

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ**

Ордена Трудового Красного Знамени

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Московский технический университет связи и информатики»

Кафедра «Математическая Кибернетика и Информационные технологии»

Дисциплина «Информационные технологии и программирование»

Лабораторная работа №6

«Коллекции в Java»

Выполнила:

Студентка группы БВТ2303

Морозова Ольга

Цель работы:

Изучение коллекций и применение полученных знаний на практике на языке программирования Java.

Ход работы:

Задание 1.

Напишем программу, которая считывает текстовый файл и выводит на экран топ-10 самых часто встречающихся слов в этом файле с помощью класса Map.

```
TopWords.java x Work.java Stack.java SellsHashMap.java
1 package Lab6.Exercise1;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.util.*;
6 import java.util.regex.*;
7
8 public class TopWords {
9     public static void main(String[] args) throws FileNotFoundException{
10         String filePath = "c:/Minilynx/Second/ITaP/Lab6/Exercise1/Text.txt";
11         File file = new File(filePath);
12         Scanner scanner = null;
13         try {
14             scanner = new Scanner(file);
15         } catch (NullPointerException e) {
16             System.out.println(e.getMessage());
17             System.out.println("Попытка обращения по ссылке равной null.");
18         } catch (FileNotFoundException e) {
19             System.out.println(e.getMessage());
20             System.out.println("Файл по указанному пути не существует.");
21         }
22         Map<String, Integer> map = new HashMap<>();
23         StringBuilder sb = new StringBuilder();
24         Pattern pat = Pattern.compile( regex: "[A-Za-zА-ЯЁа-яё]+([\\-']+[A-Za-zА-ЯЁа-яё]+)*");
25         Matcher mat;
26         while (scanner.hasNextLine()) {
27             mat = pat.matcher(scanner.nextLine());
28             while (mat.find()) {
29                 sb.append(mat.group());
30                 sb.replace( start: 0, sb.length(), str: String.valueOf(sb.charAt(0)).toUpperCase() + sb.substring( start: 1).toLowerCase());
31
32                 if (map.containsKey(sb.toString())) {
33                     map.put(sb.toString(), map.get(sb.toString()) + 1);
34                 } else {
35                     map.put(sb.toString(), 1);
36                 }
37                 sb.delete(0, sb.length());
38             }
39             scanner.close();
40             String[] max = new String[2];
41             int len = Math.min(map.size(), 10);
42             for (int i = 0; i < len; i++) {
43                 max[0] = "";
44                 max[1] = "0";
45                 for (Map.Entry<String, Integer> m : map.entrySet()) {
46                     if (Integer.parseInt(max[1]) < m.getValue()) {
47                         max[0] = m.getKey();
48                         max[1] = m.getValue().toString();
49                     }
50                 }
51                 System.out.println(max[0] + ": " + max[1]);
52                 map.remove(max[0]);
53             }
54         }
55     }
```

Задание 2.

Напишем обобщенный класс `Stack<T>`, который реализует стек на основе массива. Класс имеет методы `push` для добавления элемента в стек, `pop` для удаления элемента из стека и `peek` для получения верхнего элемента стека без его удаления

```
TopWords.java  Work.java  Stack.java  SellsHashMap.java

1  package Lab6.Exercise2;
2
3  public class Stack<T> {  2 usages
4      private T[] data;  6 usages
5      private int size;  10 usages
6
7      public Stack(int cap) {  1 usage
8          data = (T[]) new Object[cap];
9          size = 0;
10     }
11
12     public void push(T el) {  5 usages
13         if (size < data.length) {
14             data[size] = el;
15             size++;
16         } else {
17             System.out.println("Стек переполнен. Не получилось добавить " + String.valueOf(el));
18         }
19     }
20
21     public T pop() {  2 usages
22         if (size > 0) {
23             T taken = data[size - 1];
24             data[size - 1] = null;
25             size--;
26             return taken;
27         } else {
28             System.out.println("Стек пуст.");
29             return null;
30         }
31     }
32
33     public T peek() {  3 usages
34         if (size > 0) {
35             return data[size - 1];
36         } else {
37             System.out.println("Стек пуст.");
38             return null;
39         }
40     }
41 }
```

И класс, в котором проверяем его работу.

```
TopWords.java  Work.java ×  Stack.java  SellHashMap.java

1  package Lab6.Exercise2;
2
3  public class Work {
4      public static void main(String[] args) {
5          Stack<Integer> stack = new Stack<> (cap: 3);
6
7          System.out.println(stack.peek());
8
9          stack.push(1);
10         stack.push(2);
11         stack.push(3);
12         stack.push(4);
13
14         System.out.println(stack.pop());
15         System.out.println(stack.peek());
16
17         stack.push(5);
18
19         System.out.println(stack.pop());
20         System.out.println(stack.peek());
21     }
22 }
```

Задание 3.

Напишем программу для учета продаж в магазине. Программа позволяет добавлять проданные товары в коллекцию, выводить список проданных товаров, а также считать общую сумму продаж и наиболее популярный товар.

Для этого используем класс HashMap.

```
TopWords.java  Work.java  Stack.java  SellsHashMap.java ×

1  package Lab6.Exercise3;
2
3  import java.util.HashMap;
4  import java.util.Map;
5
6  public class SellsHashMap {
7      static HashMap<String, Integer> map = new HashMap<>(); 15 usages
8
9      public static void main(String[] args) {
10         checkSells();
11         System.out.println("Сумма продаж равна " + sumSells());
12         System.out.println(topSells());
13
14         sell( prod: "молоко");
15         sell( prod: "МОЛОКО", amount: 5);
16         sell ( prod: "арбуз");
17         sell ( prod: "Дыня", amount: 2);
18
19         checkSells();
20         System.out.println("Сумма продаж равна " + sumSells());
21
22         System.out.println(topSells());
23     }
24
25     public static void sell(String prod) { 2 usages
26         prod = String.valueOf(prod.charAt(0)).toUpperCase() + prod.substring( beginIndex: 1).toLowerCase();
27         if (map.containsKey(prod)) {
28             map.put(prod, map.get(prod) + 1);
29         } else {
30             map.put(prod, 1);
31         }
32     }
```

```

34     public static void sell(String prod, Integer amount) { 2 usages
35         prod = String.valueOf(prod.charAt(0)).toUpperCase() + prod.substring(beginIndex: 1).toLowerCase();
36         if (map.containsKey(prod)) {
37             map.put(prod, map.get(prod) + amount);
38         } else {
39             map.put(prod, amount);
40         }
41     }
42
43     public static void checkSells() { 2 usages
44         if (map.isEmpty()) {
45             System.out.println("Продаж не было");
46         } else {
47             for (Map.Entry<String, Integer> m : map.entrySet()) {
48                 System.out.println(m.getKey() + ": " + m.getValue().toString());
49             }
50         }
51     }
52
53     public static int sumSells() { 2 usages
54         if (map.isEmpty()) {
55             return 0;
56         } else {
57             int sum = 0;
58             for (Map.Entry<String, Integer> m : map.entrySet()) {
59                 sum += m.getValue();
60             }
61             return sum;
62         }
63     }

```

```

65  @ public static Map.Entry<String, Integer> topSells() { 2 usages
66      if (map.isEmpty()) {
67          System.out.println("Продаж не было");
68          return null;
69      } else {
70          String[] max = new String[] { "", "0" };
71          for (Map.Entry<String, Integer> m : map.entrySet()) {
72              if (Integer.parseInt(max[1]) < m.getValue()) {
73                  max[1] = String.valueOf(m.getValue());
74                  max[0] = m.getKey();
75              }
76          }
77          for (Map.Entry<String, Integer> m : map.entrySet()) {
78              if (max[0].equals(m.getKey())) {
79                  return m;
80              }
81          }
82          return null;
83      }
84  }
85  }

```

Вывод:

Мы изучили коллекции, существующие классы коллекций в Java и сами написали стек с использованием полученных знаний.

GitHub - https://github.com/MiniLynx13/ITaP_Lab6