

Fakulta riadenia a informatiky  
Informatika

Rozpracovanie semestrálnej pracej  
*Semestrálna práca AUS*

doc. Ing. **Miroslav Kvaščay** PhD.  
STREDA 12, 13  
2021/2022

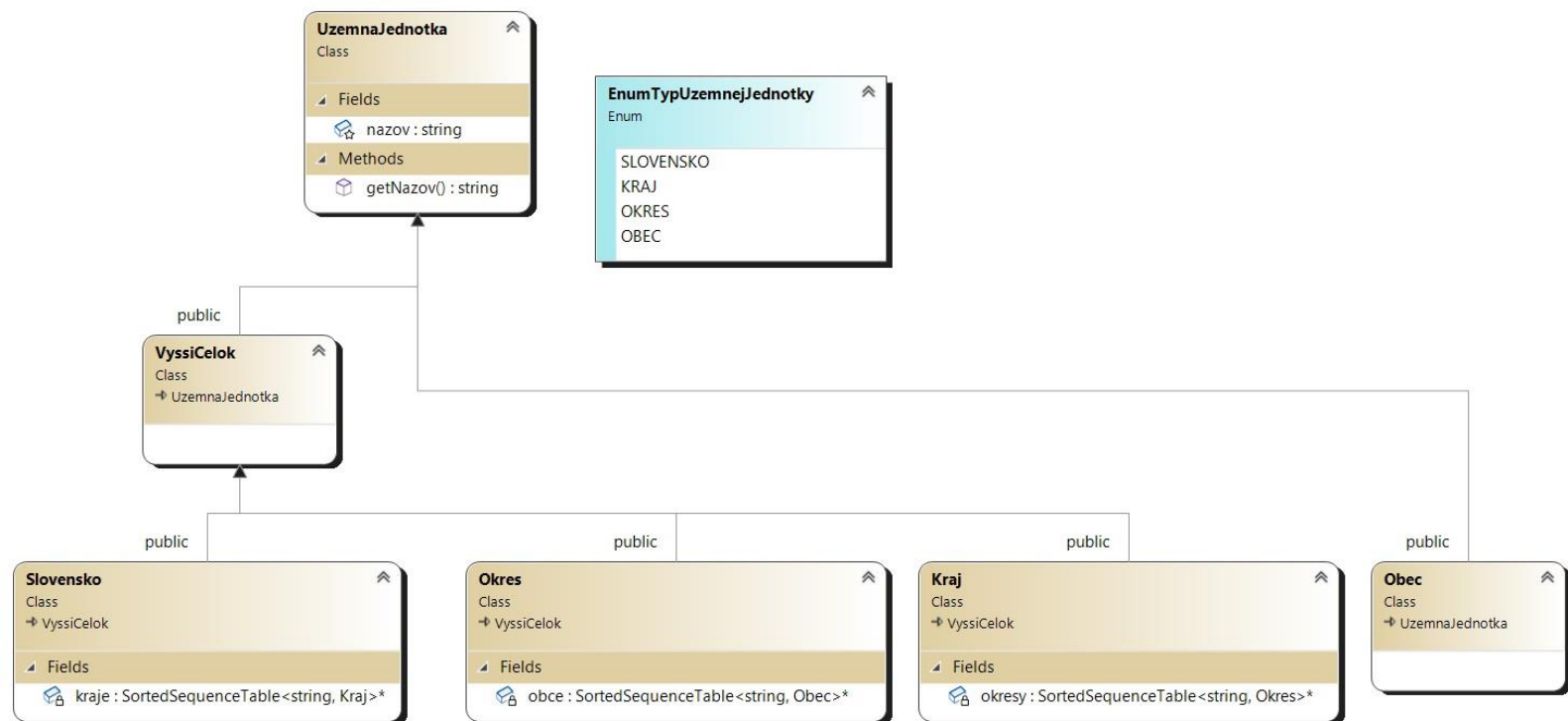
Maroš Gorný, 5ZYI21

## Obsah

Semestrálna práca .....	1
Balíčky a triedy .....	3
UzemnaJednotka .....	3
Pohlavie .....	4
Vzdelanie .....	5
EkonomickaVekovaSkupina .....	5
Filter .....	6
Kriterium .....	7
Triedenie .....	9
Structures .....	9
Použité údajové štruktúry .....	10
Array .....	10
SequenceTable .....	10
SortedSequenceTable .....	10
UnsortedSequenceTable .....	10
Úroveň 1 – Vypísanie informácií o obciach .....	11
Úroveň 2 – Zoradenie obcí .....	11
Zoradenie obcí podľa názvu .....	11
Úroveň 3 – Vypísanie informácií o vzdelaní v územných jednotkách a ich zoradenie ..	12

## Balíčky a triedy

### UzemnaJednotka



### EnumTypUzemnejJednotky

- Enum pre typ územnej jednotky

### UzemnaJednotka

- Abstraktná trieda pre územné jednotky

### VyssiCelok

- Abstraktná trieda pre vyšší celok územných jednotiek

### Slovensko

- Územná jednotka

### Kraj

- Územná jednotka

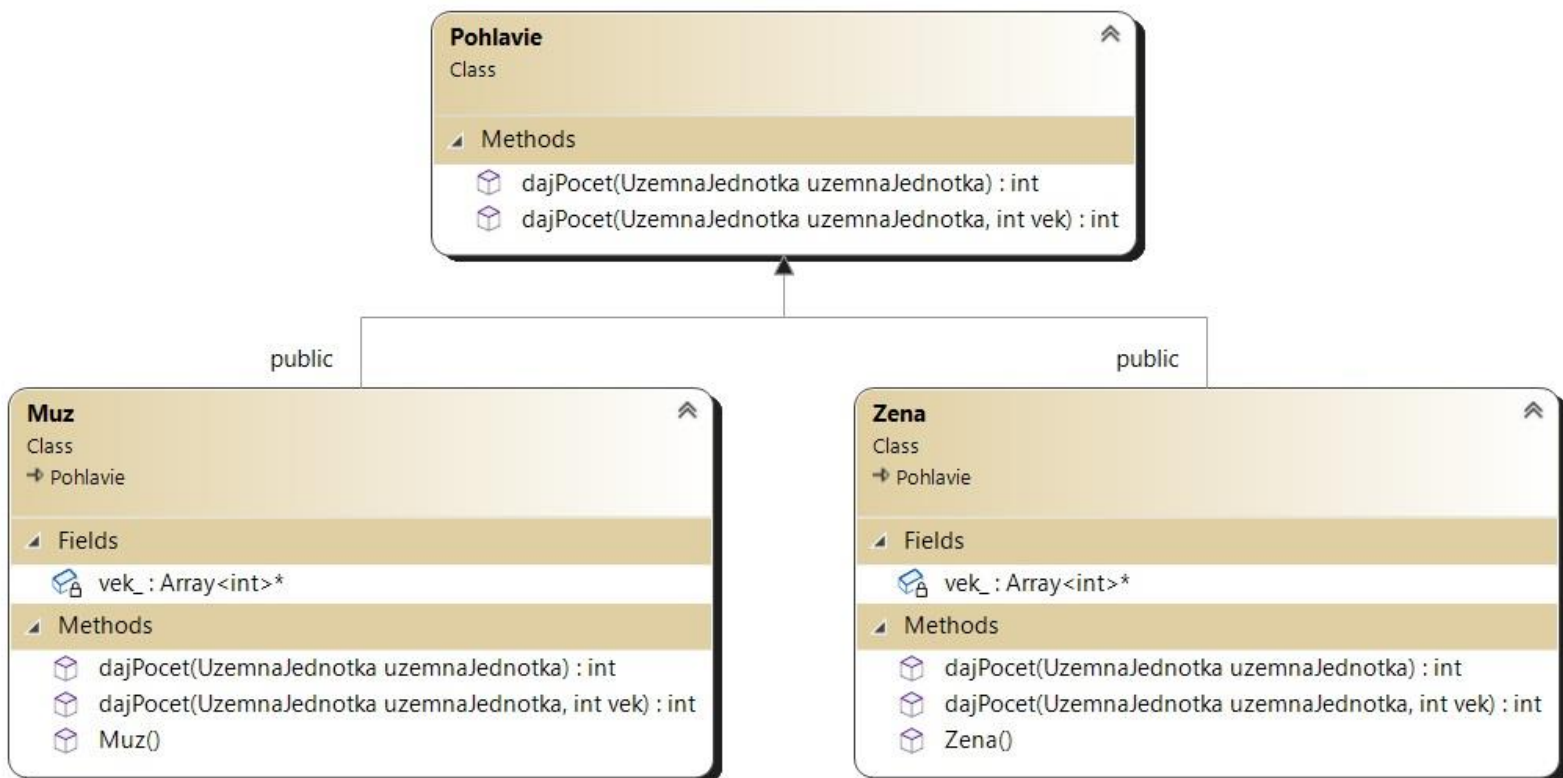
### Okres

- Územná jednotka

### Obec

- Územná jednotka

## Pohlavie



### Pohlavie

- Abstraktná trieda pre pohlavie

### Muz

- Trieda muž, v ktorej je uložený vek pre danú územnú jednotku

### Zena

- Trieda žena, v ktorej je uložený vek pre danú územnú jednotku

## Vzdelanie

**Vzdelanie**

Class

Fields

typyVzdelania : SortedSequenceTable<string, int>\*

Methods

Vzdelanie(UzemnaJednotka uzemnaJednotka)

### Vzdelanie

- Trieda vzdelanie, v ktorej si na základe územnej jednotky získam vzdelanie.

## EkonomickaVekovaSkupina

**EVS**

Class

Fields

muzi : Muz\*

zeny : Zena\*

Methods

dajPoproductivnych() : UnsortedSequenceTable<int, int>

dajPredproduktivnych() : UnsortedSequenceTable<int, int>

dajProduktivnych() : UnsortedSequenceTable<int, int>

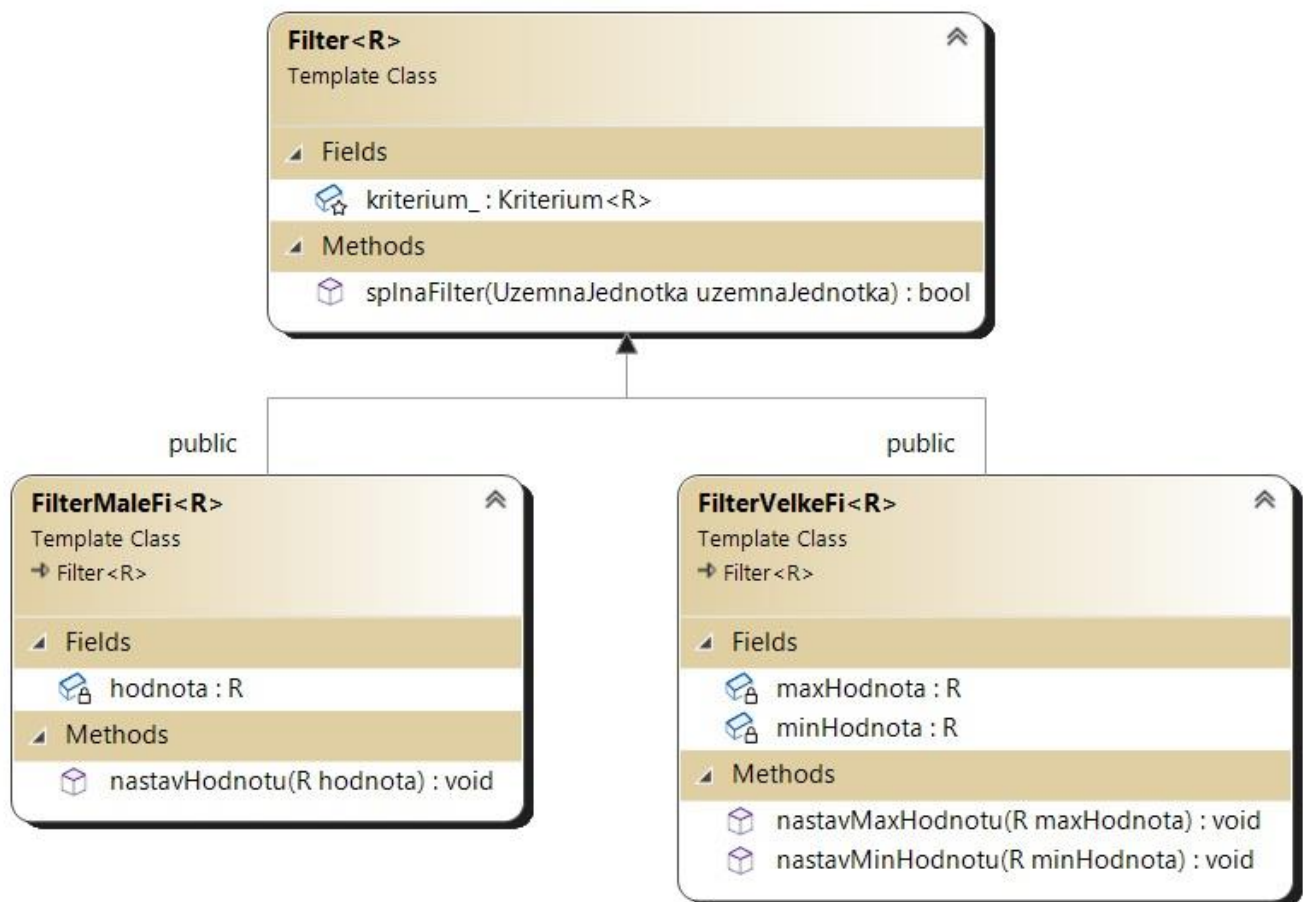
EVS(UzemnaJednotka uzemnaJednotka)

nastavEVS(EnumPohlavie pohlavie) : void

### EVS

- Trieda EVS, v ktorej môžem zisťovať vekovú skupinu

## Filter



### Filter

- Abstraktná trieda pre filter, ktorý používa kritérium a vracia hodnotu false alebo true.

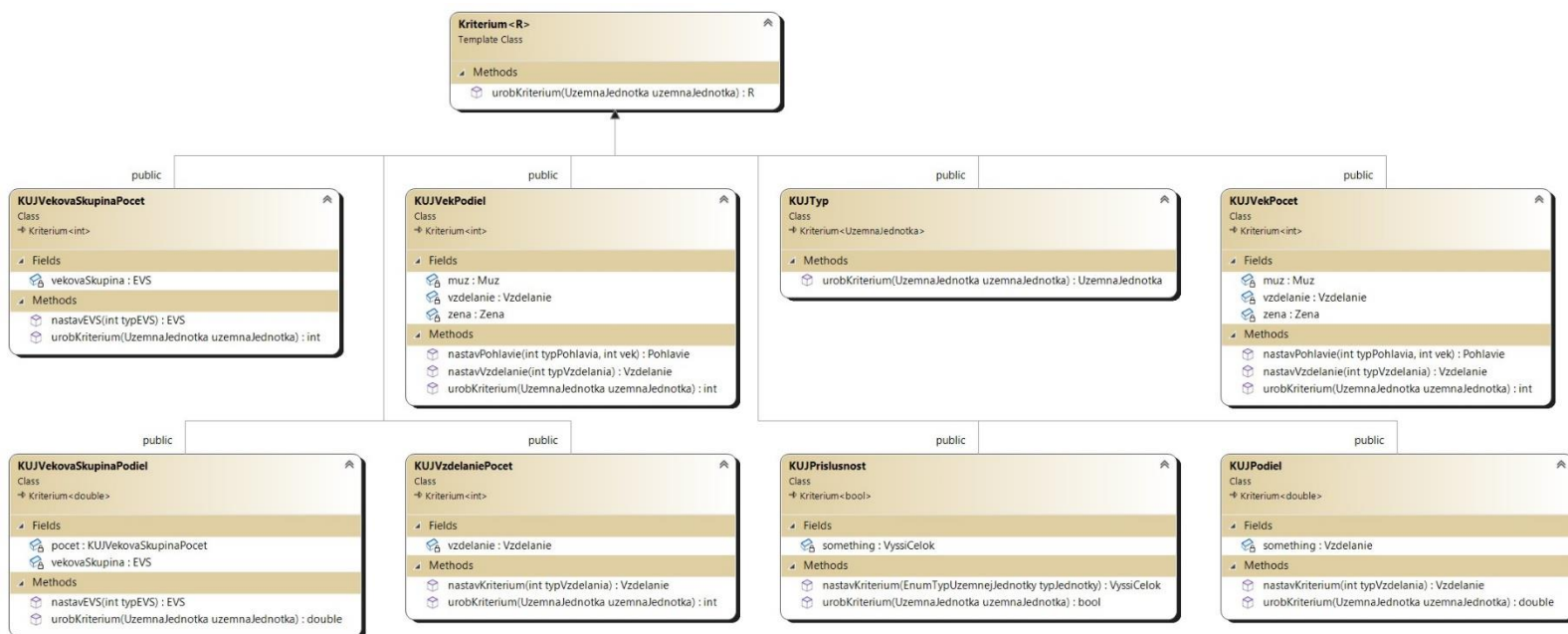
### FilterMaleFi

- Trieda filter malé Fi, kde sa filtruje podľa hodnoty.

### FilterVelkeFi

- Trieda filter veľké Fi, kde sa filtruje na základe intervalu.

## Kriterium



## Kriterium

- Abstraktná trieda kritérium so šablónovou návratovou hodnotou.

## KUJNadradena

- Zložitosť  $O(1)$ 
  - o Ešte je potrebný doplniť atribút, ale každá územná jednotka si bude pamätať svojho predka.

## KUJPrislusnost

- Zložitosť  $O(1)$ 
  - o Vzhľadom na to, že sa viem spýtať, že akého mám predka, viem rýchlo zistiť, či sa nachádzam vo vyššom celku.

## KUJTyp

- Zložitosť  $O(1)$ 
  - o Vďaka typu objektu viem zistiť aj daný typ.

## KUJVekovaSkupinaPocet

- Zložitosť  $O(\log n)$ , kde  $n$  je veľkosť prvkov v pohlaví (muž/žena/obidve)

## KUJVekovaSkupinaPodiel

- Zložitosť  $O(1) + O(\log n)$ , kde  $n$  je veľkosť prvkov v pohlaví
  - o Do triedy obec si naimplementujem aj atribút počet obyvateľov.

### KUJVekPocet

- Zložitosť  $O(\log n)$ , kde  $n$  je veľkosť prvkov v pohlaví(muž/žena/obidve)
  - o Do triedy obec si naimplementujem aj atribút počet obyvateľov.

### KUJVekPodiel

- Zložitosť  $O(1) + O(\log n)$ , kde  $n$  je veľkosť prvkov v pohlaví.
  - o Do triedy obec si naimplementujem aj atribút počet obyvateľov.

### KUJVzdelaniePocet

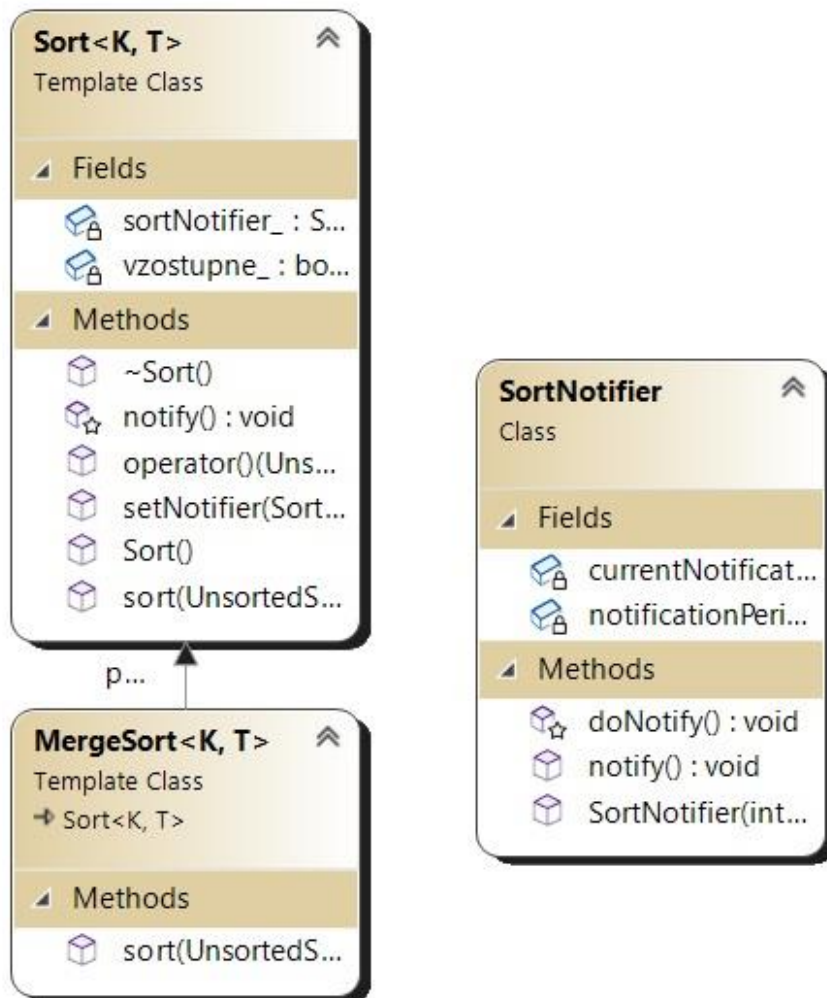
- Zložitosť  $O(\log n)$ , kde  $n$  je veľkosť prvkov vo vzdelaní

### KUJVzdelaniePodiel

- Zložitosť  $O(1) + O(\log n)$ , kde  $n$  je veľkosť prvkov vo vzdelaní
  - o Do triedy obec si naimplementujem aj atribút počet obyvateľov.



## Triedenie



Sort

Merge\_sort

## Structures

Štruktúry vytvorené na hodinách a v domáciach zadaniach

## Použité údajové štruktúry

### Array

Pole je použité v triede Muz a Zena. Je veľkosti 101, kde bude ukladaný daný vek pre muža alebo ženu. Prvok na nulte pozícii mi dá počet mužov/žien s vekom 0, prvok na 100-tej pozícii mi dá počet mužov/žien s vekom 100+.

### Zložitosť

Vlož prvok na danú pozíciu -  $O(1)$

Daj prvok na danej pozícii -  $O(1)$

### SequenceTable

Sekvenčná tabuľka je v mojom návrhu použitá častejšie. V triede Muz a Zena môžem používať kľúč ako index, teda nepotrebujem tabuľku. Na ostatné triedy v územnej jednotke už kľúče ako index použiť nemôžem, preto budem potrebovať tabuľku do ktorej môžem vkladať ako kľúč hociký objekt alebo prvok.

### SortedSequenceTable

Utriedenú sekvenčnú tabuľku v mojom návrhu budem používať pri vytvorení hierarchie pre územné jednotky. Je použitá tam, kde prvok musím hľadať častejšie, teda pre územné jednotky. Taktiež by sa vďaka bisekcii dala využiť na rýchlejšie vkladanie do tabuľky keď je na danú tabuľku aplikovaný filter.

### Zložitosť

Vlož prvok –  $O(n)$

Nájdí prvok –  $O(\log n)$

### UnsortedSequenceTable

Neutriedenú sekvenčnú tabuľku v mojom návrhu budem používať pri triedení, teda sortovaní prvkov. Jeho jednotkovú zložitosť vkladania prvkov využijem na to, že rýchlo vytvorím tabuľku ktorú aj tak následne musím pretriediť. Bolo by teda zbytočné používať utriedenú sekvenčnú tabuľku.

### Zložitosť

Vlož prvok –  $O(1)$

Nájdí prvok –  $O(n)$

## Úroveň 1 – Vypísanie informácií o obciach

Abstraktná trieda `UzemnaJednotka` má atribút `nazov`. Trieda `obec` je zdedená z tejto abstraktnej triedy a preto má tento atribút tiež. Objekt `Obec` bude vytvorený tak, že vytiahne z CSV súboru `obec` a do svojho atribútu napíše meno obce z CSV súboru.

`Slovensko` má v sebe pointer na utriedenú tabuľku, v ktorej sú kraje.

`Kraj` má v sebe pointer na utriedenú tabuľku, v ktorej sú okresy.

`Okres` má v sebe pointer na utriedenú tabuľku, v ktorej sú obce.

Tým pádom, sa viem dostať z jednej územnej jednotky na hocikakú obec, ktorá je pod danou územnou jednotkou.

Ak by som chcel napríklad počet daného typu vzdelania, využijem na to kritérium `KUJVzdelaniePocet`, ktorý si zoberie danú územnú jednotku a vráti počet daného typu vzdelanie pre danú územnú jednotku. Takto budú fungovať všetky kritéria.

Ak by som chcel použiť filter, znova musím zadať danú územnú jednotku a použiť príslušné kritérium. Ak by som chcel filtrovať podľa intervalu, musím použiť `FilterVelkeFi`, kde vyberám všetky hodnoty ktoré sa nachádzajú medzi min a max hodnotou. Ak by som chcel filtrovať podľa hodnoty, musím použiť `FilterMaleFi`, ktorý mi vráti hodnoty ktoré spĺňajú danú požiadavku.

## Úroveň 2 – Zoradenie obcí

### Zoradenie obcí podľa názvu

Ak chcem zoradiť obce podľa názvu, môžem na to použiť vlastnosť charakteru, a teda že viem, či je daný charakter väčší, alebo menší ako ten druhý. Treba však dávať pozor na to, aby bol každý prvok bez špeciálnych znakov, inak porovnávanie nebude fungovať korektne.

Následne môžem využiť triedu `Sort` a vybrať si triediaci algoritmus, v mojom prípade to bude `mergeSort` a môžem pridať boolean, či chcem triediť vzostupne, alebo zostupne.

Čo sa týka filtra a kritéria, môžem postupovať tak ako v predošlej úrovni, teda môžem si vybrať danú územnú jednotku a vybrať kritérium, poprípade si vyberiem filter a použijem dané kritérium.

## Úroveň 3 – Vypísanie informácií o vzdelaní v územných jednotkách a ich zoradenie

Vzhľadom na to, že predošlé úrovne boli implementované na územnú jednotku a nie len na obec, okres alebo kraj, môžem predošlú implementáciu bez problémov použiť aj pre túto úroveň.

Vstupný typ parametra mi prichádza územná jednotka, na základe toho sa môžem rozhodnúť, či pracujem s územnou jednotkou obec, okres alebo kraj. Ak by sa zvolil napríklad Žilinský kraj, ten má pod sebou všetky okresy a okresy majú pod sebou dané obce. Preto by sa spočítali všetky objekty ktoré sú podmnožinou daného objektu.