

10.10

HashMap и TreeMap

В этом материале мы изучим ещё одну разновидность коллекций — `map`. В `map`'е может храниться соответствие ключей этим значениям,, при этом ключи должны быть обязательно уникальными (ключи хранятся в `set`, который мы изучали в прошлом видео).

Разберём простой пример. Создаём коллекцию `HashMap` по тем же правилам, что и `List` или `Set`. Например, мы создаём список товаров `goodsList`.

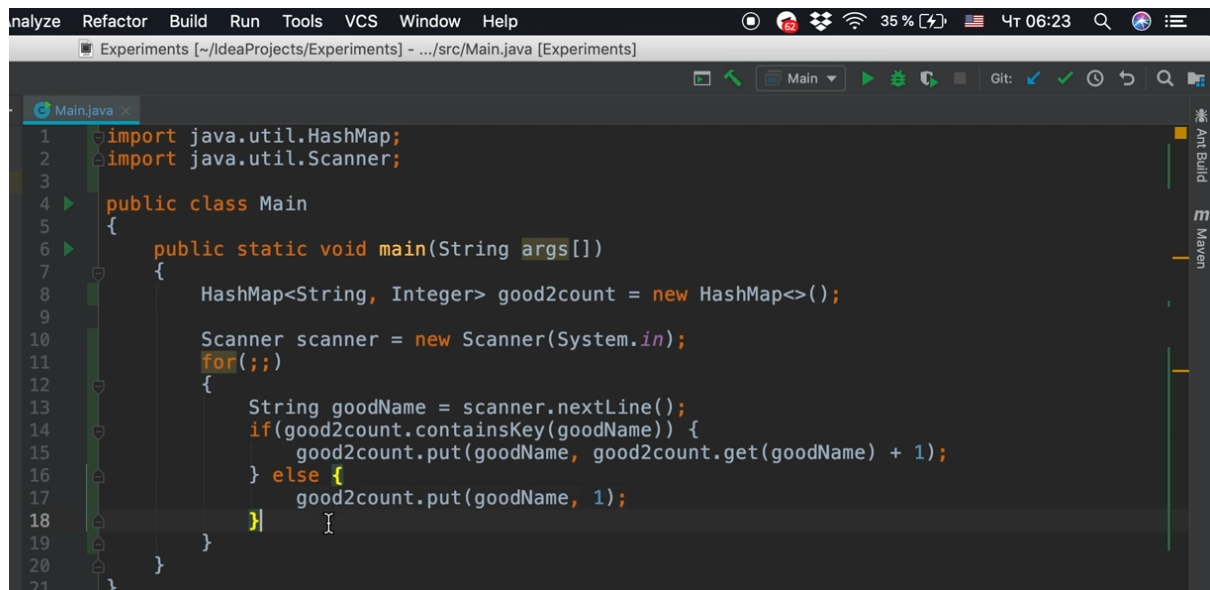
```
{  
    HashMap<String>, Integer good2count = new HashMap<>();
```

Напишем программу, которая будет из консоли выводить новый товар.

```
Scanner scanner = new Scanner(System.in);  
for(;;)  
    String goodName scanner.nextLine();
```

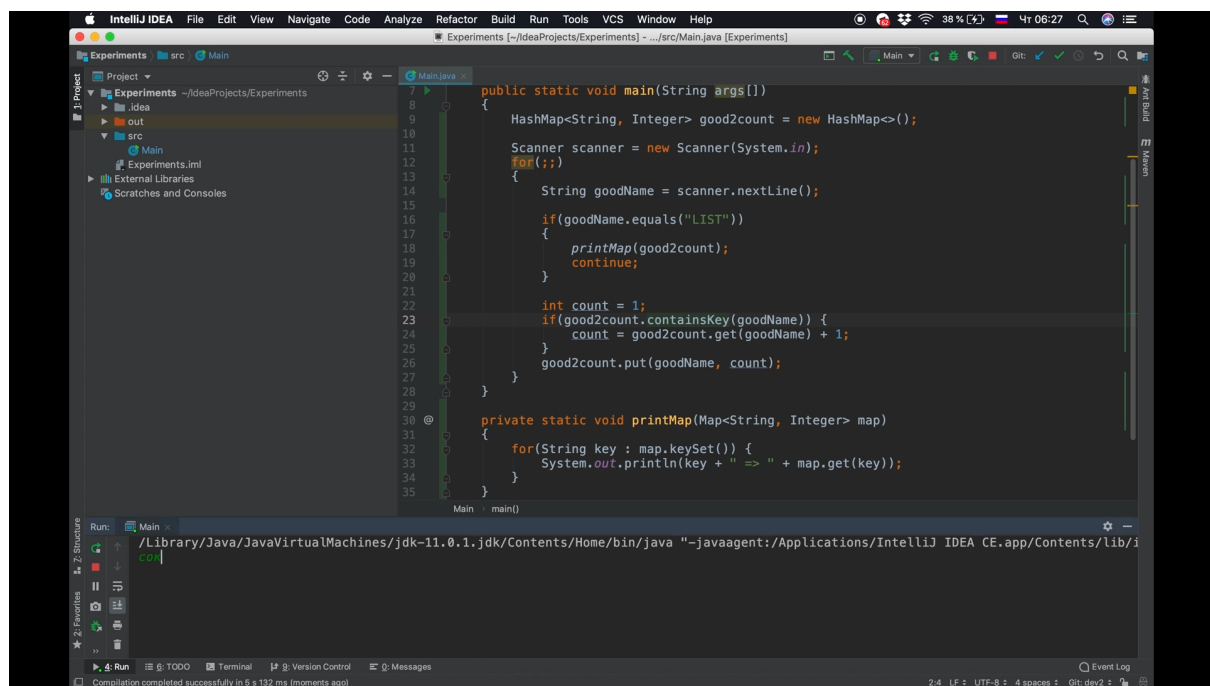
Добавляем проверку: проверяем, содержит ли `HashMap` ключ с именем данного товара. С помощью метода `put` (эквивалент добавления в коллекции `Map`, в коллекции `List` и `Set` это был метод `add`), у которого есть два параметра — ключ (`goodName`) и значение, которое нам неизвестно. Значение нам необходимо взять при помощи метода `get` и прибавить к нему единицу.

Если этого товара здесь нет, то через `else` добавляем в проверку этот товар со значением 1.



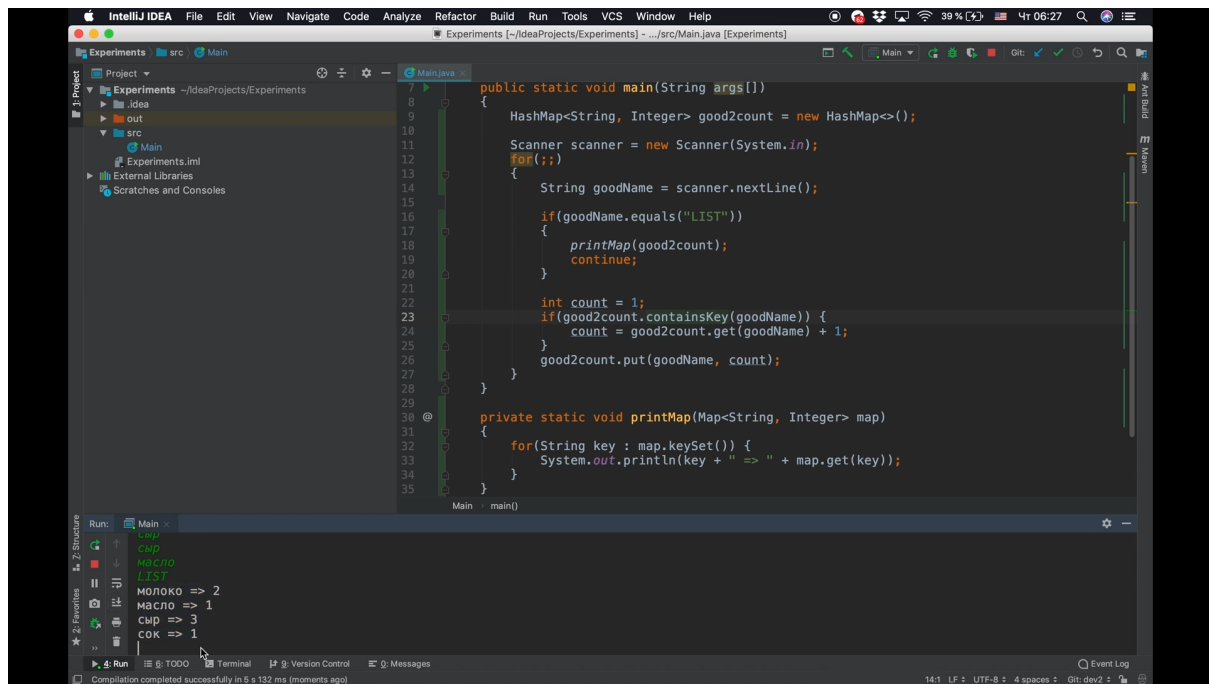
```
1 import java.util.HashMap;
2 import java.util.Scanner;
3
4 public class Main
5 {
6     public static void main(String args[])
7     {
8         HashMap<String, Integer> good2count = new HashMap<>();
9
10        Scanner scanner = new Scanner(System.in);
11        for(;;)
12        {
13            String goodName = scanner.nextLine();
14            if(good2count.containsKey(goodName)) {
15                good2count.put(goodName, good2count.get(goodName) + 1);
16            } else {
17                good2count.put(goodName, 1);
18            }
19        }
20    }
21 }
```

После добавления цикла для печати и упрощения уже написанного нами кода финальный код должен выглядеть так:



```
7     public static void main(String args[])
8     {
9         HashMap<String, Integer> good2count = new HashMap<>();
10
11        Scanner scanner = new Scanner(System.in);
12        for(;;)
13        {
14            String goodName = scanner.nextLine();
15
16            if(goodName.equals("LIST"))
17            {
18                printMap(good2count);
19                continue;
20            }
21
22            int count = 1;
23            if(good2count.containsKey(goodName)) {
24                count = good2count.get(goodName) + 1;
25            }
26            good2count.put(goodName, count);
27        }
28    }
29
30    private static void printMap(Map<String, Integer> map)
31    {
32        for(String key : map.keySet()) {
33            System.out.println(key + " => " + map.get(key));
34        }
35    }
36 }
```

В консоль после этого можно вводить список в консоль, и после команды list она выдаст список продуктов с указанным количеством:



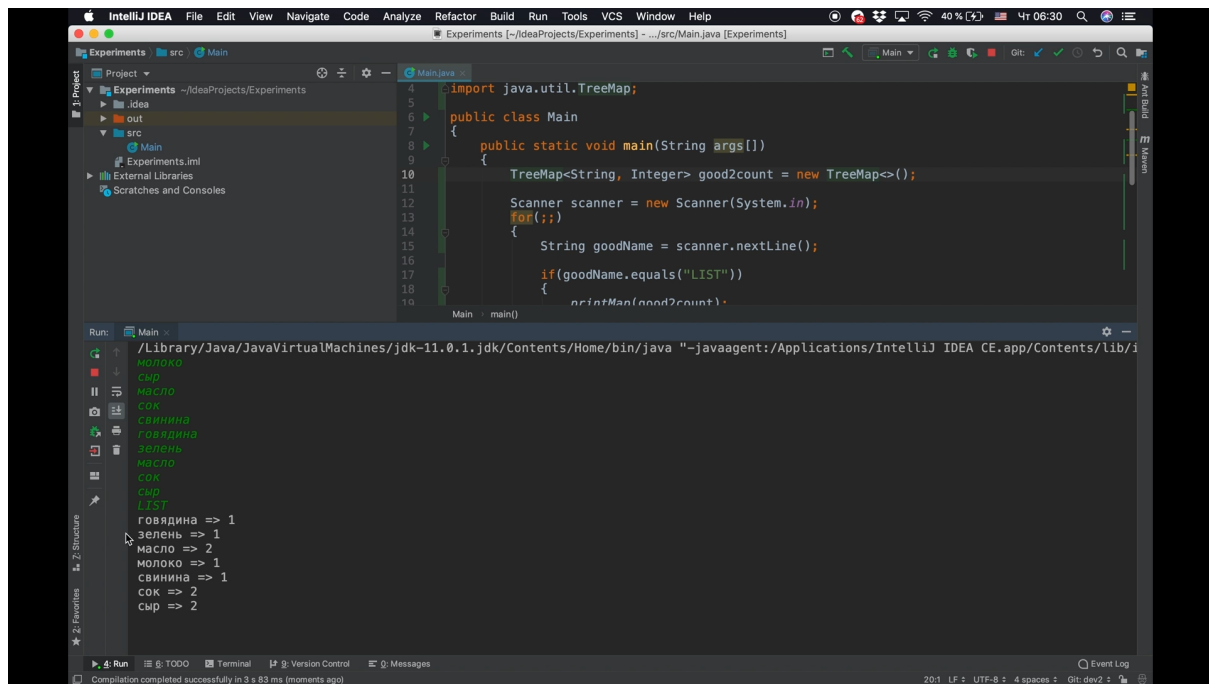
Ещё раз вспомним, какие методы есть у HashMap:

- put — в него вписывают ключ и значения;
- get — он получает ключи и выдаёт значения;
- remove — удаляет определённые ключи и соответствующие им значения.

С помощью цикла, который мы вводили выше, можно перебирать ключи map'a. С этих ключей можно получить вот так:

```
Set<String> keys = map.keySet();
```

Обратите внимание: когда мы получили список продуктов с количеством, ключи в нём не отсортированы. Чтобы это исправить, достаточно HashMap заменить на TreeMap во всех строках кода, где он встречается. Затем надо проимпортировать этот класс и запустить программу:



Теперь все товары вывелись в алфавитном порядке.