

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

Algoritmy a údajové štruktúry
Semestrálna práca

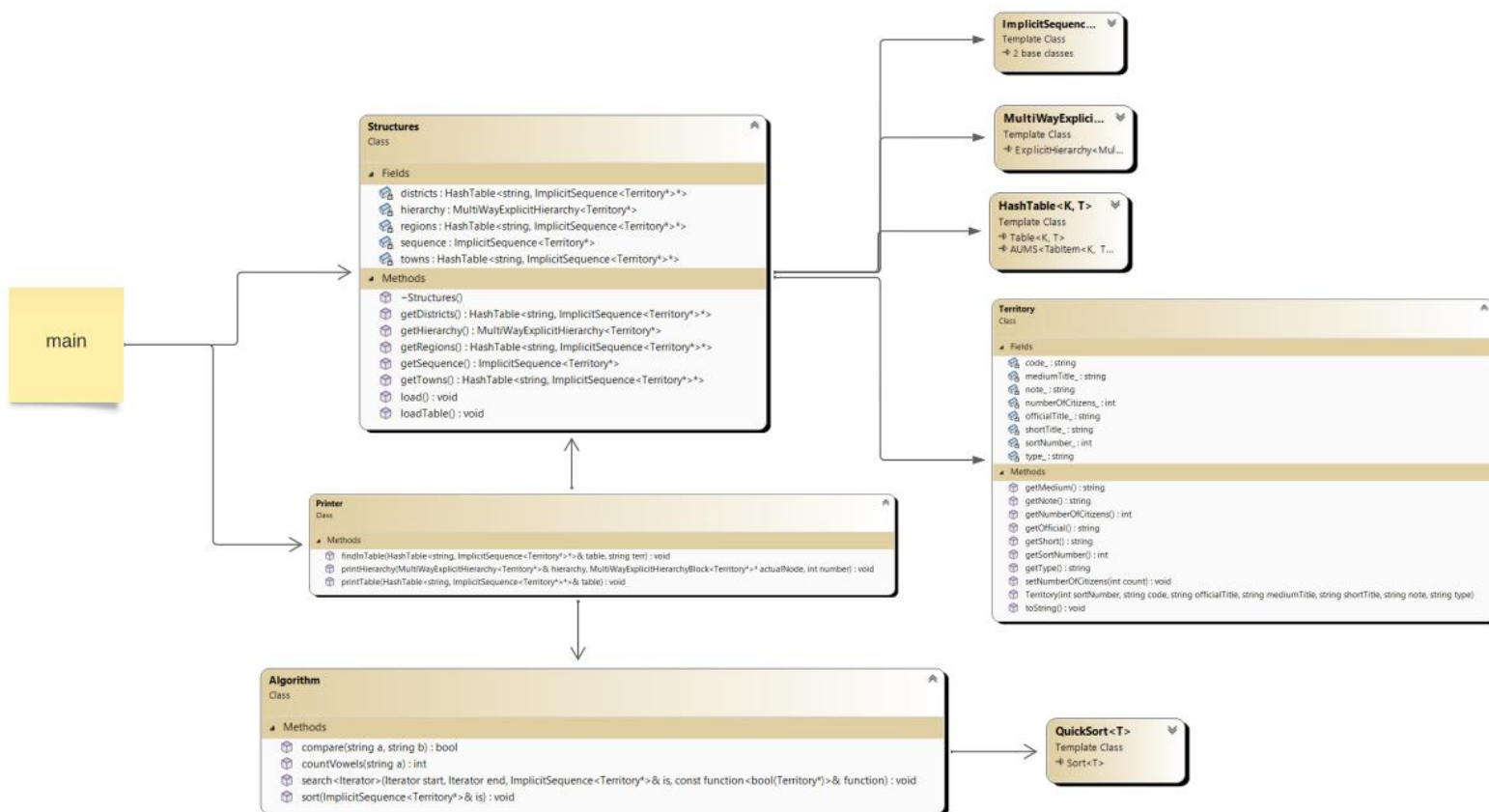
Návrh aplikácie

Aplikácia je vložená ako samostatný projekt do riešenia datastructures. Projekt je zložený zo 4 tried. V main-e prebieha slučka programu. Trieda Territory predstavuje jednu územnú jednotku. V triede Structures sú uložené údajové štruktúry a metódy na ich načítanie. Trieda Printer slúži na výpis prvkov. V triede Algorithm sa nachádzajú algoritmy z 1. a 4. úrovne.

Použité údajové štruktúry

hierarchia – viaccestná explicitná hierarchia
tabuľky – tabuľka s rozptýlenými záznamami
sekvencie – implicitná sekvencia

UML diagram



Používateľská príručka:

Po spustení programu sa zobrazí aktuálne územie, s možnosťami, ktoré môže používateľ využiť. Po vybraní príslušnej možnosti sa táto vykoná – ak vyžaduje dodatočné informácie, vypýta si ich od používateľa a znova vypíše aktuálne územie a možnosti alebo skončí.

Možnosti: vypísať všetky podradené územia
 prejsť na nadradené územie
 prejsť na podradené územie

vypísať podradené územia začínajúce na reťazec
 vypísať podradené územia obsahujúce reťazec
 vypísať podradené územia s počtom obyvateľov aspoň
 vypísať podradené územia s počtom obyvateľov maximálne
 vypísať územia daného typu
 vyhľadať v tabuľke
 ukončiť

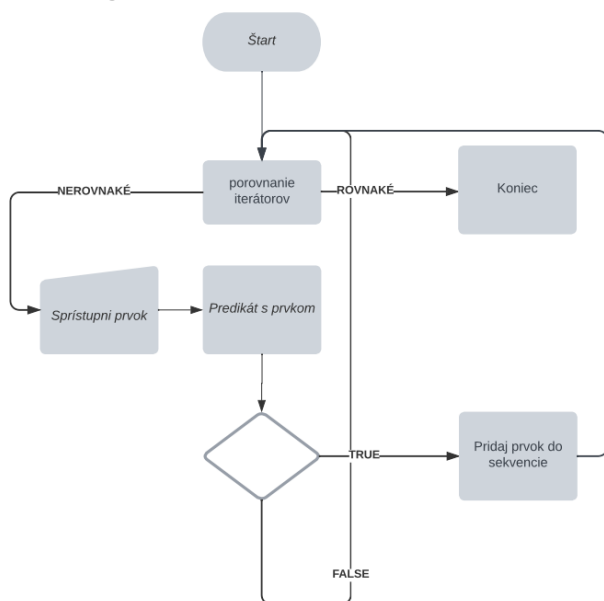
Po vypísaní území sa program spýta, či ich chce používateľ zoradiť a ak áno, má 3 možnosti zoradenia: podľa počtu samohlások, abecedne, podľa počtu obyvateľov.

Úroveň 1

- **Popis**

Algoritmus pracuje nad dvojicou iterátorov, prehľadáva prvky od prvého iterátora po druhý. Sprístupní prvok z iterátora a na základe výsledku funkcie s týmto prvkom rozhodne, či prvok začleniť do sekvencie. Ako predikát bola použitá lambda funkcia.

- **Uml diagram**



- **Programátorská príručka**

Algoritmus sa volá s dvojicou iterátorov, implicitnou sekvenciou a funkciou ktorá preberie objekt z iterátora a rozhodne, či ho algoritmus má alebo nemá zahrnúť do sekvencie. Sekvencia by mala byť prázdna, algoritmus ju neprepisuje, ale pridáva do nej prvky. Pre pridanie ďalšieho predikátu treba pridať do metódy PrintHierarchy nový predikát (zadefinovať mu číslo) a potom metódu zavolať s týmto číslom. Možné rozšírenie: Preťažiť metódu s ďalším parametrom - typom územia a vypisovať len tie.

Úroveň 2

- **Načítavanie**

Ako koreň hierarchie je vytvorené územie republika. Pre každý riadok súboru sa prečíta riadok, rozdelí sa na slová, vytvorí sa nová územná jednotka. Ak sa načítava kraj pridá sa ako syn koreňa. Ak sa načítava okres, porovnajú sa 3. a 4. znak v code s uloženými a ak sa nerovnajú prejde sa na ďalší kraj. Ak sa načítava obec, kontroluje sa 6. znak v code s uloženým, ak sa nerovná prejde sa na ďalší okres, tiež sa kontrolujú 3 a 4 znak s uloženými a ak sa nerovnajú prejde sa na ďalší kraj. To akému kraju/okresu sa aktuálne priraduje si program udržiava ako poradie a pri prejdení na ďalší sa rozumie zvýšiť poradie o 1. Program si tiež udržiava poradie aktuálne zaradovaného územia. Zložitosť načítavania je $O(n)$. Keďže územia označené ako Zahraničia nemajú časť note, program im namiesto note ukladá prázdny reťazec.

- **Iterátor hierarchie**

Program si neuchováva iterátor, pracuje s vrcholom, na ktorom sa aktuálne nachádza. Na posuv medzi vrcholmi využíva program funkcie vrchola (accessParent, accessSon).

- **Výpis**

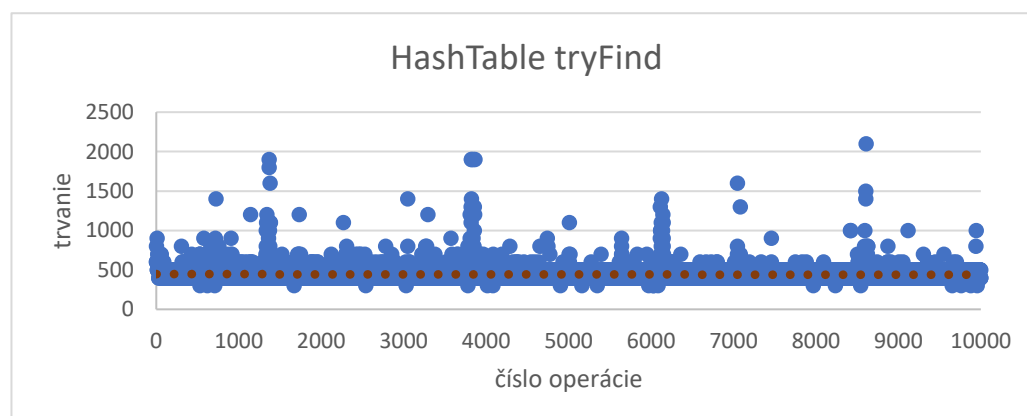
Na vypísanie prvkov je použitý algoritmus z 1. úrovne, pracuje nad iterátorom s priamym poradím.

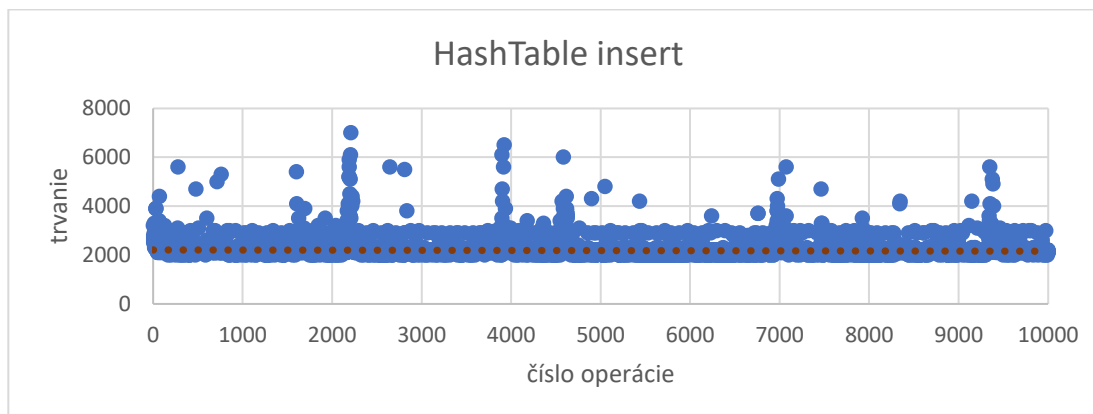
Úroveň 3

- **Popis**

Ako tabuľky boli použité tabuľky s rozptýlenými záznamami. Trieda Structures si uchováva jednu tabuľku pre každý typ územia. Ako kľúče sú použité oficiálne názvy území, hodnoty sú implicitné sekvencie. Ak sa nájdu dve územia s rovnakým názvom, obe sú zaradené do tejto sekvencie pod rovnaký názov.

- **Analýza**





Obidve operácie boli vykonané 10000 krát a ich trvanie je zaznamenané v tabuľkách. Červená prerušovaná čiara je trendová spojnica. Z grafov vyplýva, že obidve operácie majú konštantné zložitosti.

Úroveň 4

- **Popis**

Ako triediaci algoritmus je použitý quicksort. Vkladaná je do neho lambda funkcia, ktorá používa pomocné metódy na získanie počtu samohlások v slove a porovnanie dvoch slov. Pri porovnávaní sa ignorujú medzery a veľkosť písmena.

- **Porovnanie samohlások**

Dvojhlásky sú považované za dve samohlásky/ “ä” a “ô” za jednu.

- **Abecedné porovnanie**

- Dvojhlásky sú považované za dve písmená/ “ä” a “ô” za jedno. Pre každú dvojicu písmen sa písmeno vyhľadá v abecede a porovnávajú sa indexy. Písmená “ch”, “dz”, “dž” sú považované za jedno písmeno.

Bonus

- **Popis**

K územným jednotkám sú pridané aj počty ich obyvateľov, údaje boli čerpané z databázy DataCube. Exportované boli ako súbor csv, nech sú v rovnakom formáte ako vstupné súbory k semestrálnej práci. Načítavajú sa spolu so vstupnými súbormi – riadok z jedného súboru koreluje s riadkom z druhého súboru.

- **Pridaná hodnota**

Pri každom vypísaní územnej jednotky sa vypíše aj jej počet obyvateľov. Boli vytvorené 2 nové predikáty – nájdi územia s počtom obyvateľov aspoň a najviac.