

Laboratorium 3: Kolekcje.

Zadania:

1. Utwórz klasę *LiczbaNaturalna*, zawierającą następujące atrybuty:

- a. **wartosc**
- b. **suma** – suma cyfr,
- c. **iloczyn** – iloczyn jej cyfr,
- d. **parzysta** – status, czy jest to liczba parzysta czy nie.

Napisz program, który utworzy (pustą) listę obiektów tego typu, a następnie:

[3p]

- a) Wypełni listę danymi wprowadzonymi z klawiatury lub z pliku (tylko liczby, pozostałe atrybuty mają się wyliczać). Podczas dodawania należy sprawdzać czy dany obiekt już nie znajduje się w kolekcji, aby nie było duplikatów. Podanie wartości nie-liczbowej kończy wczytywanie.
- b) wypisze dane wszystkich obiektów z kolekcji w kolejności wprowadzania danych,
- c) wypisze obiekty posortowane rosnąco według liczby, sumy cyfr, iloczynu cyfr,
- d) wypisze te obiekty, których suma cyfr jest większa od średniej sumy cyfr wszystkich liczb,
- e) usunie z kolekcji te obiekty, które są liczbami nieparzystymi i ponownie wypisze zawartość i rozmiar kolekcji.

Wskazówki: W klasie *Liczba* nadpisz metody *toString()* oraz *equals()*. W programie głównym wykorzystaj *Scanner*, *Collections.sort*, *Comparator* oraz *Iterator*.

2. Napisz program zliczający wystąpienia poszczególnych słów w pliku tekstowym. Dla ułatwienia można przyjąć, że plik nie zawiera znaków interpunkcyjnych. Program ma wypisać wyniki:

[2p]

- a) posortowane alfabetycznie (rosnąco),
- b) posortowane według częstości wystąpień (malejąco).

Wskazówki: Wykorzystaj *mapy*, *Scanner*, *Collections.sort*, *Comparator*.

3. Dana jest lista napisów. Wypisz listę posortowaną:

[2p]

- a) alfabetycznie,
- b) wg długości napisów,
- c) wg ilości samogłosek w napisach,
- d) najpierw słowa zawierające literę „a”, potem wszystkie inne.

4. Korzystając z kolekcji w języku Java napisz programy generujące następujące struktury:

[2p]

- a) listę kwadratów liczb od x do y (gdzie $x < y$), ale tylko nieparzystych i podzielnych przez 9
- b) listę zawierającą n pierwszych wyrazów ciągu Fibonacciego
- c) mapę typu `Map<Integer, String>` zawierającą n kolejnych liczb naturalnych (1: "1", 2: "2" itd.)

5. Napisz program, który wczytuje plik tekstowy, a następnie zlicza długość każdego słowa. Wynikiem działania programu jest mapa typu `Map<Integer, List<String>>`, gdzie liczbie znaków jest przypisana lista słów o takiej długości (np. 1: [„i”, „a”], 2: [„bo”, „to”, „ja”], 3: [„ale”, „gdy”] itd.)

[2p]

6. Napisz program, który dla danej listy liczb całkowitych zwróci string zawierający wszystkie liczby oddzielone przecinkami. Dodatkowo, liczby parzyste mają być poprzedzone literą p, a nieparzyste literą n. Np. dla listy (1,2,3,4) powinniśmy otrzymać „n1,p2,n3,p4”.

[1p]