



ŽILINSKÁ UNIVERZITA V ŽILINE

Fakulta riadenia
a informatiky

Algoritmy a údajové štruktúry 1

SEMESTRÁLNA PRÁCA

Vypracoval: Mário Žilinčík
Akademický rok: 2024/2025
Cvičiaci: Ing. Michal Mrena, PhD.
Cvičenie: Utorok 16:00

Obsah

1.	Návrh Aplikácie	4
1.1.	UzemnaJednotka	4
1.2.	Typ	4
1.3.	Gender	5
1.4.	Loader.....	5
1.5.	Predicates	5
1.6.	Algoritmus.....	6
1.7.	HierarchyNavigator	6
1.8.	TableAnalyzer	6
1.9.	UzemnaJednotkaTable	7
1.10.	SortAlgoritmus	7
1.11.	Main	7
2.	UML Diagram Tried + ÚŠ	8
2.1.	UML Diagram Aktivít Univerzálneho Algoritmu	9
3.	Používateľská príručka	10
3.1.	Úroveň 1	10
3.2.	Úroveň 2.....	11
3.3.	Úroveň 3	15
4.	Úprava vstupných súborov	16
5.	Úroveň 1	17
5.1.	Rozbor vhodnosti údajovej štruktúry std::vector	17
5.2.	Programátorská príručka	17
6.	Úroveň 2	18
6.1.	Načítavanie údajov	18
6.2.	Rozbor vhodnosti údajovej štruktúry	19

7.	Úroveň 3	20
7.1.	Analýza časovej zložitosti – vlož	20
7.2.	Analýza časovej zložitosti – nájdí	20
7.3.	Rozbor vhodnosti údajovej štruktúry	21
8.	Úroveň 4	21
8.1.	Rozbor vhodnosti údajovej štruktúry	21
9.	Záver.....	22
9.1.	Zoznam použitých údajových štruktúr	22

1. Návrh Aplikácie

1.1. UzemnaJednotka

UzemnaJednotka je trieda, ktorá reprezentuje záznam o konkrétnom type územnej jednotky. Obsahuje atribúty name, code a type, ktoré sú spracované pri načítavaní údajov zo vstupných súborov. Okrem základných atribútov trieda obsahuje aj implicitnú sekvenciu typu YearPopulationData. Táto štruktúra v sebe zahŕňa ďalšiu štruktúru PopulationData, ktorá uchováva údaje o počte obyvateľov podľa pohlavia a celkový počet. Týmto prístupom môže jedna územná jednotka obsahovať vlastný zoznam populačných údajov pre viacero rôznych rokov.

Trieda poskytuje metódy na získanie názvu, kódu a typu jednotky, ako aj metódy na získanie počtu obyvateľov pre konkrétny rok alebo pohlavie. Ďalej ponúka možnosť pridávať nové záznamy populácie, upravovať existujúce a vypisovať údaje pre konkrétny rok alebo všetky dostupné roky.

1.2. Typ

Typ je enum trieda, ktorá slúži na uchovávanie informácie o kategórii územnej jednotky. Pomocou tejto hodnoty vieme každú územnú jednotku zaradiť do jednej z nasledujúcich skupín:

- ROOT – Koreň
- GEO – Geografické delenie
- REPUBLIKA – Spolková republika
- REGION – Región
- OBEC – Obec

1.3. Gender

Gender je enum trieda, ktorá slúži na označenie, aké populačné dáta chceme z územnej jednotky získať. Používa sa pri volaní metódy *getPopulation*, ktorá vráti počet obyvateľov pre dané pohlavie (alebo celkový počet) v zadanom roku. Enum obsahuje tri hodnoty:

- MALE – Počet mužov
- FEMALE – Počet žien
- TOTAL – Celkový počet obyvateľov (muži + ženy)

1.4. Loader

Trieda Loader je zodpovedná za načítanie údajov zo vstupných .csv súborov a ich transformáciu na objekty typu *UzemnaJednotka*. Tieto objekty sú následne vkladané do štruktúry typu *std::vector*. Trieda podporuje načítanie jedného alebo viacerých súborov naraz. Každý riadok vstupného súboru predstavuje samostatný záznam, z ktorého sa vytvorí alebo doplní príslušná územná jednotka.

Okrem spracovania údajov pre jednotlivé obce zabezpečuje trieda aj načítanie hierarchie územného členenia. Najprv sa načítajú geografické celky, následne republiky a regióny zo súboru *uzemie.csv*, a potom sa obce zo súboru *obce.csv* priradujú k správnym rodičovským vrcholom.

Súčasťou funkčnosti triedy je aj kumulatívne sčítavanie údajov smerom nahor v hierarchii – od obcí cez regióny až po vyššie úrovne. Pri načítavaní sa zároveň všetky územné jednotky ukladajú aj do tabuľky, ktorá umožňuje ich efektívne vyhľadávanie pomocou názvu, typu a kódu.

1.5. Predicates

Predicates je hlavičkový súbor navrhnutý s dôrazom na univerzálnosť pomocou šablón. Využíva generický typ *T*, čo umožňuje vytvárať predikáty pre ľubovoľný typ objektu. Súbor obsahuje štyri funkcie, ktoré vracajú predikát vytvorený pomocou lambda výrazov. Tieto predikáty následne vyhodnocujú pravdivostnú podmienku a vracajú hodnotu *true* alebo *false* podľa toho, či daný objekt spĺňa požadované kritérium. Tento súbor je následne využívaný v univerzálnom algoritme, kde sa predikáty používajú ako vstupné podmienky pri filtrovaní objektov.

1.6. Algoritmus

Algoritmus je hlavičkový súbor implementovaný pomocou šablón, s cieľom zabezpečiť maximálnu univerzálnosť pri spracovaní rôznych typov údajových štruktúr. Využíva generický typ *T*, vďaka čomu je možné filtrovať a spracovávať ľubovoľné typy objektov.

Trieda pracuje s iterátormi a predikátmi, ktoré jej umožňujú prechádzať zadaný rozsah údajov a vybrať len tie objekty, ktoré spĺňajú danú podmienku. Hlavná metóda *filter* prijíma začiatok a koniec iterátora a ľubovoľný predikát, ktorý vracia hodnotu *true* alebo *false*. Pre jednoduchšie použitie sú definované aj pomocné metódy:

- *filterWithContainsStr*
- *filterWithHasMaxResidents*
- *filterWithHasMinResidents*

Tieto metódy používajú konkrétne predikáty zo súboru *Predicates* a následne volajú univerzálnu metódu *filter*.

Trieda zároveň obsahuje aj metódy na výpis, ktoré vypisujú údaje v závislosti od zadaného roku alebo všetky roky. Metóda *filterSortAndPrint* umožňuje nad vyfiltrovanou množinou vykonať aj triedenie podľa zvoleného kritéria a následný výpis.

1.7. HierarchyNavigator

HierarchyNavigator je pomocná trieda určená na navigáciu v hierarchii typu *MultiWayExplicitHierarchy*, v ktorej sú uložené objekty *UzemnaJednotka*.

Trieda neuchováva hierarchiu samotnú, ale pracuje nad jej existujúcou inštanciou. Uchováva aktuálnu pozíciu v hierarchii prostredníctvom ukazovateľa *currentPosition*. Trieda poskytuje metódy na presun na rodiča, konkrétneho potomka, či výpis priamych potomkov aktuálneho vrcholu.

1.8. TableAnalyzer

Trieda *TableAnalyzer* slúži na testovanie výkonnosti operácií v tabuľkách, ako je vkladanie a vyhľadávanie. Využíva generovanie náhodných hodnôt pre kľúče a dáta, ktoré následne vkladá do tabuľky alebo ich v nej hľadá. Samotná trieda dedí z triedy *ComplexityAnalyzer*.

1.9. UzemnaJednotkaTable

Trieda UzemnaJednotkaTable slúži na správu územných jednotiek rozdelených podľa ich typu. Pre každý typ územnej jednotky uchováva samostatnú utriedenú tabuľku, ktorá umožňuje efektívne vyhľadávanie podľa názvu územnej jednotky.

Pri vkladaní územnej jednotky sa najprv určí jej typ a vyberie sa príslušná tabuľka. Následne sa podľa názvu jednotky (kľúča) buď:

- Vytvorí nový záznam obsahujúci implicitný list, ak sa názov ešte v tabuľke nenachádza
- Územná jednotka sa doplní do existujúceho implicitného listu

Každý kľúč v tabuľke tak odkazuje na implicitný list územných jednotiek s rovnakým názvom. Týmto prístupom môžeme uchovávať aj viaceré jednotky s rovnakým názvom, ktoré sa však líšia kódom.

1.10. SortAlgoritmus

Trieda SortAlgoritmus poskytuje funkcionality na triedenie údajových štruktúr podľa zvoleného komparátora. Obsahuje dve lambda funkcie, ktoré slúžia ako komparátory pre usporiadanie údajov:

- CompareAlphabetical – Porovnáva názvy územných jednotiek podľa nemeckého abecedného poradia, pričom zohľadňuje aj znaky s diakritikou
- ComparePopulation – Porovnáva počet obyvateľov pre zadaný rok a pohlavie

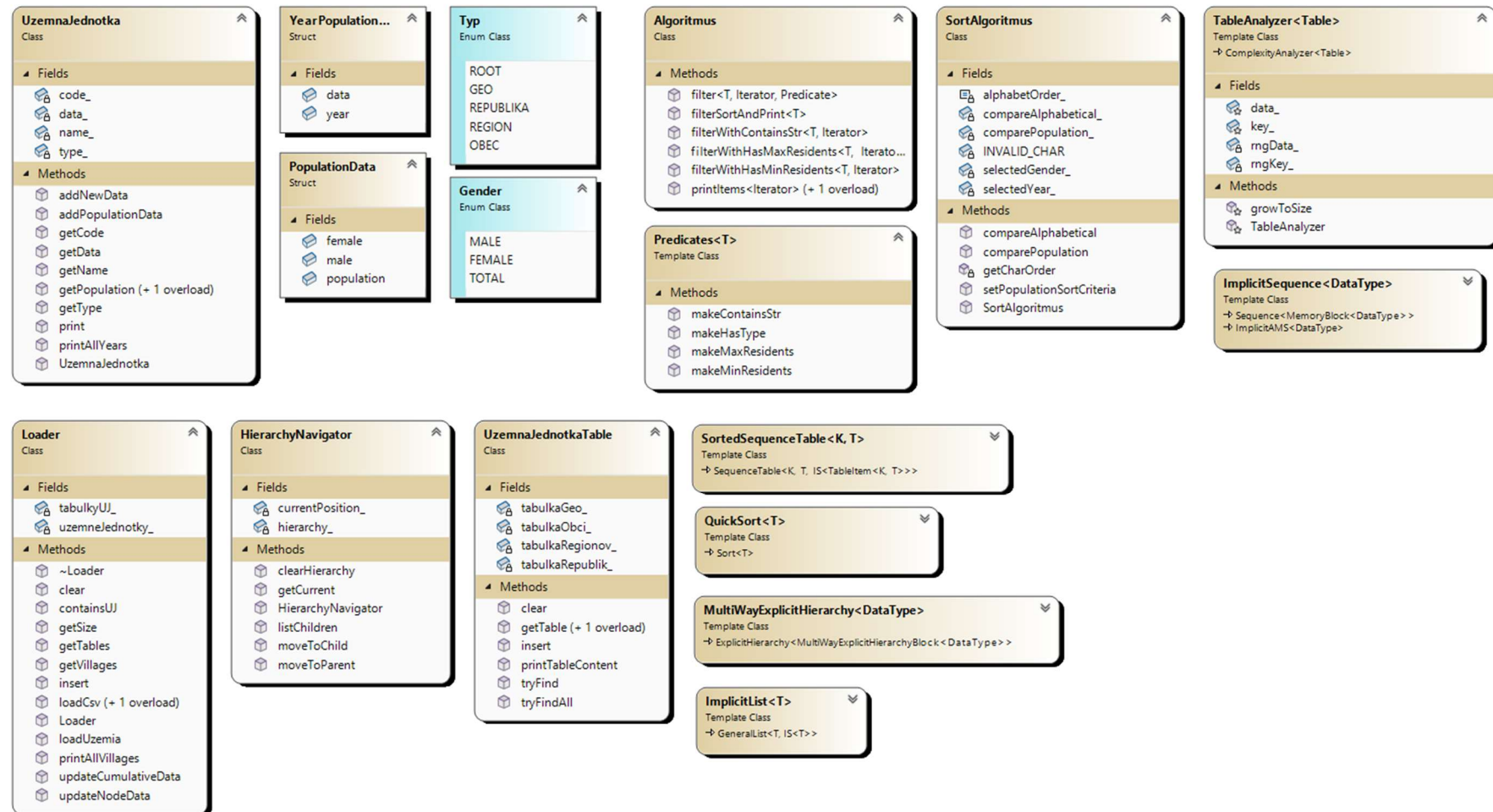
Používateľ má možnosť si dynamicky nastaviť rok a pohlavie, podľa ktorého má komparátor pracovať. Trieda funguje ako obal nad triediacim algoritmom QuickSort, ktorému poskytuje konkrétne komparátory pre triedenie údajov.

1.11. Main

Trieda Main predstavuje vstupný bod celej aplikácie. Zabezpečuje inicializáciu a prepojenie hlavných komponentov, ako sú Loader, Algoritmus, QuickSort, SortAlgoritmus a HierarchyNavigator.

Trieda zároveň slúži na riadenie logiky programu, v rámci ktorého sa volajú funkcie na filtrovanie, triedenie a navigáciu v hierarchii. Trieda taktiež poskytuje jednoduché textové používateľské rozhranie, ktoré umožňuje používateľovi interaktívne zadávať príkazy.

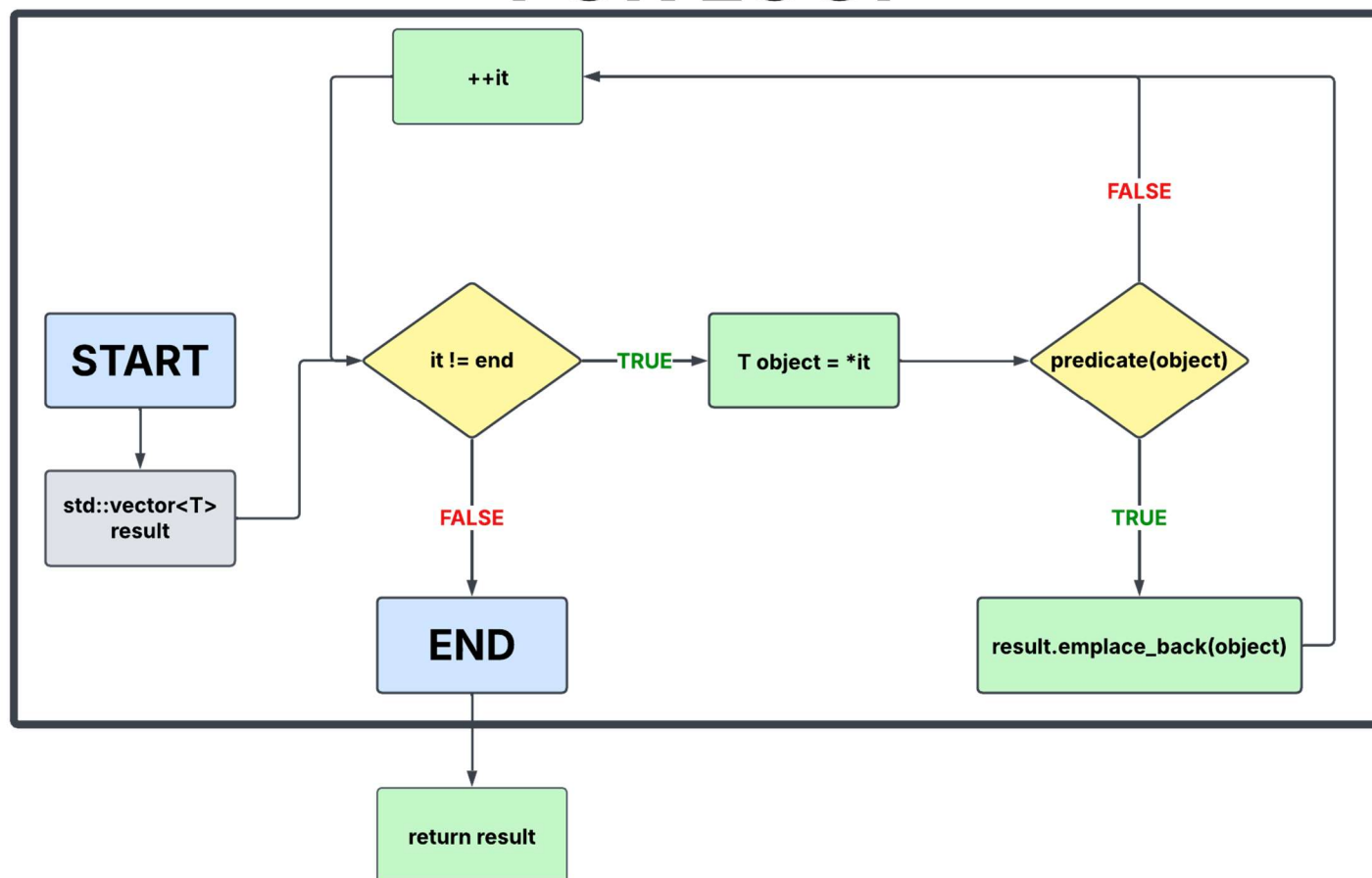
2. UML Diagram Tried + Úš



Obrázok 1- UML Diagram Tried

2.1. UML Diagram Aktivít Univerzálneho Algoritmu

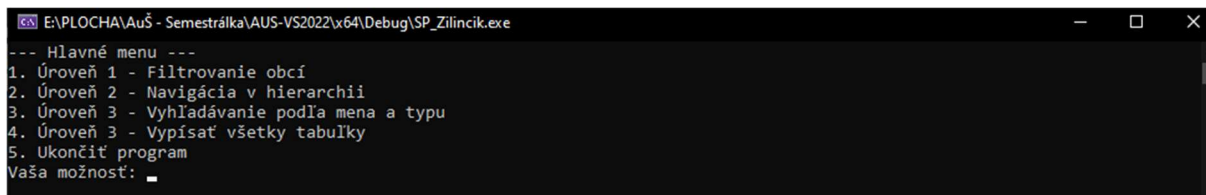
FOR LOOP



Obrázok 2- Diagram Aktivít

3. Používateľská príručka

Ovládanie programu prebieha prostredníctvom textového vstupu a výstupu v konzolovom rozhraní. Po spustení programu je používateľovi zobrazená ponuka možností, pomocou ktorých môže interagovať s aplikáciou. Program pracuje v nekonečnom cykle a zostáva aktívny, kým ho používateľ neukončí explicitne voľbou príkazu s číselným označením „5“, alebo zatvorením konzoly.



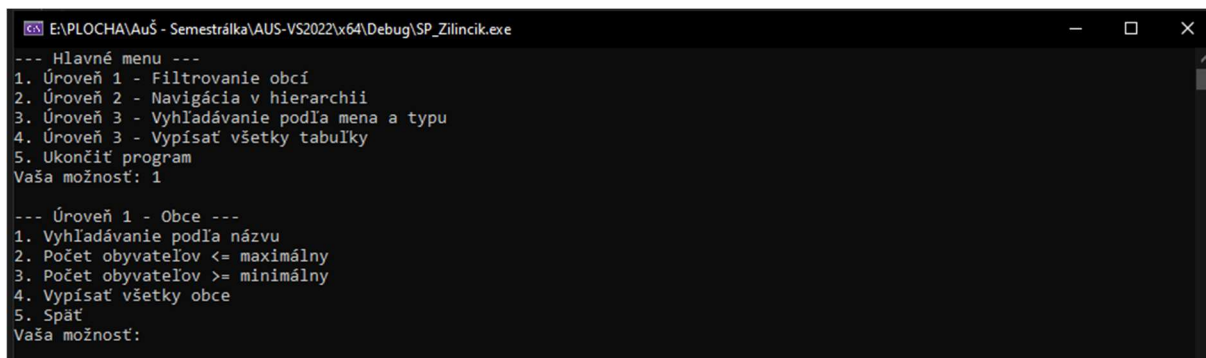
Obrázok 3- Spustenie programu

Po zadaní číselnej hodnoty a teda presunu na danú úroveň program ďalej vypíše používateľovi možnosti z danej úrovne.

3.1. Úroveň 1

V rámci úrovne 1 má používateľ možnosť:

- **Vyhľadať obce podľa názvu**
- **Vyhľadať obce podľa počtu obyvateľov, ktorý je \leq ako zadané číslo**
- **Vyhľadať obce podľa počtu obyvateľov, ktorý je \geq ako zadané číslo**
- **Vypísať všetky obce**
- **Krok späť**



Obrázok 4- Možnosti úrovne 1

Po zadaní parametrov pre vyhľadanie sa následne používateľovi vypíšu vyhľadané prvky.

```

--- Úroveň 1 - Obce ---
1. Vyhľadávanie podľa názvu
2. Počet obyvateľov <= maximálny
3. Počet obyvateľov >= minimálny
4. Vypísať všetky obce
5. Späť
Vaša možnosť: 1
Zadajte reťazec:
Krumbach
Zadajte rok:
2020
Chcete použiť SORT?[A/N]
N
Krumbach | R: 2020 | C: <32315> | M: 1153 | Ž: 1119 | P: 2272 | T: OBEC
Krumbach | R: 2020 | C: <80221> | M: 543 | Ž: 493 | P: 1036 | T: OBEC
--- Úroveň 1 - Obce ---
1. Vyhľadávanie podľa názvu
2. Počet obyvateľov <= maximálny
3. Počet obyvateľov >= minimálny
4. Vypísať všetky obce
5. Späť
Vaša možnosť: _

```

Obrázok 5 - Output úrovne 1

3.2. Úroveň 2

V rámci úrovne 2 má používateľ možnosť:

- Presunúť sa na rodiča
- Presunúť sa na potomka
- Zobrazit' všetkých potomkov
- Použiť predikát
- Vypísať populačné údaje aktuálneho vrcholu
- Krok späť

```

E:\PLOCHA\Auš - Semestrálna\AUS-VS2022\64\Debug\SP_Zilincik.exe
--- Hlavné menu ---
1. Úroveň 1 - Filtrovanie obcí
2. Úroveň 2 - Navigácia v hierarchii
3. Úroveň 3 - Vyhľadávanie podľa mena a typu
4. Úroveň 3 - Vypísať všetky tabuľky
5. Ukončiť program
Vaša možnosť: 2


Aktuálna pozícia: RAKÚSKO

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobrazit' všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť
Vaša možnosť: _

```

Obrázok 6 - Možnosti úrovne 2

Voľbou „1“ sa používateľ pokúsi posunúť v hierarchii na rodiča aktuálneho vrcholu. Pokiaľ sa používateľ momentálne nachádza v koreni a teda nie je možnosť sa posunúť vyššie, program ho o tom informuje a používateľ nie je nikde posunutý.



```
E:\PLOCHA\AuŠ - Semestrálka\AUS-VS2022\x64\Debug\SP_Zilincik.exe

Aktuálna pozícia: RAKÚSKO

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobrazíť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť
Vaša možnosť: 1

[INFO] You're already at the root.

Aktuálna pozícia: RAKÚSKO

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobrazíť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť
Vaša možnosť: _
```

Obrázok 7 - Posun na rodiča

Voľbou „2“ sa používateľ pokúsi posunúť v hierarchii na jedného z viacerých potomkov aktuálneho vrcholu. Program používateľovi vypíše všetkých aktuálnych synov daného vrcholu a očakáva vstup od používateľa. Pokiaľ sa používateľ úspešne posunie na jedného zo synov, stáva sa z neho aktuálny vrchol. V prípade, že vrchol už nemá synov, používateľ je o tom informovaný a už sa nemôže posunúť nižšie.



```
E:\PLOCHA\AuŠ - Semestrálka\AUS-VS2022\x64\Debug\SP_Zilincik.exe

Aktuálna pozícia: RAKÚSKO

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobrazíť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť
Vaša možnosť: 2

[INFO] Potomkovia aktuálneho vrcholu:
[0] Eastern Austria
[1] Southern Austria
[2] Western Austria
Vadajte index syna:
0

[INFO] You moved to: Eastern Austria

Aktuálna pozícia: Eastern Austria

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobrazíť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť
Vaša možnosť:
```

Obrázok 8 - Posun na syna

Voľbou „3“ program používateľovi vypíše všetkých aktuálnych potomkov aktuálneho vrcholu. Vo výpise sa nachádzajú všetky populačné dáta pre každý rok, spoločne aj s kódom a typom územnej jednotky.

```

E:\PLOCHA\AuŠ - Semestrálka\AUS-VS2022\x64\Debug\SP_Zilincik.exe
Aktuálna pozícia: Mittelburgenland

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobraziť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť
Vaša možnosť: 3
Mittelburgenland C: 111
Rok: 2020 | C: 111 | Ž: 18996 | M: 18388 | P: 37384 | T: REGION
Rok: 2021 | C: 111 | Ž: 19047 | M: 18406 | P: 37453 | T: REGION
Rok: 2022 | C: 111 | Ž: 19074 | M: 18450 | P: 37524 | T: REGION
Rok: 2023 | C: 111 | Ž: 19182 | M: 18561 | P: 37743 | T: REGION
Rok: 2024 | C: 111 | Ž: 19226 | M: 18581 | P: 37807 | T: REGION
Deutschkreutz C: <10801>
Rok: 2020 | C: <10801> | Ž: 1587 | M: 1482 | P: 3069 | T: OBEC
Rok: 2021 | C: <10801> | Ž: 1619 | M: 1495 | P: 3114 | T: OBEC
Rok: 2022 | C: <10801> | Ž: 1636 | M: 1524 | P: 3160 | T: OBEC
Rok: 2023 | C: <10801> | Ž: 1650 | M: 1530 | P: 3180 | T: OBEC
Rok: 2024 | C: <10801> | Ž: 1639 | M: 1527 | P: 3166 | T: OBEC

```

Obrázok 9 - Výpis všetkých potomkov vrcholu

Voľbou „4“ môže používateľ filtrovať potomkov aktuálneho vrcholu pomocou predikátov, podobne ako v úrovni 1. Navyše je dostupná možnosť overiť typ územnej jednotky, pričom program informuje, či typ aktuálneho vrcholu zodpovedá zadanej hodnote.

```

E:\PLOCHA\AuŠ - Semestrálka\AUS-VS2022\x64\Debug\SP_Zilincik.exe
--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobraziť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť
Vaša možnosť: 4
--- Predikáty ---
1. Názov obsahuje reťazec
2. Počet obyvateľov <= maximálny
3. Počet obyvateľov >= minimálny
4. Typ územnej jednotky
Vaša možnosť: 4
Zadajte typ (0-ROOT, 1-GEO, 2-REPUBLIKA, 3-REGION, 4-OBEC):
0
[INFO] Typ sa nezhoduje.

```

Obrázok 10 - Predikát typu územnej jednotky

Voľbou „5“ program používateľovi vypíše populačné údaje aktuálneho vrcholu pre každý rok.

```

E:\PLOCHA\AuŠ - Semestrálka\AUS-VS2022\x64\Debug\SP_Zilincik.exe

[INFO] You're already at the root.

Aktuálna pozícia: RAKÚSKO

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobrazíť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť

Vaša možnosť: 5
RAKÚSKO C: 0
Rok: 2020 | C: 0 | Ž: 4522292 | M: 4378772 | P: 8901064 | T: ROOT
Rok: 2021 | C: 0 | Ž: 4535257 | M: 4396484 | P: 8931741 | T: ROOT
Rok: 2022 | C: 0 | Ž: 4553444 | M: 4425485 | P: 8978929 | T: ROOT
Rok: 2023 | C: 0 | Ž: 4619957 | M: 4484815 | P: 9104772 | T: ROOT
Rok: 2024 | C: 0 | Ž: 4643918 | M: 4514832 | P: 9158750 | T: ROOT

Aktuálna pozícia: RAKÚSKO

```

Obrázok 11 - Vypísanie údajov aktuálneho vrcholu

V rámci 1. a 2. úrovne je taktiež možné po použití predikátu výsledky pred výpisom usporiadať. Pred usporiadaním program používateľa vyzve, či chce použiť sort. Odpoveďou „A“ následne program používateľa vyzve aby zadal akým spôsobom chce prvky usporiadať. Voľbou „A“ sa prvky usporiadajú abecedne. Voľbami „M“ – na základe populácie mužov, voľbami „Ž“ – na základe populácie žien, „P“ – na základe celkovej populácie.

```

E:\PLOCHA\AuŠ - Semestrálka\AUS-VS2022\x64\Debug\SP_Zilincik.exe

--- Predikáty ---
1. Názov obsahuje reťazec
2. Počet obyvateľov <= maximálny
3. Počet obyvateľov >= minimálny
4. Typ územnej jednotky
Vaša možnosť: 1
Zadajte reťazec:
Krum
Zadajte rok:
2020
Chcete použiť SORT?[A/N]
A
SORT ABECEDY [A] / SORT POPULÁCIE [M/Ž/P]?
A
Krumau am Kamp | R: 2020 | C: <31321> | M: 375 | Ž: 375 | P: 750 | T: OBEC
Krumbach | R: 2020 | C: <80221> | M: 543 | Ž: 493 | P: 1036 | T: OBEC
Krumbach | R: 2020 | C: <32315> | M: 1153 | Ž: 1119 | P: 2272 | T: OBEC
Krummußbaum | R: 2020 | C: <31517> | M: 761 | Ž: 773 | P: 1534 | T: OBEC
Krumppendorf am Wörthersee | R: 2020 | C: <20415> | M: 1643 | Ž: 1810 | P: 3453 | T: OBEC

Aktuálna pozícia: RAKÚSKO

--- Úroveň 2 - Hierarchia ---
1. Ísť na rodiča
2. Ísť na potomka
3. Zobrazíť všetkých potomkov
4. Použiť predikát
5. Údaje o populácii aktuálneho vrcholu
6. Späť

Vaša možnosť:

```

Obrázok 12 - Využitie QuickSortu

3.3. Úroveň 3

V rámci úrovne 3 má používateľ možnosť vyhľadať územné jednotky zadaním názvu a typu hľadanej jednotky. Po zadaní vstupných parametrov program používateľovi vypíše výsledok.

```
E:\PLOCHA\AuŠ - Semestrálka\AUS-VS2022\x64\Debug\SP_Zilincik.exe
Vaša možnosť: 3

--- Úroveň 3 - Vyhľadavanie v tabuľke ---
1. Hľadať podľa mena a typu
2. Späť
Vaša možnosť: 1
Zadajte názov jednotky:
Krumbach
Zadajte typ (0-ROOT, 1-GEO, 2-REPUBLIKA, 3-REGION, 4-OBEC):
4
[INFO] Nájdenných 2 jednotiek s názvom 'Krumbach':
-----
Krumbach C: <32315>
Rok: 2020 | C: <32315> | Ž: 1119 | M: 1153 | P: 2272 | T: OBEC
Rok: 2021 | C: <32315> | Ž: 1137 | M: 1174 | P: 2311 | T: OBEC
Rok: 2022 | C: <32315> | Ž: 1158 | M: 1175 | P: 2333 | T: OBEC
Rok: 2023 | C: <32315> | Ž: 1197 | M: 1213 | P: 2410 | T: OBEC
Rok: 2024 | C: <32315> | Ž: 1181 | M: 1209 | P: 2390 | T: OBEC
-----
Krumbach C: <80221>
Rok: 2020 | C: <80221> | Ž: 493 | M: 543 | P: 1036 | T: OBEC
Rok: 2021 | C: <80221> | Ž: 513 | M: 553 | P: 1066 | T: OBEC
Rok: 2022 | C: <80221> | Ž: 520 | M: 563 | P: 1083 | T: OBEC
Rok: 2023 | C: <80221> | Ž: 528 | M: 566 | P: 1094 | T: OBEC
Rok: 2024 | C: <80221> | Ž: 543 | M: 577 | P: 1120 | T: OBEC

--- Úroveň 3 - Vyhľadavanie v tabuľke ---
1. Hľadať podľa mena a typu
2. Späť
Vaša možnosť:
```

Obrázok 13 - Vyhľadavanie v tabuľke

4. Úprava vstupných súborov

Všetky pôvodné vstupné súbory boli kódované v UTF-8, čo spôsobovalo problémy pri zobrazovaní niektorých špeciálnych znakov, ako je napríklad „ß“. Z tohto dôvodu bolo kódovanie všetkých súborov konvertované na Windows-1250 pomocou nástroja Notepad++, čím sa zabezpečila správna interpretácia diakritiky a špeciálnych znakov.

Súbory 2020.csv, 2021.csv, 2022.csv, 2023.csv a 2024.csv obsahovali viaceré nadbytočné stĺpce a textové hodnoty, ktoré boli odstránené. Rok reprezentujúci údaje bol presunutý do prvého riadku súboru, čo umožnilo jednoduchšie spracovanie pri načítavaní. Hlavičky stĺpcov boli upravené tak, aby zodpovedali načítavaným hodnotám, a všetky údaje boli usporiadané do jednotného formátu.

Zo súboru „obce.csv“ bol odstránený posledný riadok, ktorý neobsahoval žiadne relevantné údaje.

Súbor „uzemie.csv“ nebol nijak upravovaný – jeho štruktúra bola plne vyhovujúca požiadavkám načítavania.

5. Úroveň 1

5.1. Rozbor vhodnosti údajovej štruktúry `std::vector`

V prvej úrovni semestrálnej práce bola zvolená údajová štruktúra `std::vector`, ktorá predstavuje dynamické pole s možnosťou automatického rozširovania kapacity pri vkladaní nových prvkov. Jej výhodou je, že prvky sú uložené v súvislom pamäťovom priestore, čo umožňuje veľmi rýchly a efektívny prístup k údajom pomocou indexovania – v čase $O(1)$.

Vektor bol zároveň vhodnou voľbou aj z praktického hľadiska: ponúka jednoduché použitie, podporuje iteráciu pomocou štandardných iterátorov, a zároveň poskytuje priame metódy na vkladanie, výber a manipuláciu s prvkami, čo je ideálne pre spracovanie a filtrovanie načítaných údajov.

5.2. Programátorská príručka

Návrh a implementácia univerzálneho algoritmu na filtrovanie údajov boli realizované s dôrazom na jednoduchosť a opätovnú použiteľnosť. Hlavná metóda *filter()* pracuje so vstupnými iterátormi reprezentujúcimi začiatok a koniec ľubovoľnej sekvenčnej údajovej štruktúry. Využitím šablón umožňuje algoritmus automatické odvodenie typu objektov, ktoré sú spracovávané.

Počas behu algoritmus iteruje cez všetky prvky v zadanom rozsahu a na každý z nich aplikuje zadaný predikát – t.j. logickú podmienku, ktorá určuje, či má byť prvok zahrnutý do výsledku. Preddefinované predikáty sú implementované pomocou lambda funkcií v súbore `Predicates.h`.

Pre zjednodušenie použitia boli vytvorené pomocné metódy, ktoré interne využívajú hlavnú funkciu *filter()* a vkladajú do nej konkrétny predikát. Používateľ tak pri volaní zadáva iba požadovaný parameter filtrovania a rozsah pomocou iterátorov.

V prípade rozšírenia programu o ďalší predikát je odporúčané vytvoriť pomocou lambda funkcií novú funkciu v súbore `Predicates.h` a následne vytvoriť pomocnú funkciu v súbore `Algoritmus.h`. Návratová hodnota tejto funkcie bude univerzálny algoritmus ktorý ako parameter získa novo vytvorenú lambda funkciu ktorá prijíma určité parametre v závislosti od predikátu.

6. Úroveň 2

6.1. Načítavanie údajov

Načítavanie údajov prebieha v triede Loader. Proces začína čítaním súboru „uzemie.csv“, ktorý obsahuje geografické členenie Rakúska – geografické delenie, spolkové republiky a regióny. Každá územná jednotka obsahovala svoj špecifický kód, ktorý slúži ako identifikátor aj index v hierarchii.

Príklad členenia:

- Eastern Austria - <AT1> - Prvý syn koreňa
- Burgenland - <AT11> - Prvý syn rodiča (Eastern Austria)
- Mittelburgenland - <AT111> - Prvý syn Burgenlandu
- Nordburgenland - <AT112> - Druhý syn Burgenlandu

Pri spracovaní riadku sa získa názov a kód územnej jednotky. Kód je potrebné orezať, aby sa odstránili nežiadúce znaky a získala sa čistá číselná forma. Na základe dĺžky kódu sa identifikuje úroveň hierarchie (**1 – GEO, 2 – REPUBLIKA, 3 – REGION**) a podľa toho sa územná jednotka vloží na správne miesto v strome.

Pri vkladaní sa používa metóda *emplaceSon()*, ktorá vloží danú územnú jednotku na pozíciu podľa vypočítaného indexu (odčítaním 1 kvôli nulovému indexovaniu). V závislosti od dĺžky kódu sa jednotka zaradí na príslušnú úroveň hierarchie:

- Dĺžka 1 – jednotka predstavuje geografické delenie a vkladá sa priamo pod koreň.
- Dĺžka 2 – jednotka predstavuje spolkovú republiku a vkladá sa ako potomok geografického delenia.
- Dĺžka 3 – jednotka predstavuje región a vkladá sa ako potomok konkrétnej spolkovej republiky.

Po spracovaní vyšších úrovní sa spracováva súbor „obce.csv“, ktorý obsahuje záznamy o jednotlivých obciach. Princíp zostáva rovnaký – najprv sa kód obce oreže, aby bolo možné identifikovať jej umiestnenie v hierarchii. Následne sa naviguje do konkrétnej vetvy stromu (**GEO -> REPUBLIKA -> REGION**).

Zložitosť načítavania je v tomto prípade $O(n)$, keďže spracovávame každý riadok len jeden krát.

6.2. Rozbor vhodnosti údajovej štruktúry

V tejto úrovni programu bola použitá explicitná viaccestná hierarchia (MultiWayExplicitHierarchy), ktorá umožňuje reprezentovať strom, kde každý vrchol môže mať ľubovoľný počet synov. Táto štruktúra je ideálna pre modelovanie územných jednotiek, pretože počet potomkov jednotlivých vrcholov nie je dopredu známy.

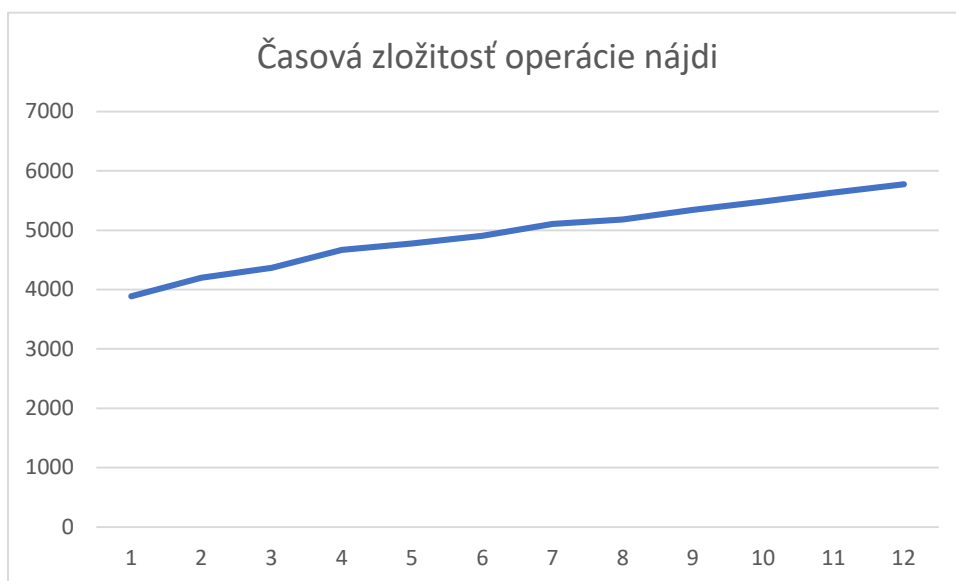
7. Úroveň 3

Pre implementovanú tabuľku som využil metódy na dynamické zväčšovanie jej veľkosti a generovanie náhodných hodnôt pre kľúče a dáta, čím som následne vykonal analýzu časového správania jednotlivých operácií.

7.1. Analýza časovej zložitosti – vlož

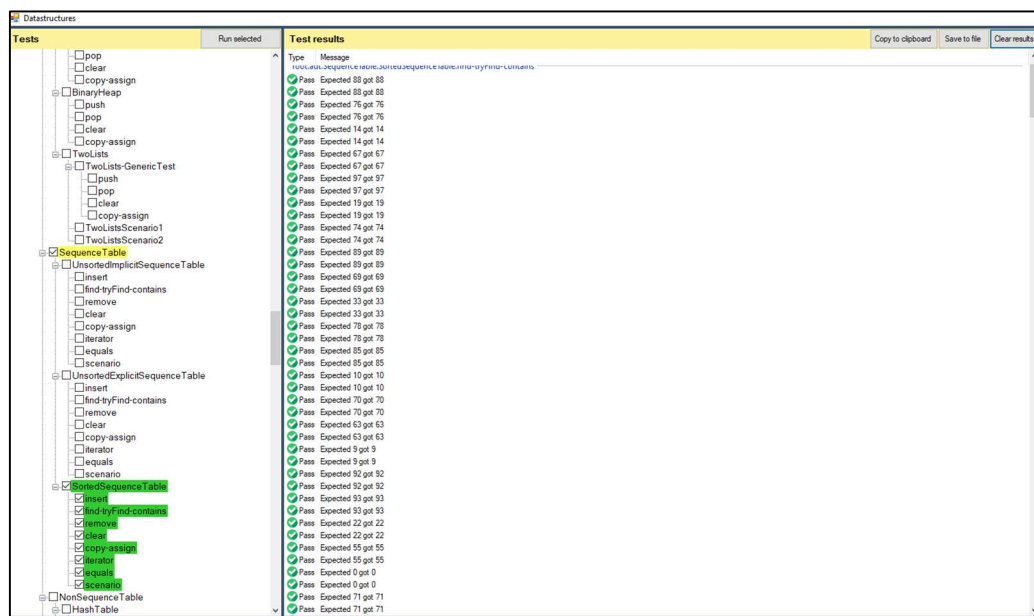


7.2. Analýza časovej zložitosti – nájdí



7.3. Rozbor vhodnosti údajovej štruktúry

V tejto úrovni som sa rozhodol použiť utriedenú sekvenčnú tabuľku (SortedSequenceTable). Táto tabuľa je vysoko efektívna pri vkladaní a prístupe ku prvkom. Ako kľúč tabuľky bol vybraný názov územnej jednotky, vďaka čomu som mal možnosť prvky v tabuľke vyhľadávať na základe názvov. Ako dáta bol použitý implicitný list (ImplicitList). Táto štruktúra bola vhodná pre ukladanie viacerých jednotiek s rovnakým názvom, pričom táto štruktúra taktiež umožňuje efektívne vkladanie prvkov na koniec listu a následný efektívny prístup k nim.



Obrázok 14 - Test SortedSequenceTable

8. Úroveň 4

8.1. Rozbor vhodnosti údajovej štruktúry

V poslednej úrovni som pre triediaci algoritmus zvolil QuickSort, ktorý sa zdal ako vhodná voľba na základe efektívnej výpočtovej zložitosti ktorá sa pohybuje od hodnôt $O(N \cdot \log_2(N))$ po $O(N^2)$.

Pre triedenie sme vytvorili pomocnú triedu SortAlgoritmus, v ktorej sú implementované lambda funkcie, ktoré QuickSort použije ako komparátory pre triedenie určenej štruktúry.

Trieda obsahuje dva komparátory:

- CompareAlphabetical – Porovnávanie názvu územných jednotiek
- ComparePopulation – Porovnávanie určenej populácie územných jednotiek

9. Záver

9.1. Zoznam použitých údajových štruktúr

V rámci práce boli použité nasledovné štruktúry, ktoré boli implementované zo školských materiálov z knižnice DataStructures:

- Implicitný List (ImplicitList)
- Implicitná Sekvencia (ImplicitSequence)
- Utriedená Sekvenčná Tabuľka (SortedSequenceTable)
- Rýchle Triedenie (QuickSort)
- Viaccestná Explicitná Hierarchia (MultiWayExplicitHierarchy)