**Rozbor použitých tried/štruktúr :**

* Trieda **Citac :**  Najdôležitejšia trieda aplikácie, načítava dáta z .csv súborov,

reaguje na vstupy od použivateľa. Obsahuje hlavné metódy na riešenie úloh v zadaní. Použité **údajové štruktúry**: **okresyTAB** – HashTable, v ktorej klúčom je názov okresu, a dátami sú príslušné smerníky na objekty triedy Okres.

**krajeTAB** – HashTable , klúč: názov kraja, dáta: smerník na Kraj.

HashTable som použil z dôvodu rýchleho vyhľadávania a vkladania, bolo treba predefinovať hash funkciu aby dokázala zahashovať string.

* Trieda **Okres:** predstavuje konkrétny okres, má svoj názov ( reálne ho mať nemusel, pretože sa drží v tabulke okresyTAB ako klúč, ale pre jednoduchosť ho má), taktiež drži v sebe konkrétne dáta pre daný okres, načítané v triede Citac.

Tieto dáta drží v štruktúrach: **zoznamUCH** – ArrayList prvkov ZaznamUCH\*, kedže pri načítavaní dát čítam postupne roky v poradí (2001,2002...,2016),tak napr na indexe 0 mám ZaznamUCH pre rok 2001 a pod.. to znamená O(1) pre vyhľadanie konkrétneho záznamu pre daný rok.

**zoznamNEZ** – ArrayList prvkov ZaznamNEZ\*, rovnaký dôvod použitia, ale zmena je v indexovaní , na **indexe 0** držím odkaz na ZaznamNEZ pre hodnoty mieri nezamestnanosti. **index 1** – Ekonomicky aktívni, **index 2** – Disponibilný počet.

* Trieda **ZaznamUCH** – predstavuje konkrétny záznam z údajov o Uchádzačoch o zamestnanie. Nepotrebuje žiadnu štruktúru.
* Trieda **ZaznamNEZ** – predstavuje konkrétny riadok pre okres(napr. o miere Nezamestnanosti), zo súboru o Nezamestnanosti. Použitá **štruktúra:** ArrayList prvkov udajNEZ , dôvod použitia: index 0 – údaje pre rok 2001 , index 1 – údaje pre rok 2002 atď.. to znamená O(1) pre získanie konkrétnych hodnôt.
* Trieda **UdajNEZ –** konkrétny údaj z údajov o Nezamestnanosti.
* Trieda **Kraj** – predstavuje konkrétny kraj. Použitá štruktúra: Na správne priradenie okresov do kraja som použil HashTable kde kľúčom je názov kraja a dáta sú smerníky na ArrayList<Okres\* >, kde tento arraylist obsahuje odkazy na všetky okresy patriace do kraja.
* Trieda **SortKey** – trieda predstavuje len dvojicu string , hodnota. Kedže sorting riešim cez triedenie Neusporiadanej sekvenčnej tabuľky, (kde kľúč musí byť unikátny), trieda predstavuje klúč v tejto tabuľke , má definované potrebné operátory na fungovanie sortu (< , > , == ).
* trieda **Vstup** – slúži len na komunikáciu s použivateľom a potrebné vstupné dáta posiela triede Citac, ktorá už tieto dáta spracuje.

**Zložitosti algoritmov zo zadania:**

1. **Vyhľadanie okresu podľa názvu a zobrazenie štatistík pre rozpätie rokov**: keďže okresy držím v HashTable, v ktorej najväčšia dĺžka zreťazenej tabuľky je 5 , tak môžeme hovoriť o zložitosti O(1),

Následne získanie údaju pre jeden rok z rozsahu má zložitosť O(1) pretože viem vypočítať na ktorom indexe v danom ArrayListe držim údaje pre daný rok, to znamená pre daný rozsah rokov to je O(k), kde k je počet rokov v rozsahu. 🡪 celková zložitosť O(1 + k).

1. **Zoradenie okresov v danom roku pre ženy/muži , vzostupne/zostupne :**

Najskôr naplním Neusporiadanú Sekvenčnú tabuľku (kde kľúčom je už spomínaný SortKey) : podobne ako v 1., ale pre každý okres, to znamená O(n) , n = počet okresov .

Následne túto tabuľku triedim HeapSort algoritmom: O(n log n).

Na koniec stačí okresy vypísať : O(n) 🡪 celková zložitosť O(2n+n log n).

1. **Vyhľadanie okresu s najmenším/najväčším údajom v danom rozpätí rokov:**

Prejdem raz HashTable a porovnávam požadované hodnoty, najväčšiu , prípadne najmenšiu si zapamätám. Prejdenie Hashtable O(n) 🡪 celková zložitosť O(n\*k) kde k = počet rokov, n= počet okresov

1. **Vyhľadanie kraju a zobrazenie sumárnych štatistík pre rozpätie rokov:**

Vyhľadanie kraju má zložitosť O(1) , pretože HashTable.

Následne už len stačí prejsť jeden ArrayList v ktorom sú okresy pre príslušný kraj a nasčítať a zobraziť príslušné hodnoty to znamená zložitosť : O(n\*k) , kde n je počet okresov v kraji a k je počet rokov v rozsahu.

1. **Zoraďenie krajov v danom roku pre ženy/muži, vzostupne/zostupne:**

Takisto ako v 3. najskôor naplním jednu Neusporiadanú tabuľku, kde kľúčom SortKey, ktorý má názov kraju a príslušnú hodnotu , túto hodnotu musím najskôr vypočítať ( podľa zadania) : ciže výpočet tejto hodnoty má zložitosť O(n) , n = počet okresov v kraji. To znamená naplnenie tabuľky 🡪 O(k\*n),k = počet krajov , n = počet okresov v kraji.

Následne sa táto tabuľka utriedi HeapsSort-om O(k log k) a vypíše O(k).

Celková zložitosť 🡪 O(k\*n + k log k + k).

1. **Vyhľadanie kraju s najväčším,najmenším údajom v danom rozpätí rokov:**

Podobne ako v 3. Raz prejdem HashTabulku krajov, a vypočítam príslušnú hodnotu , najväčšiu ,prípadne najmenšiu si zapamätám. Ale v tomto prípade ma výpočet hodnoty pre jeden kraj zložitosť O(n\*r), n = počet okresov v kraji , r = počet rokov. To znamená pre všetky kraje zložitosť O(k\*(n\*r)) .