Zadanie 12 *26, 28 stycznia 2021 r.*

KURS JĘZYKA JAVA

PRZETWARZANIE AGREGUJĄCE KOLEKCJI

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

Zadanie 1.

W pliku `dane1.txt` zapisane są liczby całkowite z zakresu od 1 do 10^9. Format tego pliku jest następujący: w jednej linii może być zapisana co najwyżej jedna liczba (jeśli linia nie zawiera liczby to nazywamy ją linią pustą); liczba jest zapisana w systemie dziesiętnym bez zer wiodących; białe znaki (space i tabulacje) przed liczbą, za liczbą oraz w linii pustej należy zignorować; na końcu każdej linii może się znajdować komentarz, rozpoczynający się od sekwencji "//"(linia pusta także może zawierać komentarz).

Przeczytaj plik z danymi wiersz po wierszu, sprawdź za pomocą wyrażeń regularnych czy format pliku jest zgodny ze specyfikacją (jeśli nie, to zgłoś wyjątek) i wczytane liczby wstaw do kolekcji `ArrayList<Integer>`. Zastosuj konstrukcję *try-with-resources* przy czytaniu danych z pliku:

```
try (BufferedReader br = new BufferedReader(new FileReader(path))) {
    for (String ln = br.readLine(); ln != null; ln = br.readLine()) {
        ...
    }
}
catch (Exception ex) { ... }
```

Następnie za pomocą operacji agregujących na stumieniu `Stream<>` (wywołanie metody `stream()` na kolekcji) i wyrażeń lambda wykonaj następujące polecenia:

1. wypisz liczby z kolekcji uporządkowane od największej do najmniejszej wartości;
2. wypisz te liczby z kolekcji, które są liczbami pierwszymi;
3. wypisz sumę wszystkich liczb z kolekcji, które są < 1000.
4. wypisz ile spośród wszystkich liczb w kolekcji jest podzielnych przez 13.

Zadanie 2.

W pliku `dane2.txt` zapisane są trójkąty w postaci długości trzech boków (3 dodatnie liczby rzeczywiste). Format tego pliku jest następujący: w jednej linii może być zapisany co najwyżej jeden trójkąt, czyli trzy liczby rzeczywiste oddzielone białymi znakami (jeśli linia nie zawiera liczby to nazywamy ją linią pustą); liczby są zapisane w systemie dziesiętnym z opcjonalną

częścią ułamkową po kropce dziesiętnej; białe znaki (space i tabulacje) na początku linii, na końcu linii oraz w linii pustej należy zignorować; na końcu każdej linii może się znajdować komentarz, rozpoczynający się od sekwencji "//" (linia pusta także może zawierać komentarz).

Przeczytaj plik z danymi wiersz po wierszu, sprawdź za pomocą wyrażeń regularnych czy format pliku jest zgodny ze specyfikacją (jeśli nie to zgłoś wyjątek) i wczytane trójkąty wstaw do kolekcji `LinkedList<Trojkat>`, gdzie `Trojkat` to zdefiniowana klasa do reprezentacji trójkąta (w konstruktorze sprawdź poprawność danych, czyli długości boków). Zastosuj konstrukcję *try-with-resources* przy czytaniu danych z pliku.

Następnie za pomocą operacji agregujących na stumieniu `Stream<>` (wywołanie metody `stream()` na kolekcji) i wyrażeń lambda wykonaj następujące polecenia:

1. wypisz trójkąty z kolekcji uporządkowane od najmniejszego do największego obwodu;
2. wypisz te trójkąty z kolekcji, które są trójkątami prostokątnymi;
3. wypisz ile spośród wszystkich trójkątów w kolekcji jest równobocznych;
4. wypisz dwa trójkąty z kolekcji, których pola są odpowiednio najmniejsze i największe.

Zadanie 3.

Zdefiniuj wyliczenie `Pierwiastki` dla 18 pierwszych pierwiastków chemicznych z *tablicy Mendelejewa* (od helu do argonu). Każdy pierwiastek w wyliczeniu ma mieć nazwę zgodną z symbolem chemicznym; dodatkowo każdy pierwiastek powinien posiadać pełną polską nazwę, liczbę atomową i masę atomową.

W programie testującym wypisz wszystkie zdefiniowane pierwiastki, używając nadpisanej metody `toString()`.