacom strome. Na základe tejto informácie, vieme ešte pred vložením, pripadne vymazaním objektu zistiť, či je daný kľúč v strome, alebo nie.

Nakoľko je veľká pravdepodobnosť, že sa nové objekty budú vkladať postupne od najmenšieho kľúču po najväčší, alebo od naväčšieho kľúču po najmenší, môže sa stať, že náš strom zdegeneruje do lineárneho zoznamu. Preto by bolo v budúcnosti vhodné tento strom zmeniť napríklad na Treap, ktorý má oveľa menšiu šancu zdegenerovať na lineárny zoznam.

#### Pole a zásobník

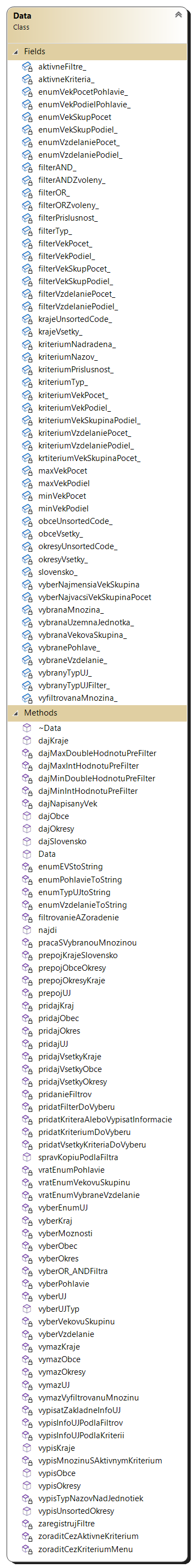
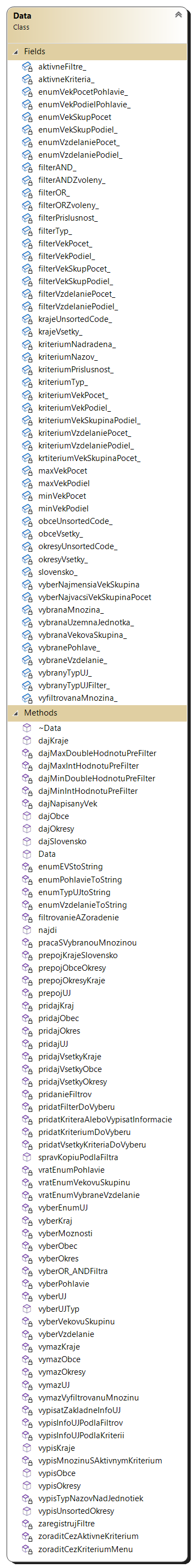
Polia a zásobníky sme si neimplementovali a využili sme už vopred zadefinované implementácie jazyku C#.

Polia sme využívali väčšinou len ako úložný priestor a väčšinou sme v konečnom dôsledku museli prejsť v najhoršom prípade všetky objekty ktoré v ňom boli.

Zásobníky sme využili v prípadoch prehliadky, či už stromov, alebo podstromov.

## Diagram tried a ich popis

### Data

Trieda určená hlavne na logiku pri používaní aplikácie teda na use cases. Taktiež vytvára územné jednotky, prepája ich a pomocou tejto triedy ich aj filtrujem, hľadám aj triedim. 

### UzemnaJednotka – Balíček

Graphical user interface, application

Description automatically generated

#### UzemnaJednotka

Územná jednotka je abstraktná trieda v ktorej sú všetky potrebné informácie ktoré má územná jednotka mať, teda meno, kód, počet obyvateľov, typ, nad jednotku a či patrí do vyššieho celku alebo nie.

Taktiež má gettre na svoje atribúty a vie spočítať obyvateľov, vek ľudí v danej územnej jednotke, poprípade počet podľa vzdelania alebo zaradenia do ekonomickej vekovej skupiny.

#### VyssiCelok

Vyšší celok je potomok od UzemnaJednotka a taktiež je to abstraktná trieda. Avšak vyšší celok si pamätá pod celok a má prekryté nejaké metódy.

#### Slovensko

Trieda je potomok vyššieho celku a je spravená aby som mal všetky územné jednotky v jednom celku.

#### Kraj

Trieda je potomok vyššieho celku, vie do seba pridať okresy, ale taktiež si pamätá všetky obce ktoré sú pod ňou. + všetky zdedené metódy.

#### Okres

Trieda je potomok vyššieho celku, vie do seba pridať obce a nastaviť nad kraj. + všetky zdedené metódy.

#### Obec

Trieda je potomok územnej jednotky a pamätá si vzdelanie a vek ktoré sú uložené v poli.  
Taktiež si vie zo seba vytiahnuť všetky dôležité údaje.

### Kriterium – Balíček

Graphical user interface, application

Description automatically generated

#### Kriterium<R>

Kritérium je abstraktná trieda so šablónovým parametrom, ktorý je návratová hodnota kritéria. Táto trieda má jednu metódu a to urobKriterium, ktoré vráti hodnotu podľa územnej jednotky podľa kritéria.

#### KUJPrislusnost

Potomok triedy Kriterium a vracia bool hodnotu, či sa územná jednotka nachádza vo vyššom celku alebo nie. Túto hodnotu si vyberie ako getter z územnej jednotky.

#### KNazov

Potomok triedy Kriterium a vracia názov územnej jednotky pod stringom. Túto hodnotu si vyberie ako getter z územnej jednotky.

#### KUJVekPocet

Potomok triedy Kriterium a vracia počet obyvateľov v danom vekovom rozmedzí a danom pohlaví. Je teda možné nastaviť interval veku a pohlavie(muž, žena, obidve). Túto hodnotu vypočíta pomocou použitia funkcie getVek v územnej jednotke.

#### KUJVekPodiel

Potomok triedy Kriterium a vracia podiel obyvateľov v danom vekovom rozmedzí a danom pohlaví. Je teda možné nastaviť interval veku a pohlavie(muž, žena, obidve). Túto hodnotu vypočíta pomocou použitia funkcie getVek v územnej jednotke ktorú predelí počtom obyvateľov danej územnej jednotky ktorý získa cez getter a vynásobí stovkou.

#### KUJVekovaSkupinaPocet

Potomok triedy Kriterium a vracia počet obyvateľov v danej ekonomickej vekovej skupine. Je teda možné nastaviť ekonomickú vekovú skupinu. Túto hodnotu vypočíta pomocou použitia funkcie getPocetVekovejSkupiny v územnej jednotke.

#### KUJVekovaSkupinaPodiel

Potomok triedy Kriterium a vracia podiel obyvateľov v danej ekonomickej vekovej skupine. Je teda možné nastaviť ekonomickú vekovú skupinu. Túto hodnotu vypočíta pomocou použitia funkcie getPocetVekovejSkupiny v územnej jednotke ktorú vydelí s počtom obyvateľov danej územnej jednotky a vynásobí stovkou.

#### KUJNadradena

Potomok triedy Kriterium a vracia územnú jednotku ktorá je nad jednotkou danej územnej jednotky. Túto hodnotu si vyberie ako getter z územnej jednotky.

#### KUJTyp

Potomok triedy Kriterium a vracia enum typ územnej jednotky danej územnej jednotky. Túto hodnotu si vyberie ako getter z územnej jednotky.

#### KUJVzdelaniePocet

Potomok triedy Kriterium a vracia počet obyvateľov s daným vzdelaním. Je teda možné nastaviť vzdelanie. Túto hodnotu vypočíta pomocou použitia funkcie getPocetVzdelanie v územnej jednotke.

#### KUJVzdelaniePodiel

Potomok triedy Kriterium a vracia podiel obyvateľov s daným vzdelaním. Je teda možné nastaviť vzdelanie. Túto hodnotu vypočíta pomocou použitia funkcie getPocetVzdelanie v územnej jednotke ktorú predelí počtom obyvateľov danej územnej jednotky ktorý získa cez getter a vynásobí stovkou.

### Filter – Balíček

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

#### Filter

Abstraktná trieda všetkých filtrov. Vyhodnocuje, či daná územná jednotka spĺňa filter alebo nie.

#### FilterKriterium<R>

Abstraktná trieda, ktorá je potomkom Filtra. Pamätá si kritérium so šablónovou návratovou hodnotou. podľa ktorého vyhodnocuje či filter danú/é hodnotu/y spĺňa, alebo nie.

#### FilterMaleFi<R>

Abstraktná trieda, ktorá je potomkom FilterKriterium. Vyhodnocuje podľa zadaného kritéria, či daná hodnota spĺňa filter alebo nie.

#### FilterVelkeFi<R>

Abstraktná trieda, ktorá je potomkom FilterKriterium. Vyhodnocuje podľa zadaného kritéria, či sa hodnota nachádza v intervaly alebo nie.

#### KompozitnyFilter

Trieda pomocou ktorej môžeme používať viac filtrov naraz, je potomkom Filtra. Viem tam filtre zaregistrovať, ale aj vymazať. Všetky filtre si pamätá v poli.

#### Filter\_AND

Trieda je potomkom KompozitnyFilter a vyhodnocuje či všetky filtre spĺňajú podmienku. Ak jeden filter nespĺňa, vracia false.

#### Filter\_OR

Trieda je potomkom KompozitnyFilter a vyhodnocuje či aspoň jeden filter spĺňa podmienku. Ak aspoň jeden filter spĺňa, vracia true, inak false.

#### FilterVzdelaniePocet

Trieda je potomkom FilterMaleFi a používa kritérium KUJVzdelaniePocet, kde sa môže nastaviť vzdelanie.

#### FilterPrislusnost

Trieda je potomkom FilterMaleFi a používa kritérium KUJPrislusnost.

#### FilterTyp

Trieda je potomkom FilterMaleFi a používa kritérium KUJTyp.

#### FilterVekovaSkupinaPodielMedzi

Trieda je potomkom FilterVelkeFi a používa kritérium KUJVEkovaSkupinaPodiel. Viem si v ňom nastaviť rozmedzie a taktiež ekonomickú vekovú skupinu.

#### FilterVekovaSkupinaPocetMedzi

Trieda je potomkom FilterVelkeFi a používa kritérium KUJVEkovaSkupinaPocet. Viem si v ňom nastaviť rozmedzie a taktiež ekonomickú vekovú skupinu

#### FilterVekPocetMedzi

Trieda je potomkom FilterVelkeFi a používa kritérium KUJVekPocet. Viem si v ňom nastaviť vekové rozmedzie ale taktiež rozmedzie počtu a pohlavie.

#### FilterVekPodielMedzi

Trieda je potomkom FilterVelkeFi a používa kritérium KUJVekPodiel. Viem si v ňom nastaviť vekové rozmedzie ale taktiež rozmedzie podielu a pohlavie.

#### FilterVzdelaniePocetMedzi

Trieda je potomkom FilterVelkeFi a používa kritérium KUJVzdelaniePocet. Viem si v ňom nastaviť dané vzdelanie a rozmedzie počtu.

#### FilterVzdelaniePodielMedzi

Trieda je potomkom FilterVelkeFi a používa kritérium KUJVzdelaniePocet. Viem si v ňom nastaviť dané vzdelanie a rozmedzie podielu.

### Triedenie – Balíček

Graphical user interface, text, application

Description automatically generated

#### Sort<K,T,R>

Abstraktná trieda so šablónovými parametrami, K – kľúč v tabuľke, T – dáta v tabuľke, R – návratový typ pre kritérium. Daná trieda triedi tabuľku, buď podľa kritéria alebo bez neho podľa kľúča.

#### QuickSort<K,T,R>

Potomok triedy Sort. Je to daná inštancia triedy Sort, takže robí presne to isté, akurát je to už presne zvolený spôsob triedenia.

### VyberoveKriteria – Balíček

### Diagram Description automatically generated

#### VyberoveKriterium<R>

Abstraktná trieda so šablónovým parametrom R – ktorý slúži ako návratová hodnota kritéria. Taktiež daná trieda vie vybrať najlepšiu/najhoršiu hodnotu podľa kritéria z danej tabuľky.

#### VyberoveKriteriumMax<R>

Potomok triedy VyberoveKriterium. Trieda hľadá najväčšiu hodnotu podľa kritéria.

#### VyberoveKriteriumMin<R>

Potomok triedy VyberoveKriterium. Trieda hľadá najmenšiu hodnotu podľa kritéria.

#### VNajvacsiaVekovaSkupina

Potomok triedy VyberoveKriteriumMax. Trieda hľadá najväčšiu hodnotu podľa vopred zvoleného kritéria KUJVekovaSkupinaPocet a vopred zvolenú ekonomickú vekovú skupinu.

#### VNajmensiaVekovaSkupina

Potomok triedy VyberoveKriteriumMin. Trieda hľadá najmenšiu hodnotu podľa vopred zvoleného kritéria KUJVekovaSkupinaPocet a vopred zvolenú ekonomickú vekovú skupinu.

### Enumy

Enum slúži na lepšiu prehľadnosť podľa nejakých častí.

#### EnumVekovaSkupina

* PREDPRODUKTIVNI
* PRODUKTIVNI
* POPRODUKTIVNI
* NULL\_ENUM\_VEKOVA\_SKUPINA

#### EumVzdelanie

* BEZ\_UKONCENEHO\_VZDELANIA\_0\_14\_ROKOV
* ZAKLADNE\_VZDELANIE
* STREDNE\_ODBORNE\_VZDELANIE
* UPLNE\_STREDNE\_VZDELANIE
* VYSSIE\_ODBORNE\_VZDELANIE
* VYSOKOSKOLSKE\_VZDELANIE
* BEZ\_SKOLSKEHO\_VZDELANIA\_15\_A\_VIAC\_ROKOV
* NEZISTENE
* NULL\_ENUM\_VZDELANIE

#### EnumTypUzemnejJednotky

* SLOVENSKO
* KRAJ
* OKRES
* OBEC
* NULL\_ENUM\_UJ

#### EnumPohlavie

* ZENA
* MUZ
* OBIDVE\_POHLAVIA
* NULL\_ENUM\_POHLAVE

## Popis výpočtu kritérií

#### KUJPrislusnost

Kritérium vráti atribút územnej jednotky ktorý sa jej nastaví hneď pri vytvorení.

#### KNazov

Kritérium vráti atribút územnej jednotky ktorý sa jej nastaví hneď pri vytvorení.

#### KUJVekPocet

Kritérium vráti hodnotu pomocou funkcie územnej jednotky getVek().

Táto funkcia je implementovaná inak vo Vyššom celku a inak v Obci.

V obci vráti vek z poľa ktoré má uložené ako atribút. Ak je to interval, tak dané položky spočíta. Pri pohlaví sa mení len pozícia v poli. Ak je vek väčší ako 100, automatický sa nastaví na 100 a ak je menší ako 0, nastaví sa na 0.

Vo vyššom celku sa táto funkcia zavolá v každom objekte v svojom pod celku a súčet sa spočíta a vráti.

#### KUJVekPodiel

Kritérium vráti hodnotu pomocou funkcie územnej jednotky getVek(), ktorá je vydelená počtom obyvateľov a vynásobená stovkou.

getVek() funkcia je implementovaná inak vo Vyššom celku a inak v Obci.

V obci vráti vek z poľa ktoré má uložené ako atribút. Ak je to interval, tak dané položky spočíta. Pri pohlaví sa mení len pozícia v poli. Ak je vek väčší ako 100, automatický sa nastaví na 100 a ak je menší ako 0, nastaví sa na 0.

Vo vyššom celku sa táto funkcia zavolá v každom objekte v svojom pod celku a súčet sa spočíta a vráti.

Počet obyvateľov danej územnej jednotky sa vráti cez atribút ktorý je nastavený hneď po vzniku územnej jednotky.

#### KUJVekovaSkupinaPocet

Kritérium vráti hodnotu pomocou funkcie územnej jednotky getPocetVekovejSkupiny(), ktorá ale potrebujem enum a to typ vekovej skupiny.

Táto funkcia je implementovaná inak vo Vyššom celku a inak v Obci.

V obci vráti súčet obyvateľov s vekom v danej EVS. Vo vyššom celku sa už len zavolá táto metóda pre každého člena v pod celku a hodnoty sa spočítajú.

#### KUJVekovaSkupinaPodiel

Kritérium vráti hodnotu pomocou funkcie územnej jednotky getPocetVekovejSkupiny(), ktorá ale potrebujem enum a to typ vekovej skupiny, následne sa vydelí počtom obyvateľov a vynásobí sa stovkou.

Vo vyššom celku sa táto funkcia zavolá v každom objekte v svojom pod celku a súčet sa spočíta a vráti.

Počet obyvateľov danej územnej jednotky sa vráti cez atribút ktorý je nastavený hneď po vzniku územnej jednotky.

#### KUJNadradena

Kritérium vráti atribút územnej jednotky ktorý sa jej nastaví hneď pri vytvorení.

#### KUJTyp

Kritérium vráti atribút územnej jednotky ktorý sa jej nastaví hneď pri vytvorení.

#### KUJVzdelaniePocet

Kritérium vráti hodnotu pomocou funkcie územnej jednotky getPocetVzdelania(), ktorá ale potrebuje enum, typ vzdelania.

Táto funkcia je implementovaná inak vo Vyššom celku a inak v Obci.

V obci vráti počet z poľa ktoré má uložené ako atribút.

Vo vyššom celku sa táto funkcia zavolá v každom objekte v svojom pod celku a súčet sa spočíta a vráti.

#### KUJVzdelaniePodiel

Kritérium vráti hodnotu pomocou funkcie územnej jednotky getPocetVzdelania(), ktorá ale potrebuje enum, typ vzdelania, táto funkciu sa následne vydelí počtom obyvateľov a vynásobí sa stovkou.

Táto funkcia je implementovaná inak vo Vyššom celku a inak v Obci.

V obci vráti počet z poľa ktoré má uložené ako atribút.

Vo vyššom celku sa táto funkcia zavolá v každom objekte v svojom pod celku a súčet sa spočíta a vráti.

Počet obyvateľov danej územnej jednotky sa vráti cez atribút ktorý je nastavený hneď po vzniku územnej jednotky.

## Zložitosť operácií

### Načítanie dát

#### Vloženie obcí

Do utriedenej aj neutriedenej sekvenčnej tabuľky spolu – O(N + log N + V + Z + S)

* N – počet obcí
* V – počet políčok pre vek
* Z – počet políčok pre typ vzdelania
* S – počet políčok pre vek (spočítavanie obyvateľov)

#### Vloženie okresov

Do utriedenej aj neutriedenej sekvenčnej tabuľky spolu– O(N + log N)

* N – počet okresov

#### Vloženie krajov

Do utriedenej aj neutriedenej sekvenčnej tabuľky spolu – O(N + log N)

* N – počet krajov

#### Prepojenie obcí a okresov

O(2\*N + log M )

* N – počet obcí, spočítanie obyvateľov obcí
* M – počet okresov

#### Prepojenie okresov a krajov

O(2\*N + log M )

* N – počet okresov, spočítanie obyvateľov okresov
* M – počet krajov

#### Prepojenie krajov a Slovenska

O(2\*N + M )

* N – počet krajov, spočítanie obyvateľov krajov
* M – počet obci, ktoré si kraj bude pamätať

### Kritéria

#### KUJPrislusnost

O(1)

#### KNazov

O(1)

#### KUJVekPocet

O(R + N)

* R – Vekové rozmedzie (0 až 9 = 10)
* N – počet pod jednotiek

#### KUJVekPodiel

O(R + N + 1)

* R – Vekové rozmedzie
* N – Počet pod jednotiek

#### KUJVekovaSkupinaPocet

O(R + N)

* R – Vekové rozmedzie
  + Predproduktívni = 15
  + Produktívni = 50
  + Poproduktívni = 35
* N – Počet pod jednotiek

#### KUJVekovaSkupinaPodiel

O(R + N + 1)

* R – Vekové rozmedzie
  + Predproduktívni = 15
  + Produktívni = 50
  + Poproduktívni = 35
* N – Počet pod jednotiek

#### KUJNadradena

O(1)

#### KUJTyp

O(1)

#### KUJVzdelaniePocet

O(1)

#### KUJVzdelaniePodiel

O(1 + N + 1)

* N – Počet pod jednotiek

### Filtre

O(N + O(Kriterium))

* N – Počet filtrovaných prvkov

### Triedenie

O(log N)

* N – Počet triedených prvkov

### Vyhľadávanie najlepších

O(N)

* N – Počet prvkov v množine

### Vypísanie údajov

O(N + O(Kriterium) + k)

* N – Počet územných jednotiek ktoré sa majú vypísať
* Kriterium – Kritérium podľa ktorého sa vypíšu údaje
* k – počet kritérií.

## Používateľská príručka

Aplikácia vždy ukáže, aké má používateľ možnosti. Niekedy je treba na konzolu zadať znak, poprípade stačí stlačiť tlačidlo. Ak aplikácia nepovie, že má používateľ niečo stlačiť, čaká na textový vstup. Pri výbere danej obce/okresu/kraja treba napísať správne celý názov územnej jednotky. Pre pomoc sa aplikácia spýta, či chce používateľ vypísať všetky možné územné jednotky s ktorými sa dá pracovať a následne používateľ môže odpísať presný názov územnej jednotky. Pokiaľ sa chce používateľ vrátiť späť, väčšinou platí, že musí napísať charakter „q“ a stlačiť enter.

Pre potvrdenie napísaného textu musí používateľ enter stlačiť dva krát. Ak ho aplikácia vyzýva na stlačenie nejakého tlačidla, ale používateľ chce údaje zadať cez textový vstup, stačí stlačiť ľubovoľné tlačidlo, odporúča sa ale vždy stláčať enter.