Laboratorium 4 – Struktury danych

Zad 1

Stwórz program, który znajdzie wszystkie unikatowe nazwiska występujące w pliku podanym przez użytkownika. W tym celu w pakiecie *kolekcje.posl.pl* stwórz klasę uruchamieniową (z metodą main()) o nazwie *Unikaty*, a w niej metodę main, która wczyta zwartość pliku *baza_numerów.txt* a następnie, na podstawie wyszuka zbiór unikatowych nazwisk. Użyj do tego odpowiedniej kolekcji. Wyniki wyświetl na ekranie i zapisz do pliku *unikaty.txt*

Zad 2

Często spotykanym problemem jest stworzenie histogramu określonych wartości, np. zliczenie ile razy dany student był na zajęciach, przy założeniu że dziennik obecności dostępny jest w wersji elektronicznej. Innym przykładem jest policzenie średniej ocen studentów z każdego z przedmiotów. Do tej samej klasy problemów należy policzenie częstości występowania słów w dokumencie tekstowym.
Zrealizuj poniższe zadanie

1. Stwórz klasę *Czestosc* a w niej metodę *main,* w której umieść cały kod porgramu
2. Zastanów się i dobierz odpowiednią kolekcję, którą najlepiej jest wykorzystać do policzenia częstości występowania słów w pliku tekstowym.
3. Dla dokumentu tekstowego zaproponowanego przez prowadzącego policz częstość słów, w tym celu wykorzystaj klasę Scanner. Klasa Scanner domyślnie separuje każdy nowy wyraz
4. Po zliczeniu częstości słów, wypisz na ekranie częstość występowania pierwszych dziesięciu (według kolejności alfabetycznej) słów. Wyniki umieść w sprawozdaniu

Zad 3

Korzystając z zestawu kolekcji dostępnych w Javie (pakiet java.collections) zastanów się i stwórz książkę telefoniczną. W tym celu stworz pakiet ksiazkaTelefoniczna, a w nim klasę *Osoba* składającą się z pól *imię, nazwisko, nrTelefonu, id* oraz odpowiednich getterów i seterów. Pole id powinno jednoznacznie identyfikować każdą osobę w bazie i powinno być kolejną liczba naturalną generowaną podczas tworzenia obiektu, unikatową dla każdej nowo wygenerowanej osoby. Zastanów się jak to osiągnąć!

Stwórz klasę *KsiazkaTelefoniczna* posiadającą metody:

List<Osoba> getByImie(String imie) – metoda powinna zwracać listę osób o podanym imieniu

List<Osoba> getByNazwisko(String imie) – metoda powinna zwracać listę osób o podanym nazwisku

void dodajOsobe(Osoba o) – metoda służy do dodania nowej osoby do bazy

Osoba remove(long id) – metoda służy do usuwania z bazy osoby o określonym id.

void *read(File file)* – metoda powinna wczytać zawartość książki telefonicznej z pliku

Stwórz klasę *Main*, która w pierwszym kroku tworzy nową książkę telefoniczną, a następnie wczytuje z podanego pliku jej zawartość (plik dostępny na stronie prowadzącego)

Plik wygenerowany jest w formacie CSV w kolejności *imie ; nazwisko ; nrTelefonu*

Zwróć uwagę że pierwszy wiersz pliku zawiera linijkę opisującą kolumny więc należy ją pominąć przy parsowani.

Następnie wyszukaj w bazie osób o imieniu Anna i wyświetl je na ekranie.

UWAGA: Zwróć uwagę że może wystąpić wiele osób o wybranym imieniu i nazwisku, dlatego zastanów się jaką wybrać strukturę danych.

Przydatne klasy narzędziowe:

1. ArrayList<T> - dynamiczna tablica
2. LinkedList<T> - dynamiczna tablica – inna implementacja – dla łatwego dodawania i usuwania obiektów
3. Map<K,V> - interfejs dla słownika składającego się z par klucz, wartość
4. HashMap<K,V> słownik składający się z par klucz – wartość określonego typu
5. HashSet<T> zbiór wartości unikatowych (nie ma możliwości zapisania dwóch identycznych numerów)
6. TreeSet<t> - zbiór automatycznie posortowanych wartości (sortowanie odbywa się na etapie wkładania do zbioru)
7. SortedMap<K,V> interfejs posortowanego słownika składającego się z par klucz – wartość określonego typu
8. TreeMap<K,V> - implementacja słownika SortedSet
9. java.util.Comparator – umożliwia definiowanie własnych komparatorów przydatnych do sortowania