

**Aplikácia na spracovanie údajov zo Sčítania obyvateľov
domov a bytov z roku 2021**

Semestrálna práca

AUS1

Dominik Bobák

558419

1. Použité údajové štruktúry

1.1 Sorted Sequence Table

Pre uchovanie načítaných územných jednotiek bola zvolená sekvenčná utriedená tabuľka implementovaná pomocou ArrayListu. Keďže sa prvky do tabuľky pridávajú iba na začiatku programu a ďalej sa už tabuľka nemodifikuje, podstatná operácie v tabuľke je nájsť prvok. Vďaka ArrayListu môže ľahko pristupovať k jednotlivým prvkom a keďže sú jej dáta utriedené, môže pomocou bisekcie vyhľadať daný prvok. Zložitosť operácie vyhľadaj je najmenšia spomedzi ostatných tabuliek so zložitosťou $O(\log(N))$.

1.2 Unsorted Sequence Table

Pri filtrovaní územných jednotiek sa dáta, ktoré prešli filtrom, uložia do neutriedenej sekvenčnej tabuľky implementovanej pomocou ArrayListu. Pri filtrovaní nehrá úlohu utriedenie podľa kľúča, takže dáta stačí do tabuľky ukladať postupne. Keďže má táto tabuľka pre vkladanie prvku zložitosť $O(1)$ je najefektívnejšia spomedzi ostatných tabuliek. Taktiež je túto tabuľku možné utriediť podľa zvoleného parametra pokiaľ si užívateľ zvolí triedenie dát. Vďaka ArrayListu je jednoduchý prístup k prvkom a tie sa môžu jednoducho medzi sebou vymieňať.

1.3 Linked List

Zreťazený zoznam bol použitý pri práci s kompozitným filtrom. Keď bude chcieť užívateľ filtrovať, bude mať možnosť vybrať si z viacerých filtrov a tie sa následne zaregistrujú do kompozitného filtra. Keďže sa filtre pridávajú postupne a pri filtrovaní sa prechádza každým jedným filtrom, dôležitá operácia pre zoznam je vložiť prvok. Zreťazený zoznam má oproti ArrayListu výhodu v tom, že nemusí alokovať nové pole keď mu dôjde kapacita. Zložitosť operácie vložiť je $O(1)$.

1.4 Array

Pre uchovanie informácií o počte obyvateľov v danom veku bolo zvolené pole, keďže rozptyl je vždy od 0 po 100 a prístup k prvkom v poli je jednoduchý so zložitosťou $O(1)$.

2. Popis tried a UML diagramy

2.1 Record

Trieda Record slúži na načítanie územných jednotiek a ich uchovanie. V privátnych poliach si drží informácie o všetkých územných jednotkách v utriedenej sekvenčnej tabuľke. Tiež má

k dispozícii jednu územnú jednotku typu štát reprezentujúcu Slovenskú republiku. Ako posledné si drží prázdnu utriedenú sekvenčnú tabuľku pre prípad, keby užívateľ žiada typ tabuľky, ktorá nevyhovuje žiadnym podmienkam. Hlavnou metódou je metóda loadData() s parametrami reprezentujúcimi jednotlivé názvy súborov. Táto metóda následne spolu s privátnymi pomocnými metódami načíta územné jednotky, vzdelanie a vek a správne ich rozdelí. Ďalšie metódy súžia na prístup k tabuľkám a územným jednotkám. getAllTerritorialUnits() vráti tabuľku všetkých územných jednotiek. getTUTable() vráti tabuľku podľa typu UJ a prípadne nadradenej územnej jednotky. getFilteredTable() preberá ako parameter odkaz na neutriedenú tabuľku a filter a následne vyfiltrované dáta uloží do neutriedenej tabuľky. getSlovakia() vráti Slovensko. Metódou clear() sa vymažú všetky načítané územné jednotky.

Record
Class
<div> <div>Fields</div> <div> allTUs_ : Table<wstring, TerritorialUnit*>* emptyTable_ : Table<wstring, TerritorialUnit*>* Slovakia_ : State* </div> </div> <div> <div>Methods</div> <div> ~Record() clear() : void createTU(TERRITORIAL_UNIT_TYPE type, wstring code, wstring officialTitle, wstring mediumTitle, wstring shortTitle, wstring note) : TerritorialUnit* getAllTerritorialUnits() : Table<wstring, TerritorialUnit*> & getFilteredTable(TERRITORIAL_UNIT_TYPE typeTU, Filter<TerritorialUnit*> & filter, Table<wstring, TerritorialUnit*> & table, TerritorialUnit* superiorTU) : Table<wstring, TerritorialUnit*> & getSlovakia() : TerritorialUnit* getTUTable(TERRITORIAL_UNIT_TYPE type, TerritorialUnit* superiorTU) : Table<wstring, TerritorialUnit*> & loadAge(const string& filename) : void loadData(const string& regionsFile, const string& districtsFile, const string& municipalitiesFile, const string& educationFile, const string& ageFile) : void loadEducation(const string& filename) : void loadTUs(TERRITORIAL_UNIT_TYPE type, const string& filename) : void Record() registerTU(TERRITORIAL_UNIT_TYPE type, TerritorialUnit* unit) : void </div> </div>

2.2 App

Trieda App sa stará o komunikáciu s užívateľom. Ako parameter do konštruktora preberie triedu Record s ktorou ďalej pracuje. Pre začiatok komunikácie s užívateľom je nutné zavolať metódu run() ktorá beží, kým sa užívateľ nerozhodne ukončiť aplikáciu. Pri zvolení konkrétneho druhu operácie sa zavolá jedna z metód runSearch() pre bodové vyhľadávanie, runFilters() pre filtrovanie, runSorting() pre triedenie alebo runSelect() pre výber najlepšieho. Trieda si v privátnych poliach uchováva odkazy na všetky kritériá a filtre s ktorými ďalej pracuje. Taktiež má vytvorenú neutriedenú sekvenčnú tabuľku v ktorej si uchováva vyfiltrované územné jednotky zo záznamu. Ďalej má k dispozícii pomocné privátne metódy na načítanie korektných vstupov a výpis na konzolu.

App	Class
Fields	<ul style="list-style-type: none"> cAgeCount_ : CriterionTUAgeCount* cAgeGroupCount_ : CriterionTUAgeGroupCount* cAgeGroupRatio_ : CriterionTUAgeGroupRatio* cAgeRatio_ : CriterionTUAgeRatio* cEduCount_ : CriterionTUEducationCount* cEduRatio_ : CriterionTUEducationRatio* jurisdiction_ : CriterionTUJurisdiction* compositeFilter_ : CompositeFilter<TerritorialUnit*>* cSuperior_ : CriterionTUSuperior* cTitle_ : CriterionTUTitle* cType_ : CriterionTUType* fAgeCount_ : FilterTUAgeCount* fAgeGroupCount_ : FilterTUAgeGroupCount* fAgeGroupRatio_ : FilterTUAgeGroupRatio* fAgeRatio_ : FilterTUAgeRatio* fEducationCount_ : FilterTUEducationCount* fEducationRatio_ : FilterTUEducationRatio* finalTable_ : UnsortedSequenceTable<wstring, TerritorialUnit*>* fJurisdiction_ : FilterTUJurisdiction* fTitle_ : FilterTitle* fType_ : FilterTUType* record_ : Record* sCriterionMaxAgeGroup_ : SelectionCriterionMaxAgeGroup* sCriterionMinAgeGroup_ : SelectionCriterionMinAgeGroup*
Methods	<ul style="list-style-type: none"> ~App() App(Record& record) chooseAge() : int chooseAscending() : bool chooseDoubleMax() : double chooseDoubleMin() : double chooseEconomicAgeGroup() : ECONOMIC_AGE_GROUP chooseEducationType() : EDUCATION_TYPE chooseGender() : GENDER chooseIntMax() : int chooseIntMin() : int chooseTU() : TerritorialUnit* chooseTUType(bool enableALL) : TERRITORIAL_UNIT_TYPE pressEnter() : void printLine(double line) : void printLine(int line) : void printLine(string line) : void printLine(wstring line) : void readDouble() : double readInt() : int readInt(int min, int max) : int readString() : wstring run() : void runFilters() : void runSearch() : void runSelect() : void runSorting() : void

2.3 Vek

Trieda Age si uchováva informácie o počte obyvateľov v danom veku v závislosti od pohlavia.

Age
Class
<div>Fields</div> <div> <div>men_ : Array<int>*</div> <div>women_ : Array<int>*</div> </div>
<div>Methods</div> <div> <div>~Age()</div> <div>addAge(GENDER gender, Array<int>& ages) : void</div> <div>Age()</div> <div>getAgeCount(GENDER gender, int ageCategory) : int</div> <div>getAllAges(GENDER gender) : Array<int>&</div> <div>getEconomicAgeCount(GENDER gender, ECONOMIC_AGE_GROUP ageGroup) : int</div> <div>operator+=(Age& other) : Age&</div> </div>

2.4 Vzdelanie

Trieda Education si uchováva dáta o vzdelaní. V privátnych poliach má uložený počet ľudí bez ukončeného vzdelania, so základným vzdelaním, učňovským, stredným, vyšším vzdelaním, bez vzdelania a počet nezistených.

Education
Class
<div>Fields</div> <div> <div>apprenticeshipEducation_ : int</div> <div>higherEducation_ : int</div> <div>primaryEducation_ : int</div> <div>secondaryEducation_ : int</div> <div>universityEducation_ : int</div> <div>unknown_ : int</div> <div>withoutCompletedEducation_ : int</div> <div>withoutEducation_ : int</div> </div>
<div>Methods</div> <div> <div>~Education()</div> <div>addEducationCount(int count, EDUCATION_TYPE type) : void</div> <div>Education()</div> <div>Education(Education& other)</div> <div>Education(int withoutCompletedEducation, int primaryEducation, int apprenticeshipEducation, int secondaryEducation, int higherEducation, int universityEducation, int withoutEducation, int unknown)</div> <div>getEducationCount(EDUCATION_TYPE type) : int</div> <div>operator+=(Education& other) : Education&</div> </div>

2.5 Územné jednotky

2.5.1 Územná jednotka

Trieda TerritorialUnit predstavuje základnú triedu pre všetky územné jednotky. Uchováva si informácie o kóde, oficiálnemu, strednému a skrátenému názvu, poznámke, type, vzdelaní a veku. Tiež si drží informáciu o nadradenej územnej jednotke.

TerritorialUnit	⌵
Class	
⌵ Fields	
<ul style="list-style-type: none"> age_ : Age* code_ : const wstring education_ : Education* mediumTitle_ : const wstring note_ : const wstring officialTitle_ : const wstring shortTitle_ : const wstring superiorTU_ : TerritorialUnit* typeTU_ : const TERRITORIAL_UNIT_TYPE 	
⌵ Methods	
<ul style="list-style-type: none"> ~TerritorialUnit() addAge(Age& age) : void addEducation(Education& education) : void getAge() : Age& getCode() : wstring getEducation() : Education& getMediumTitle() : wstring getNote() : wstring getNumberOfCitizens() : int getOfficialTitle() : wstring getShortTitle() : wstring getSuperiorTU() : TerritorialUnit& getTypeTU() : TERRITORIAL_UNIT_TYPE setSuperiorTU(TerritorialUnit* superiorTU) : void TerritorialUnit(const wstring code, const wstring officialTitle, const wstring mediumTitle, const wstring shortTitle, const wstring note, const TERRITORIAL_UNIT_TYPE typeTU) 	

2.5.2 Obec

Trieda Municipality je typ územnej jednotky predstavujúci obec.

Municipality	⌵
Class	
→ TerritorialUnit	
⌵ Methods	
<ul style="list-style-type: none"> ~Municipality() Municipality(const wstring code, const wstring officialTitle, const wstring mediumTitle, const wstring shortTitle, const wstring note) setSuperiorTU(TerritorialUnit* superiorTU) : void 	

2.5.3 Okres

Trieda District predstavuje okres a okrem dát z územnej jednotky si uchováva ešte aj tabuľku obcí ktoré patria do okresu.

District	⌵
Class	
→ TerritorialUnit	
⌵ Fields	
<ul style="list-style-type: none"> municipalities_ : Table<wstring, TerritorialUnit*>* 	
⌵ Methods	
<ul style="list-style-type: none"> ~District() addMunicipality(Municipality* municipality) : void District(const wstring code, const wstring officialTitle, const wstring mediumTitle, const wstring shortTitle, const wstring note) getMunicipalitiest() : Table<wstring, TerritorialUnit*>& getSubordinateTUs() : Table<wstring, TerritorialUnit*>& setSuperiorTU(TerritorialUnit* superiorTU) : void 	

2.5.4 Kraj

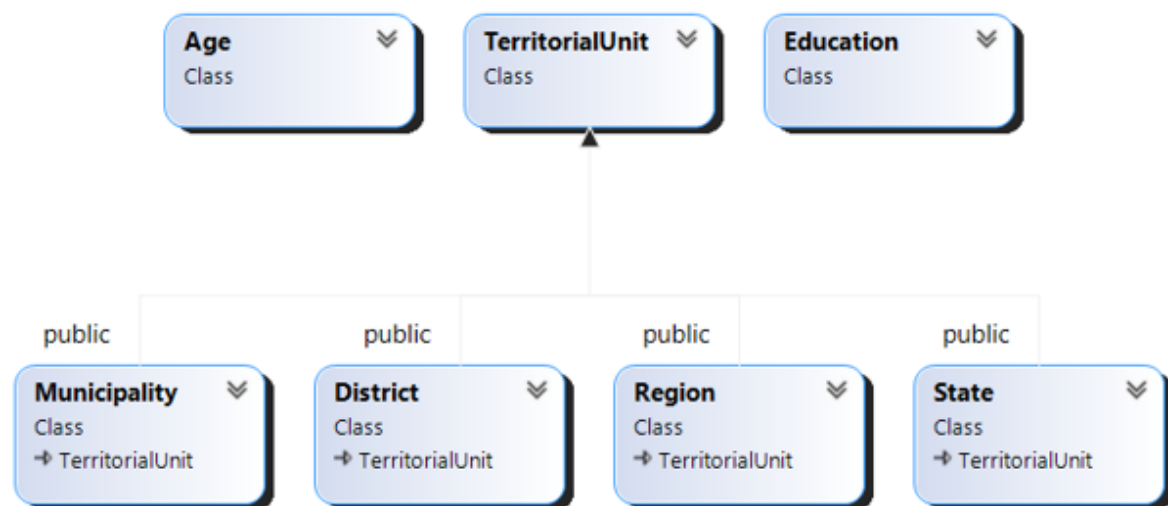
Trieda Region reprezentuje kraj a uchováva si tabuľku obcí, okresov a všetkých podriadených územných jednotiek.

Region	
Class	
→ TerritorialUnit	
Fields	
districts_	Table<wstring, TerritorialUnit*>*
municipalities_	Table<wstring, TerritorialUnit*>*
subordinateTUs_	Table<wstring, TerritorialUnit*>*
Methods	
~Region()	
addDistrict(District* district)	: void
addMunicipality(Municipality* municipality)	: void
getDistricts()	: Table<wstring, TerritorialUnit*>&
getMunicipalities()	: Table<wstring, TerritorialUnit*>&
getSubordinateTUs()	: Table<wstring, TerritorialUnit*>&
Region(const wstring code, const wstring officialTitle, const wstring mediumTitle, const wstring shortTitle, const wstring note)	
setSuperiorTU(TerritorialUnit* superiorTU)	: void

2.5.5 Štát

Trieda State predstavuje štát a uchováva si tabuľku obcí, okresov, krajov a všetkých podriadených územných jednotiek.

State	
Class	
→ TerritorialUnit	
Fields	
districts_	Table<wstring, TerritorialUnit*>*
municipalities_	Table<wstring, TerritorialUnit*>*
regions_	Table<wstring, TerritorialUnit*>*
subordinateTUs_	Table<wstring, TerritorialUnit*>*
Methods	
~State()	
addDistrict(District* district)	: void
addMunicipality(Municipality* municipality)	: void
addRegion(Region* region)	: void
getDistricts()	: Table<wstring, TerritorialUnit*>&
getMunicipalities()	: Table<wstring, TerritorialUnit*>&
getRegions()	: Table<wstring, TerritorialUnit*>&
getSubordinateTUs()	: Table<wstring, TerritorialUnit*>&
setSuperiorTU(TerritorialUnit* superiorTU)	: void
State(const wstring code, const wstring officialTitle, const wstring mediumTitle, const wstring shortTitle, const wstring note)	

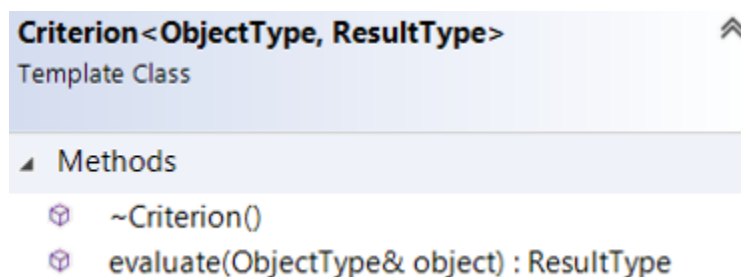


2.6 Kritériá

Kritériá slúžia v aplikácii na to aby zo zvolenej územnej jednotky vrátili jednu vlastnosť. Pokiaľ kritérium k ohodnoteniu vlastnosti potrebuje ďalšie atribúty, tieto atribúty sa nastavujú v konštruktoze ale je ich možné aj zmeniť pomocou verejných metód.

2.6.1 Kritérium

`Criterion<ObjectType, ResultType>` predstavuje základnú šablónovú triedu pre kritériá. K dispozícii má virtuálnu metódu `ohodnot` ktorá vráti požadovanú hodnotu. `ObjectType` je typ spracovávaného objektu a `ResultType` je typ návratovej hodnoty.



2.6.2 Kritérium územná jednotka

`CriterionTU<ResultType>` je základná šablónová trieda pre kritériá pracujúce s územnými jednotkami.

CriterionTU<ResultType>
Template Class
→ Criterion<TerritorialUnit*, ResultType>

2.6.3 Kritérium UJNázov

CriterionTUTitle vráti názov územnej jednotky.

CriterionTUTitle
Class
→ CriterionTU<wstring>

Methods

- ~CriterionTUTitle()
- CriterionTUTitle()
- evaluate(TerritorialUnit*& unit) : wstring

2.6.4 Kritérium UJTyp

CriterionTUType vráti typ územnej jednotky.

CriterionTUType
Class
→ CriterionTU<TERRITORIAL_UNIT_TYPE>

Methods

- ~CriterionTUType()
- CriterionTUType()
- evaluate(TerritorialUnit*& unit) : TERRITORIAL_UNIT_TYPE

2.6.5 Kritérium UJPríslušnosť

CriterionTUTJurisdiction vráti true alebo false podľa toho či územná jednotka patrí zvolenej nadradenej územnej jednotke.

CriterionTUTerritorialUnitJurisdiction	⤴
Class	
→ CriterionTU<bool>	
▾ Fields	
🔗	tu_ : TerritorialUnit*
▾ Methods	
🔗	~CriterionTUTerritorialUnitJurisdiction()
🔗	changeSuperiorTU(TerritorialUnit* newTU) : void
🔗	CriterionTUTerritorialUnitJurisdiction(TerritorialUnit* superiorTU)
🔗	evaluate(TerritorialUnit*& unit) : bool

2.6.6 Kritérium UJVzdeleniePočet

CriterionTUEducationCount vráti počet ľudí s daným vzdelaním.

CriterionTUEducationCount	⤴
Class	
→ CriterionTU<int>	
▾ Fields	
🔗	educationType_ : EDUCATION_TYPE
▾ Methods	
🔗	~CriterionTUEducationCount()
🔗	changeEducationType(EDUCATION_TYPE newEducationType) : void
🔗	CriterionTUEducationCount(EDUCATION_TYPE educationType)
🔗	evaluate(TerritorialUnit*& unit) : int

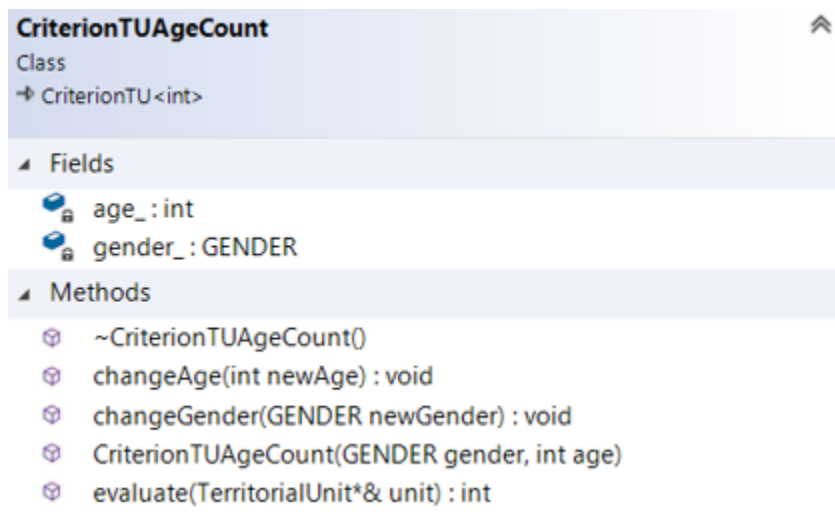
2.6.7 Kritérium UJVzdeleniePodiel

CriterionTUEducationRatio vráti podiel ľudí s daným vzdelaním ku celkovému počtu obyvateľov.

CriterionTUEducationRatio	⤴
Class	
→ CriterionTU<double>	
▾ Fields	
🔗	educationType_ : EDUCATION_TYPE
▾ Methods	
🔗	~CriterionTUEducationRatio()
🔗	changeEducationType(EDUCATION_TYPE newEducationType) : void
🔗	CriterionTUEducationRatio(EDUCATION_TYPE educationType)
🔗	evaluate(TerritorialUnit*& unit) : double

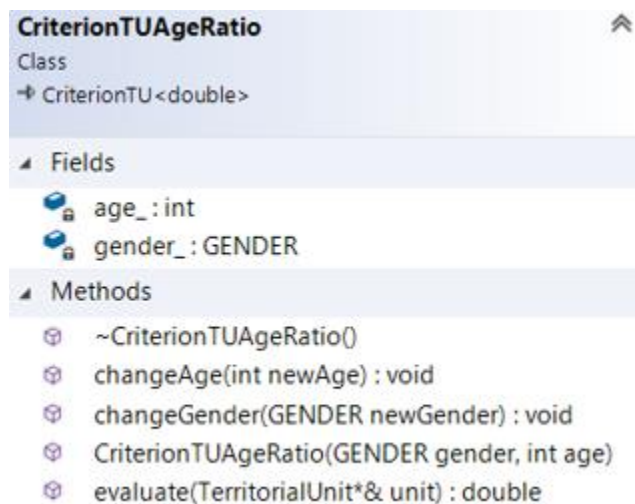
2.6.8 Kritérium UJVekPočet

CriterionTUAgeCount vráti počet ľudí s daným vekom.



2.6.9 Kritérium UJVekPodiel

CriterionTUAgeRatio vráti podiel ľudí s daným vekom ku celkovému počtu obyvateľov.



2.6.10 Kritérium UJVekováSkupinaPočet

CriterionTUAgeGroupCount vráti počet ľudí danej vekovej skupiny.

CriterionTUAgeGroupCount

Class
 ↳ CriterionTU<int>

Fields

ageGroup_ : ECONOMIC_AGE_GROUP

Methods

~CriterionTUAgeGroupCount()
 changeAgeGroup(ECONOMIC_AGE_GROUP newAgeGroup) : void
 CriterionTUAgeGroupCount(ECONOMIC_AGE_GROUP ageGroup)
 evaluate(TerritorialUnit*& unit) : int

2.6.11 Kritérium UJVekováSkupinaPodiel

CriterionTUAgeGroupRatio vráti podiel ľudí danej vekovej skupiny ku celkovému počtu obyvateľov.

CriterionTUAgeGroupRatio

Class
 ↳ CriterionTU<double>

Fields

ageGroup_ : ECONOMIC_AGE_GROUP

Methods

~CriterionTUAgeGroupRatio()
 changeAgeGroup(ECONOMIC_AGE_GROUP newAgeGroup) : void
 CriterionTUAgeGroupRatio(ECONOMIC_AGE_GROUP ageGroup)
 evaluate(TerritorialUnit*& unit) : double

2.6.12 Kritérium UJNadradená

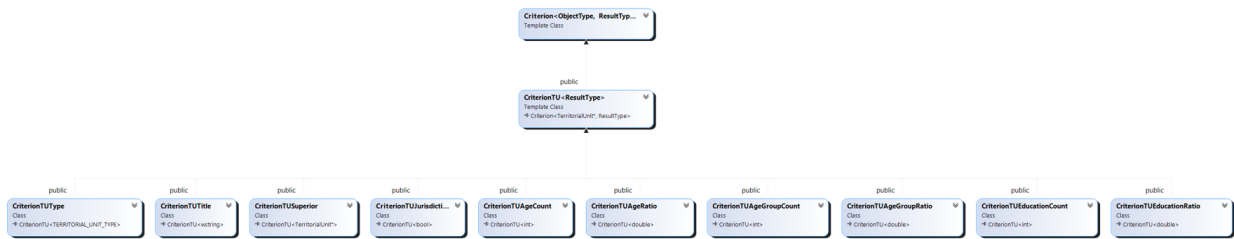
CriterionTUSuperior vráti nadradenú územnú jednotku.

CriterionTUSuperior

Class
 ↳ CriterionTU<TerritorialUnit*>

Methods

~CriterionTUSuperior()
 CriterionTUSuperior()
 evaluate(TerritorialUnit*& unit) : TerritorialUnit*



2.7 Výberové kritériá

Výberové kritériá slúžia k tomu, že z prebratej tabuľky vyberú jeden prvok, ktorého požadovaná vlastnosť je najlepšia z pomedzi ostatných prvkov. Pokiaľ výberové kritérium potrebuje k výberu dodatočné atribúty, tie sa mu nastavujú v konštruktoze alebo je ich možné zmeniť pomocou verejných metód.

2.7.1 Výberové kritérium

`SelectionCriterion<KeyType, ObjectType, ValueType>` je základná trieda pre výberové kritériá. `KeyType` je typ kľúčov v tabuľke, `ObjectType` typ spracovávaných objektov a `ValueType` typ kontrolovanej hodnoty.

```

SelectionCriterion<KeyType, ObjectType, ValueType>
Template Class

Fields
  criterion_ : Criterion<ObjectType, ValueType>*

Methods
  ~SelectionCriterion()
  isTestedValueBetter(ValueType currentValue, ValueType newValue) : bool
  selectBest(Table<KeyType, ObjectType>* table) : ObjectType&
  SelectionCriterion(Criterion<ObjectType, ValueType>* criterion)
  
```

2.7.2 Výberové kritérium zvoľ maximum

`SelectionCriterionMax<KeyType, ObjectType, ValueType>` je základná šablónová trieda pre výberové kritériá, ktoré vyhľadávajú maximálnu hodnotu. `KeyType` je typ kľúčov v tabuľke, `ObjectType` typ spracovávaných objektov a `ValueType` typ kontrolovanej hodnoty.

```

SelectionCriterionMax<KeyType, ObjectType, ValueType>
Template Class
  SelectionCriterion<KeyType, ObjectType, ValueType>

Methods
  isTestedValueBetter(ValueType currentValue, ValueType newValue) : bool
  SelectionCriterionMax(Criterion<ObjectType, ValueType>* criterion)
  
```

2.7.3 Výberové kritérium zvoľ minimum

`SelectionCriterionMin<KeyType, ObjectType, ValueType>` je základná šablónová trieda pre výberové kritériá, ktoré vyhľadávajú minimálnu hodnotu. `KeyType` je typ kľúčov v tabuľke, `ObjectType` typ spracovávaných objektov a `ValueType` typ kontrolovanej hodnoty.

```
SelectionCriterionMin<KeyType, ObjectType, ValueType>
Template Class
↳ SelectionCriterion<KeyType, ObjectType, ValueType>

Methods
  * isTestedValueBetter(ValueType currentValue, ValueType newValue) : bool
  * SelectionCriterionMin(Criterion<ObjectType, ValueType>* criterion)
```

2.7.4 Výberové kritérium najväčšia veková skupina

`SelectionCriterionMaxAgeGroup` vyberie územnú jednotku s najväčším počtom obyvateľov danej vekovej skupiny.

```
SelectionCriterionMaxAgeGroup
Class
↳ SelectionCriterionMax<wstring, TerritorialUnit*, int>

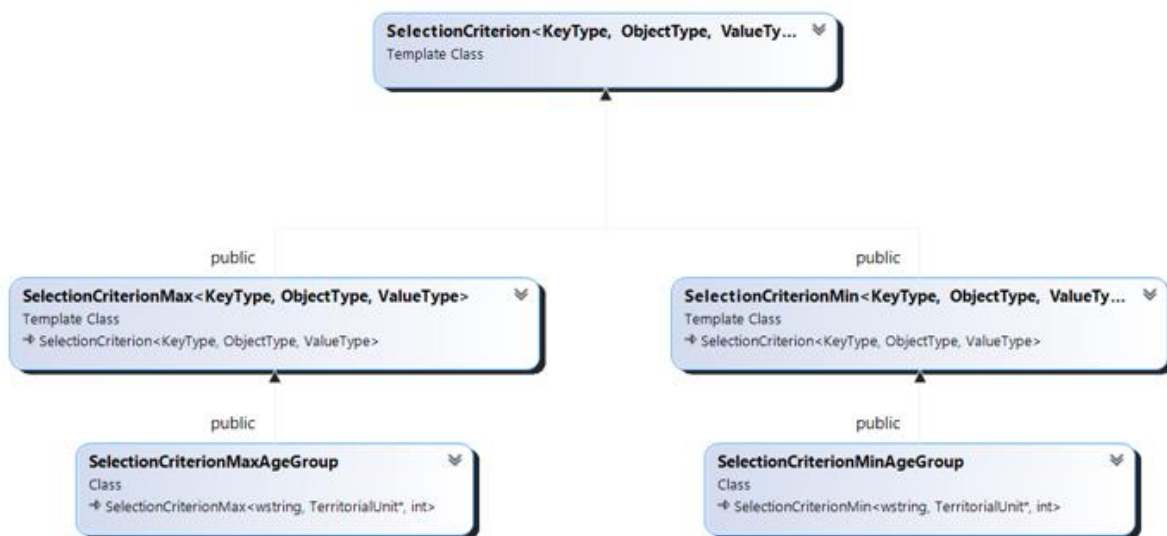
Methods
  * changeAgeGroup(ECONOMIC_AGE_GROUP newAgeGroup) : void
  * SelectionCriterionMaxAgeGroup(ECONOMIC_AGE_GROUP ageGroup)
```

2.7.5 Výberové kritérium najmenšia veková skupina

`SelectionCriterionMinAgeGroup` vyberie územnú jednotku s najmenším počtom obyvateľov danej vekovej skupiny

```
SelectionCriterionMinAgeGroup
Class
↳ SelectionCriterionMin<wstring, TerritorialUnit*, int>

Methods
  * changeAgeGroup(ECONOMIC_AGE_GROUP newAgeGroup) : void
  * SelectionCriterionMinAgeGroup(ECONOMIC_AGE_GROUP ageGroup)
```

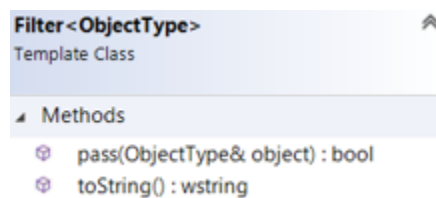


2.8 Filtre

Filtre slúžia k vyfiltrovaníu územných jednotiek na základe danej vlastnosti. Metóda `pass()` vráti `true` alebo `false` na základe toho, či územná jednotka spĺňa danú hodnotu. Každý filter spolupracuje s kritériom z ktorého získava danú vlastnosť územnej jednotky. Hodnoty do filtra a použitého kritéria sa nastavujú v konštruktore ale je ich možné zmeniť pomocou verejných metód. Tiež každý filter má metódu `toString()` ktorá vráti reťazec popisujúci filter a použité kritérium.

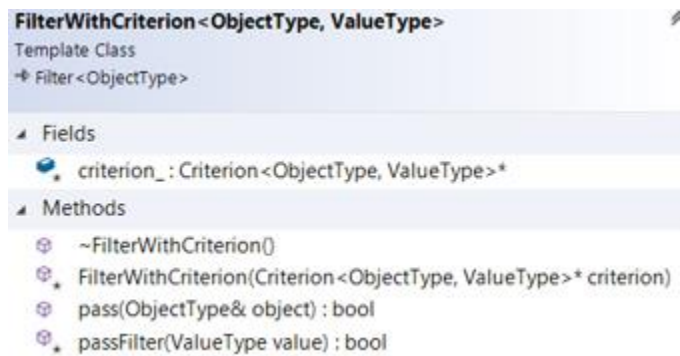
2.8.1 Filter

`Filter<ObjectType>` je základná šablónová trieda pre filtre. `ObjectType` je typ objektu ktorý sa bude filtrovať.



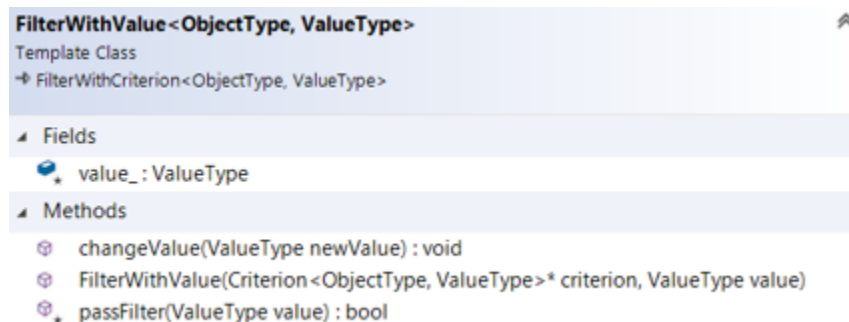
2.8.2 Filter s kritériom

`FilterWithCriterion<ObjectType, ValueType>` je základná šablónová trieda pre filtre hodnotené pomocou kritéria. `ObjectType` je typ filtrovaného objektu a `ValueType` typ kontrolovanej hodnoty.



2.8.3 Filter s hodnotou

`FilterWithValue<ObjectType, ValueType>` je základná šablónová trieda pre filtre s kritériom kde kontrolovaná hodnota musí mať presnú hodnotu. `ObjectType` je typ filtrovaného objektu a `ValueType` typ kontrolovanej hodnoty. Kontrolovaná hodnota sa nastaví v konštruktore ale je možné ju zmeniť pomocou verejnej metódy.



2.8.4 Filter s intervalom

`FilterWithInterval<ObjectType, ValueType>` je základná šablónová trieda pre filtre s kritériom kde kontrolovaná hodnota musí mať byť v určitom intervale. `ObjectType` je typ filtrovaného objektu a `ValueType` typ kontrolovanej hodnoty. Maximum a minimum v intervale sa nastavuje v konštruktore ale je ho možné zmeniť aj verejnými metódami.



2.8.5 Filter názov

FilterTitle je hodnotový filter porovnávajúci názov územnej jednotky. Filter spolupracuje s kritériom Názov.

```
FilterTitle
Class
↳ FilterWithValue<TerritorialUnit*, wstring>

Methods
  FilterTitle(wstring value)
  toString() : wstring
```

2.8.6 Filter UJTyp

FilterUJType je hodnotový filter porovnávajúci typ územnej jednotky. Filter spolupracuje s kritériom UJTyp.

```
FilterTUType
Class
↳ FilterWithValue<TerritorialUnit*, TERRITORIAL_UNIT_TYPE>

Methods
  FilterTUType(TERRITORIAL_UNIT_TYPE value)
  toString() : wstring
```

2.8.7 Filter UJPríslušnosť

FilterTUTJurisdiction je hodnotový filter porovnávajúci či územná jednotka patrí do zvoleného vyššieho územného celku. Filter spolupracuje s kritériom UJPrislusnost.

2.8.8 Filter UJVzdelaniePočet

FilterTUEducationCount je intervalový filter, ktorý kontroluje či počet obyvateľov s daným vzdelaním patrí do zvoleného intervalu. Filter spolupracuje s kritériom UJVzdelaniePodiel.

```
FilterTUEducationCount
Class
↳ FilterWithInterval<TerritorialUnit*, int>

Fields
  educationType_ : EDUCATION_TYPE

Methods
  changeEducationType(EDUCATION_TYPE newEducationType) : void
  FilterTUEducationCount(EDUCATION_TYPE educationType, int min, int max)
  toString() : wstring
```

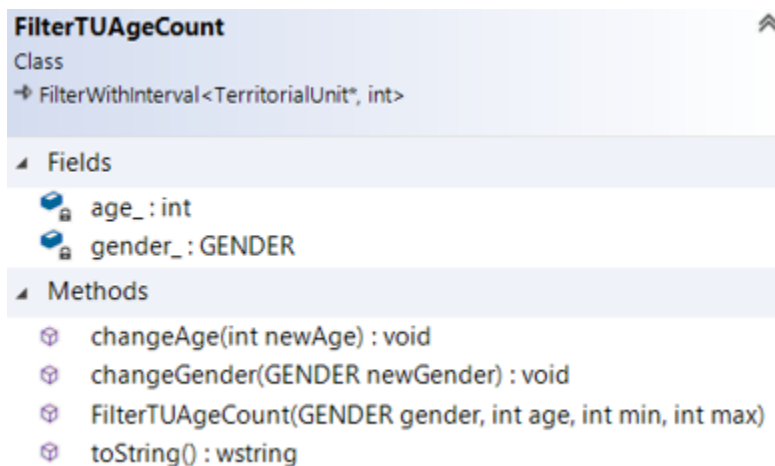
2.8.9 Filter UJVzdelaniePodiel

FilterTUEducationRatio je intervalový filter, ktorý kontroluje či podiel obyvateľov s daným vzdelaním patrí do zvoleného intervalu. Filter spolupracuje s kritériom UJVzdelaniePodiel.



2.8.10 Filter UJVekPočet

FilterTUAgeCount je intervalový filter, ktorý kontroluje či počet obyvateľov s daným vekom patrí do zvoleného intervalu. Filter spolupracuje s kritériom UJVekPocet.



2.8.11 Filter UJVekPodiel

FilterTUAgeRatio je intervalový filter, ktorý kontroluje či podiel obyvateľov s daným vekom patrí do zvoleného intervalu. Filter spolupracuje s kritériom UJVekPodiel.

FilterTUAgeRatio	⤴
Class	
→ FilterWithInterval<TerritorialUnit*, double>	
▴ Fields	
age_ : int	
gender_ : GENDER	
▴ Methods	
changeAge(int newAge) : void	
changeGender(GENDER newGender) : void	
FilterTUAgeRatio(GENDER gender, int age, double min, double max)	
toString() : wstring	

2.8.12 Filter UJVekováSkupinaPočet

FilterTUAgeGroupCount je intervalový filter, ktorý kontroluje či počet obyvateľov v danej vekovej skupine patrí do zvoleného intervalu. Filter spolupracuje s kritériom UJVekováSkupinaPočet.

FilterTUAgeGroupCount	⤴
Class	
→ FilterWithInterval<TerritorialUnit*, int>	
▴ Fields	
ageGroup_ : ECONOMIC_AGE_GROUP	
▴ Methods	
changeEconomicAgeGroup(ECONOMIC_AGE_GROUP newAgeGroup) : void	
FilterTUAgeGroupCount(ECONOMIC_AGE_GROUP ageGroup, int min, int max)	
toString() : wstring	

2.8.13 Filter UJVekováSkupinaPodiel

FilterTUAgeGroupCount je intervalový filter, ktorý kontroluje či podiel obyvateľov v danej vekovej skupine patrí do zvoleného intervalu. Filter spolupracuje s kritériom UJVekováSkupinaPodiel.

FilterTUAgeGroupRatio

Class

→ FilterWithInterval<TerritorialUnit*, double>

Fields

ageGroup_ : ECONOMIC_AGE_GROUP

Methods

- changeEconomicAgeGroup(ECONOMIC_AGE_GROUP newAgeGroup) : void
- FilterTUAgeGroupRatio(ECONOMIC_AGE_GROUP ageGroup, double min, double max)
- toString() : wstring

2.8.14 Kompozitný filter

CompositeFilter<ObjectType> je kompozitný filter do ktorého sa dajú registrovať filtre a pri testovaní prejde každým jedným a zvolený objekt otestuje. Pokiaľ nie je registrovaný žiadny filter tak pri testovaní automaticky vráti hodnotu true. ObjectType je typ filtrovaného objektu.

CompositeFilter<ObjectType>

Template Class

→ Filter<ObjectType>

Fields

filters_ : ArrayList<Filter<ObjectType>*>*

Methods

- ~CompositeFilter()
- clear() : void
- CompositeFilter()
- pass(ObjectType& object) : bool
- registerFilter(Filter<ObjectType>* filter) : void
- toString() : wstring



2.9 Vymenované typy

TERRITORIAL_UNIT_TYPE vymenované typy územných jednotiek.

EDUCATION_TYPE vymenované typy vzdelania.

DATA_TYPE vymenované typy načítavaných dát.

ECONOMIC_AGE_GROUP vymenované typy ekonomických vekových skupín.

GENDER vymenované typy pohlaví.

3. Popis zložitostí

3.1 Načítanie dát

Nové údajové štruktúry sa vkladajú do utriedenej sekvenčnej tabuľky podľa názvu pomocou bisekcie so zložitou $O(\log(N))$. Avšak keďže ich treba pridať do viacerých tabuliek tak celková zložitosť je $O(M \cdot \log(N) + \log(N))$ kde M je počet nadradených územných celkov do ktorých sa územná jednotka musí pridať a posledné $\log(N)$ je z dôvodu pridávania všetkých územných jednotiek do spoločnej tabuľky.

Pre korektné vkladanie územných jednotiek do VUC alebo pre vkladanie vzdelania a veku sa používa kód územnej jednotky. Keďže sú však územné jednotky utriedené podľa názvu je potrebné si všetky prejsť so zložitou $O(N)$

3.2 Bodové vyhľadávanie

Pri bodovom vyhľadávaní sa zvolená územná jednotka vyhľadá podľa názvu. Keďže sú územné jednotky vopred utriedené podľa názvu tak celková zložitosť operácie je $O(\log(N) + k)$ kde M je počet informácií ktoré sa majú vypísať.

3.3 Filtrovanie

Pre zistenie či územné jednotky spĺňajú podmienky filtra je potrebné ich všetky prejsť a otestovať. Celková zložitosť potom vychádza $O(N + k)$ kde k je počet filtrov ktorými má územná jednotka prejsť. Pri použití filtra UJTyp sa vyberie iba tabuľka s daným typom a pri použití filtra UJPríslušnosť sa vyberie daná tabuľka

iba z vybranej územnej jednotky. Týmto sa zredukuje počet filtrovaných jednotiek čo filtrovanie urýchli.

3.4 Triedenie

Počas triedenia si najskôr užívateľ vyberie filtre a na základe nich získa neutriedenú tabuľku vyhovujúcich prvkov. Pri nezvolení žiadneho filtra sa tiež vykoná filtrovanie s tým, že vyhovujúce budú všetky prvky. Zložitosť filtrovania je opísaná v bode 3.4. Následne sa tabuľka triedi algoritmom QuickSort so zložitosťou $O(N \cdot \log(N)) - O(N^2)$.

3.5 Výber najlepších

Pri výbere najlepších si spočiatku užívateľ vyberie filtre ktoré chce aplikovať podobne ako v bodoch 3.3 a 3.4. Následne si zvolí výberové kritérium a to so zložitosťou $O(N)$ prejde celú tabuľku a vráti najlepšiu územnú jednotku.

4. Používateľská príručka

Aplikácia komunikuje s užívateľom pomocou konzoly s jednoduchým výpisom a vstupom. Pri výbere z viacerých možností užívateľa vyzve aby zadal číslo odpovedajúce možnosti. Možnosti výberu sú zvyčajne od 1 po N kde N je počet možností. Pri niektorých výberoch je možné stlačiť 0 čo signalizuje ukončenie výberu. Pre zadávanie parametrov je užívateľ vyzvaný aby zadal korektné hodnoty.

4.1 Hlavné menu

Po spustení aplikácie si užívateľ bude môcť zvoliť z nasledujúcich možností: Bodové vyhľadávanie, Filtrovanie, Triedenie, Výber najlepších alebo Koniec. Možnosť si vyberie stlačením uvedeného čísla. Po vybratí možnosti aplikácie pokračuje k zvolenej úlohe.

```
#####
Spracovanie udajov zo Scitania obyvateľov domov a bytov z roku 2021
#####

Zadajte číslo akcie:
1. Bodové vyhľadavanie
2. Filtrovanie
3. Triedenie
4. Vyber najľahších
0. Koniec
```

4.2 Bodové vyhľadavanie

Pri bodovom vyhľadávaní je užívateľ vyzvaný aby zadal názov územnej jednotky ktorú chce vyhľadať. Následne si zvolí typ vzdelania ktoré chce vypísať a typ ekonomickej vekovej skupiny. Po navolení sa mu vyhľadá územná jednotka a vypíšu sa zvolené informácie. Po stlačení klávesy ENTER sa aplikácia vráti do hlavného menu

```
--- Bodové vyhľadavanie ---
Názov územnej jednotky:
Zilina

Zvolte typ vzdelania:
1. Bez ukončeného vzdelania
2. Základné
3. Učňovské
4. Stredné
5. Vyššie
6. Vysokoskolské
7. Bez vzdelania
8. Neznáme
6

Zvolte ekonomickú vekovú skupinu:
1. Predproduktívni
2. Produktívni
3. Poproduktívni
2

Zvolená územná jednotka:
Názov: Zilina
Typ: Obec
Vzdelanie: Vysokoskolské          Počet: 22861
Ekonomická veková skupina: Produktívni  Počet: 54975
Vyššie územné celky:
Názov: Okres Zilina          Typ: Okres
Názov: Zilinský kraj        Typ: Kraj
Názov: Slovenská republika   Typ: Stat

Pre pokračovanie stlačte klávesu ENTER
```

4.3 Filtrovanie

Po spustení filtrovania bude užívateľ vyzvaný aby si vybral jeden z uvedených filtrov. Následne je potrebné nastaviť parametre k filtrom. Pokiaľ chce užívateľ zmeniť parametre už zvoleného filtra, môže ho označiť znovu a upraviť. Po ukončení výberu sa zobrazia použité filtre a územné jednotky ktoré filterami prešli. Po stlačení klávesy ENTER sa aplikácia vráti do hlavného menu

```
--- Filtrovanie ---  
Vyberte filtre:  
1. UJVzdelaniePocet  
2. UJVzdelaniePodiel  
3. UJTyp  
4. UJPrislusnost  
5. UJVekPocet  
6. UJVekPodiel  
7. UJVekovaSkupinaPocet  
8. UJVekovaSkupinaPodiel  
0. Pokracovat
```

4.4 Triedenie

Na začiatku triedenia si užívateľ navolí kritérium podľa ktorého sa bude triediť. Následne si zvolí spôsob triedenia a ďalej si bude môcť navoliť uvedené filtre v závislosti od použitého kritéria. Následne sa vypíšu utriedené územné jednotky aj s hodnotou kritéria podľa ktorého sa triedili. Po stlačení klávesy ENTER sa aplikácia vráti do hlavného menu.

```
--- Triedenie ---  
Zvolte kritérium pre triedenie:  
1. Nazov  
2. UJVzdelaniePocet  
3. UJVzdelaniePodiel  
4. UJVekPodiel  
5. UJVekovaSkupinaPocet  
1  
  
Zvolte sposob triedenia:  
1. Vzostupne  
2. Zostupne  
1  
  
Vyberte filtre:  
1. UJTyp  
2. UJPrislusnost  
0. Pokracovat  
0
```


4.5 Výber najlepších

Pri výbere najlepších si užívateľ najskôr zvolí výberové kritérium, ekonomickú vekovú skupinu a bude si môcť vybrať z dvoch filtrov. Po navolení sa mu vypíše najlepšia územná jednotka a hodnota výberového kritéria. Po stlačení klávesy ENTER sa aplikácia vráti do hlavného menu.

```
--- Vyber najlepsich ---
Zvolte vyberove kriterium:
1. NajvacsiaVekovaSkupina
2. NajmensiaVekovaSkupina
1

Zvolte ekonomicku vekovu skupinu:
1. Predproduktivni
2. Produktivni
3. Poproduktivni
2

Vyberte filtre:
1. UJTyp
2. UJPrislusnost
0. Pokracovat
0

Najlepsia uzemna jednotka:
Nazov: Slovenska republika
Typ: Stat
Vekova skupina: Produktivni, Pocet: 3652679
```