1. Создайте ассоциативный массив с ценами на вещи, которые вы хотели бы приобрести. Затем создайте второй ассоциативный массив с теми же ключами и ценами с 10%-ной скидкой.

**def** exercise1(score: Map[String, Double]): Unit ={  
 **var** newScore=(**for**((k,v) <-score) **yield** (k,v\*0.9))  
 *println*(newScore)  
}



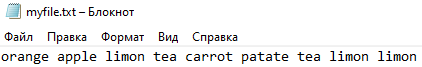
1. Напишите программу, читающую слова из файла. Используйте изменяемый ассоциативный массив для подсчета вхождений каждого слова. Для чтения слов используйте java.util.Scanner:

val in = new java.util.Scanner(new java.io.File("myfile.txt”))

while (in.hasNext()) process in.next()

Или загляните в главу 9, где представлено более идиоматичное для Scala решение. В конце выведите все слова и их счетчики.

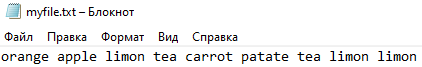
**def** exercise2(): Unit ={  
 **var** map = HashMap[String, Int]()  
  
 **val** in = **new** java.util.Scanner(**new** java.io.File("myfile.txt"))  
 **while** (in.hasNext()) {  
 **var** str = in.next()  
 **if** (map.contains(str))  
 map(str)+=1  
 **else** map(str)=1  
 }  
 *println*(map)  
}





1. Выполните предыдущее упражнение, используя неизменяемый ассоциативный массив.

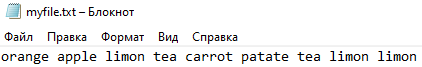
**def** exercise3(): Unit = {  
 **var** map = Map[String, Int]()  
  
 **val** in = **new** java.util.Scanner(**new** java.io.File("myfile.txt"))  
 **while** (in.hasNext()) {  
 **var** str = in.next()  
 **if** (map.contains(str))  
 map = map + (str -> (map(str) + 1))  
 **else** map = map + (str -> 1)  
 }  
 *println*(map)  
}





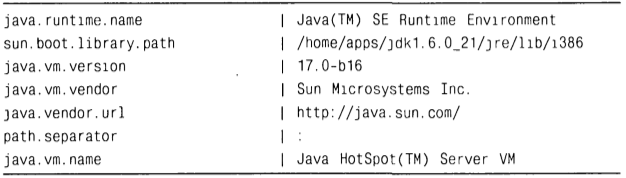
1. Выполните предыдущее упражнение, используя сортированный ассоциативный массив, чтобы слова выводились в отсортированном порядке.

**def** exercise4(): Unit = {  
 **var** map = SortedMap[String, Int]()  
  
 **val** in = **new** java.util.Scanner(**new** java.io.File("myfile.txt"))  
 **while** (in.hasNext()) {  
 **var** str = in.next()  
 **if** (map.contains(str))  
 map(str)+=1  
 **else** map(str)=1  
 }  
 *println*(map)  
}





1. Выполните предыдущее упражнение, используя java.util.TreeMap, адаптировав его для работы со Scala API.
2. Определите связанную хеш-таблицу, отображающую «Monday» в java.util.Calendar.MONDAY, и так далее для других дней недели. Продемонстрируйте обход элементов в порядке их добавления.
3. Выведите таблицу всех Java-свойств, таких как:



Для этого перед выводом таблицы нужно отыскать длину самого длинного ключа.

1. Напишите функцию minmax(values: Array[Int]), возвращающую пару, содержащую наименьшее и наибольшее значения.
2. Напишите функцию lteqgt(values: Array[Int], v: Int), возвращающую тройку, содержащую счетчик значений меньших v, равных v и больших v.
3. Что произойдет, если попытаться упаковать две строки, такие как “Hello”.zip(“World”)? Придумайте достаточно реалистичный случай использования.