

Zaawansowane programowanie obiektowe

Lab. 6

(Java 8+, Date and Time API, Guava)

1. (1 pkt) W pliku dane.txt (do ściągnięcia lokalnie) są dane osobowe:
imię nazwisko kraj zarobki
Należy wypisać sumę zarobków 2 najlepiej i 2 najgorzej zarabiających Polaków (PL)
– prawidłowy wynik: 101999 i 38900.
Ponadto dla każdego kraju występującego w pliku należy wypisać liczbę osób, w kolejności pojawiania się krajów w kolejnych wierszach.
Czyli: PL - 6, DE - 2, CZ - 3, US - 2.

Wykorzystaj konstrukcje Javy 8 (strumienie, wyrażenia lambda).
Do odczytania pliku tekstowego poszukaj odpowiedniej metody klasy Files (leniwie wstawiającej linie tekstu do strumienia).
Możesz 3 razy tworzyć strumień (z wejściowego pliku), nie więcej. W wybranych miejscach użyj peek() do podejrzenia zawartości strumienia (cel: debugowanie kodu... a może jeszcze do czego?).

2. (0,5 pkt) Dany jest słownik "counts" liczników wystąpień stringów (np. słów w tekście). Można go uaktualnić (zwiększyć licznik dla elementu "word" o 1) w następujący sposób:
counts.merge(word, 1, Integer::sum);
(przeczytaj <https://docs.oracle.com/javase/8/docs/api/java/util/Map.html#merge-K-V-java.util.function.BiFunction->)

Wykonaj to samo bez użycia metody merge na 4 sposoby:

- a) z użyciem metody containsKey,
- b) z użyciem metody get oraz sprawdzania null-a,
- c) z użyciem getOrDefault,
- d) z użyciem putIfAbsent.

Przetestuj napisany kod z użyciem JUnit.

3. (0,5 pkt) Korzystając z Date and Time API Javy 8 policz:
 - a) ile dni trwała II wojna światowa (wliczając zarówno 1.09.1939, jak i 8.05.1945);
 - b) którego dnia i miesiąca wypada 68-my dzień roku 2016;
 - c) ile razy suma cyfr liczby godzin i minut, w formacie 24-godzinnym hh:mm, wynosiła dokładnie 15, jeśli ograniczamy się do przedziału czasowego od 11:45 rano do 22:30 wieczorem (przykłady: 17:52, 21:39); napisz testy;
 - d) ile w swoim życiu przeżyłeś/aś dni 29 lutego. Nie zakładaj z góry, że np. bieżący rok nie jest przestępny.

4. (BISEKCJA, LISTA LENIWA) (1 pkt)

Algorytm bisekcji znajduje np. miejsce zerowe funkcji w danym przedziale. W takim przypadku wymagamy, aby funkcja była monotoniczna w tym przedziale i w jednym z krańców przedziału osiągała wartość ujemną, a w drugim wartość dodatnią.

Proszę napisać program realizujący algorytm bisekcji dla dowolnej prostej funkcji (np. kwadratowej), z wykorzystaniem `Collections.binarySearch(...)`. Uwaga: nie chcemy tablicować wszystkich wartości funkcji (z zadany krokiem, np. 0,001) w danym przedziale, gdyż mogłoby to być zbyt kosztowne. Niech funkcja będzie na bieżąco obliczana w danym punkcie.

Wskazówki:

- użyj listy leniwej (poszukaj „guava lazy list” etc., np. na stackoverflow) z biblioteki Guava,
- do doboru funkcji matematycznej i przedziału wejściowego pomocna może być strona: <https://www.desmos.com/calculator>