

Firma software'owa prowadzi projekty w różnych językach programowania.
Plik Prpgrammers.tsv z katalogu {user.home} zawiera informacje o programistach w postaci:

język1<TAB>nazwisko(1)<TAB>nazwisko(2)<TAB> itd
język2<TAB>nazwisko(1)<TAB>nazwisko(2)<TAB> itd
...

Stworzyć klasę ProgLang, mającą:

- konstruktor ProgLang(String nazwaPliku), w którym następuje wczytanie pliku o podanej nazwie,
- metodę getLangsMap() - zwracająca mapę, w której pod kluczem nazwa języka znajduje się kolekcja programistów tego języka,
- metodę getProgsMap() - zwracająca mapę, w której pod kluczem nazwisko programisty znajduje się kolekcja języków, w których programuje,
- metodę getLangsMapSortedByNumOfProgs() - zwracająca mapę z wejściami język -> kolekcja programistów. uporządkowaną malejąco według liczby osób znających poszczególne języki, w przypadku równej liczbę porządek jest alfabetyczny wg nazw języków,
- metodę getProgsMapSortedByNumOfLangs() - zwracająca mapę z wejściami programista -> kolekcja języków, uporządkowaną malejąco wg liczby języków znanych programiści; w przypadku równej liczby porządek jest alfabetyczny wg nazwisk,
- metodę getProgsMapForNumOfLangsGreaterTha(int n) - zwracająca mapę z wejściami programista -> kolekcja języków, dla których liczba języków jest większa od podanego n.
- metodę **sorted(...)**, wołaną z argumentami mapa i lambda-wyrażenie. Metoda zwraca posortowaną wersję dowolnej mapy przekazanej jako pierwszy argument, a porządek sortowania jest określony przez lambda wyrażenia, podane jako drugi argument,
- metodę **filtered(...)** z argumentami: dowolna mapa i lambda. Metoda zwraca mapę, która zawiera tylko te wejścia z przekazanej jako pierwszy argument mapy, które spełniają warunek podany jako drugi argument (lambda z wynikiem typu boolean).

Metod **sorted(...)** lub **filtered(...)** użyć w oprogramowaniu innych, odpowiednich, metod klasy. Mają one jednak ogólniejsze znaczenia, bo mogą być używane dla dowolnych innych map z warunkami sortowania czy filtrowania, zadawanymi przez właściwe w danych przypadkach lambda.
Uwaga: uniwersalność metod sorted i filtered)możliwość ich zasobędzie sprawdzana

Poniższa klasa Main (ktorej w projekcie nie wolno modyfikować):

```
import java.io.*;

public class Main {

    public static void main(String[] args) throws IOException {
        ProgLang pl = null;
        try {
            pl = new ProgLang(System.getProperty("user.home") + "/Programmers.tsv");
        } catch (Exception exc) {
            System.out.println("Wadliwy konstruktor: " + exc);
        }
        System.out.println("@1 Mapa językow:");
        pl.getLangsMap().forEach((k,v)->System.out.println(k+ " = " + v));
        System.out.println("@2 Mapa programistów:");
        pl.getProgsMap().forEach((k,v)->System.out.println(k+ " = " + v));
        System.out.println("@3 Języki posortowane wg liczby programistów:");
        pl.getLangsMapSortedByNumOfProgs()
            .forEach((k,v)->System.out.println(k+ " = " + v));
        System.out.println("@4 Programiści posortowani wg liczby języków:");
        pl.getProgsMapSortedByNumOfLangs()
            .forEach((k,v)->System.out.println(k+ " = " + v));
        System.out.println("@5 Oryginalna mapa języków niezmieniona:");
        pl.getLangsMap().forEach((k,v)->System.out.println(k+ " = " + v));
        System.out.println("@6 Oryginalna mapa programistów niezmienione:");
        pl.getProgsMap().forEach((k,v)->System.out.println(k+ " = " + v));
        System.out.println("@7 Mapa programistów znających więcej niż 1 język:");
        pl.getProgsMapForNumOfLangsGreaterTha(1)
            .forEach((k,v)->System.out.println(k+ " = " + v));
        System.out.println("@8 Oryginalna mapa programistów nie jest zmieniona:");
        pl.getProgsMap().forEach((k,v)->System.out.println(k+ " = " + v));
    }

}
```

dla następującej zawartości pliku Programmers.tsv:

Groovy	Z	Y	X	D		
Java	V	B	C	D	A	Z
C++	G	J	H			
C#	P	S	Q	V	D	
Scala	A	D	A			

(uwaga: sepratorami są znaki tabulacji)

wyprowadzi na konsolę wynik:

```
@1 Mapa językow:
Groovy = [Z, Y, X, D]
Java = [V, B, C, D, A, Z]
C++ = [G, J, H]
C# = [P, S, Q, V, D]
Scala = [A, D]

@2 Mapa programistów:
Z = [Groovy, Java]
Y = [Groovy]
X = [Groovy]
D = [Groovy, Java, C#, Scala]
V = [Java, C#]
B = [Java]
C = [Java]
A = [Java, Scala]
G = [C++]
J = [C++]
H = [C++]
P = [C#]
S = [C#]
Q = [C#]

@3 Języki posortowane wg liczby programistów:
Java = [V, B, C, D, A, Z]
C# = [P, S, Q, V, D]
Groovy = [Z, Y, X, D]
C++ = [G, J, H]
Scala = [A, D]

@4 Programiści posortowani wg liczby języków:
D = [Groovy, Java, C#, Scala]
A = [Java, Scala]
V = [Java, C#]
Z = [Groovy, Java]
B = [Java]
C = [Java]
G = [C++]
H = [C++]
J = [C++]
P = [C#]
Q = [C#]
S = [C#]
X = [Groovy]
Y = [Groovy]

@5 Oryginalna mapa języków niezmieniona:
Groovy = [Z, Y, X, D]
Java = [V, B, C, D, A, Z]
C++ = [G, J, H]
C# = [P, S, Q, V, D]
Scala = [A, D]

@6 Oryginalna mapa programistów niezmienione:
Z = [Groovy, Java]
Y = [Groovy]
X = [Groovy]
D = [Groovy, Java, C#, Scala]
V = [Java, C#]
B = [Java]
C = [Java]
A = [Java, Scala]
G = [C++]
J = [C++]
H = [C++]
P = [C#]
S = [C#]
Q = [C#]

@7 Mapa programistów znających więcej niż 1 język:
Z = [Groovy, Java]
D = [Groovy, Java, C#, Scala]
V = [Java, C#]
A = [Java, Scala]

@8 Oryginalna mapa programistów nie jest zmieniona:
Z = [Groovy, Java]
Y = [Groovy]
X = [Groovy]
D = [Groovy, Java, C#, Scala]
V = [Java, C#]
B = [Java]
C = [Java]
A = [Java, Scala]
G = [C++]
J = [C++]
H = [C++]
P = [C#]
S = [C#]
Q = [C#]
```

Ważne uwagi:

- zgodność informacji wyjściowej z oczekiwanym wynikiem (w tym kolejność pokazywania danych) jest istotna - wynika z zastosowania odpowiednich map i innych klas kolekcyjnych, za niezgodność w którymkolwiek z punktów 1-8 będą odejmowane 2 punkty,
- uniwersalność metod **sorted** i **filtered** (możliwość ich zastosowania dla innych niż w zadaniu map) będzie sprawdzana; brak uniwersalności każdej z metod skutkuje odjęciem 4 punktów,
- w klasie ProgLang nie wolno używać surowych typów.