

10.8

Наборы уникальных элементов

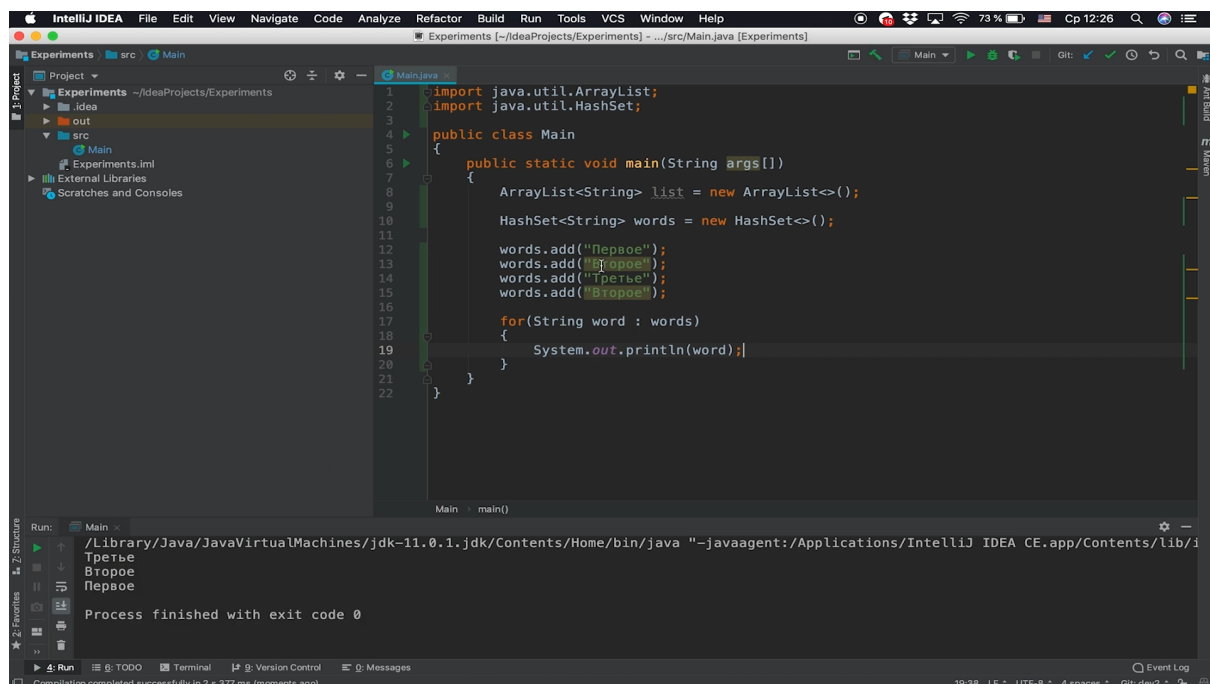
В Java существует множество различных коллекций, но мы изучим лишь некоторые.

Мы уже изучили коллекцию List. Она характерна тем, что порядок элементов строго определён и у каждого элемента есть индекс. Элементы можно добавлять, заменять, удалять. Когда вы добавляете ещё один элемент, он добавляется как следующий.

Есть и другие коллекции, и одна из них — коллекция типа Set. Коллекции List и Set представляют собой интерфейсы, о которых мы поговорим в одном из следующих модулей.

Создадим коллекцию класса HashSet. Здесь, так же как и в List, в угловых скобках указывается тип элементов, которые в них хранятся. Пусть у нас будут слова, например содержание орфографического словаря. Слова здесь можно добавлять методом add, но обратите внимание, что этот метод здесь один, без индекса.

Выводим написанный код в консоль с помощью цикла for и смотрим на результат:



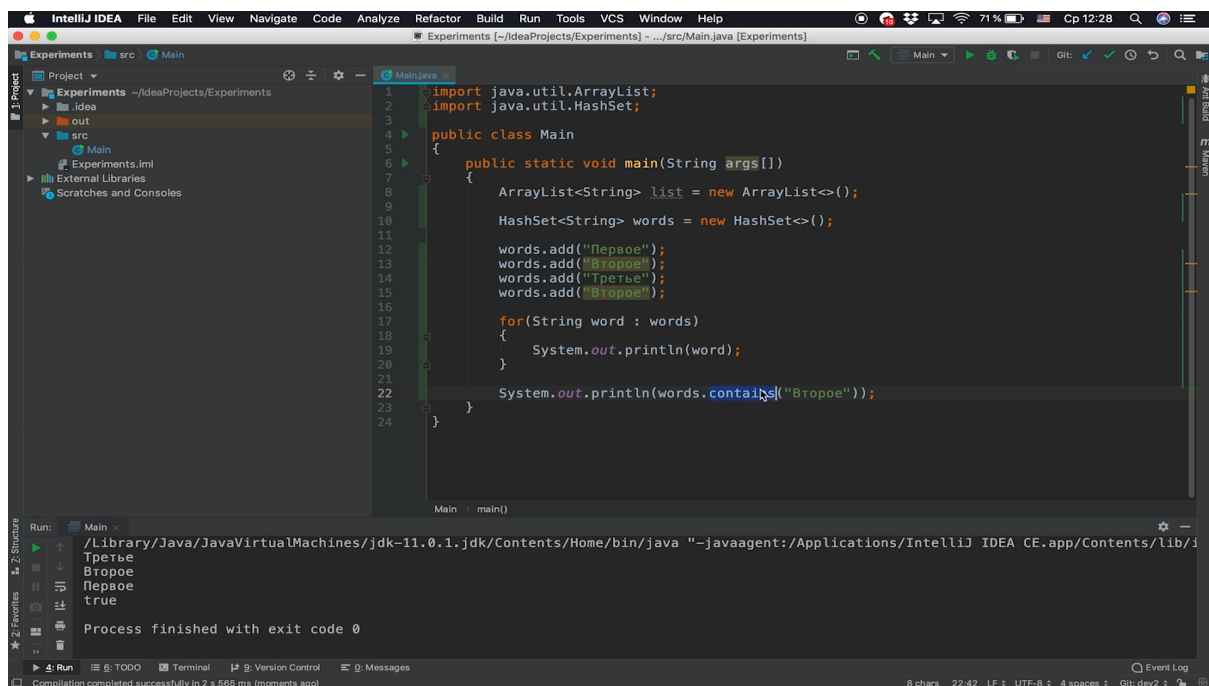
```
1 import java.util.ArrayList;
2 import java.util.HashSet;
3
4 public class Main
5 {
6     public static void main(String args[])
7     {
8         ArrayList<String> list = new ArrayList<>();
9
10        HashSet<String> words = new HashSet<>();
11
12        words.add("Первое");
13        words.add("Второе");
14        words.add("Третье");
15        words.add("Четвертое");
16
17        for(String word : words)
18        {
19            System.out.println(word);
20        }
21    }
22 }
```

Run: Main
/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/i
Третье
Второе
Первое
Process finished with exit code 0

Слова вывелись в таком порядке, в котором они хранятся в коллекции. Но, как вы можете заметить, порядок достаточно странный: он не алфавитный, и не тот, в котором мы добавляли элементы. При этом слово «Второе», которое мы записали дважды, напечаталось только один раз.

Почему так происходит? Потому, что каждая строка в эту коллекцию добавлена через так называемый hash. Из-за этого каждая строка преобразована в уникальную строку, по которой этот объект очень легко найти на программном уровне.

В списке `ArrayList`, чтобы найти какой-то объект, необходимо перебрать весь список и сравнить с каждым элементом тот элемент, который вы ищете. В случае `HashSet` для поиска вы можете воспользоваться методом `words.contains`, и этот метод сразу выдаст вам результат в формате `true/false`.



The screenshot shows the IntelliJ IDEA IDE with a project named 'Experiments'. The main file, 'Main.java', contains the following code:

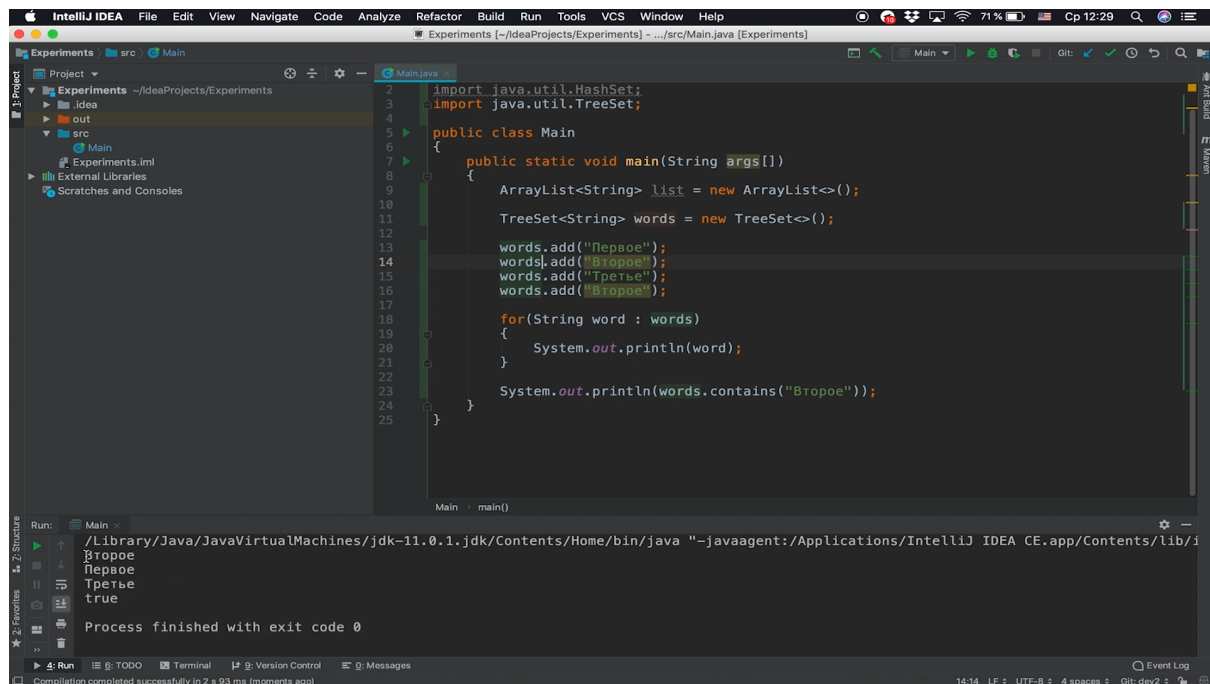
```
1 import java.util.ArrayList;
2 import java.util.HashSet;
3
4 public class Main
5 {
6     public static void main(String args[])
7     {
8         ArrayList<String> list = new ArrayList<>();
9
10        HashSet<String> words = new HashSet<>();
11
12        words.add("Первое");
13        words.add("Второе");
14        words.add("Третье");
15        words.add("Второе");
16
17        for(String word : words)
18        {
19            System.out.println(word);
20        }
21
22        System.out.println(words.contains("Второе"));
23    }
24 }
```

The output window at the bottom shows the results of the program execution:

```
Run: Main
/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/i
Третье
Второе
Первое
true
Process finished with exit code 0
```

The status bar at the bottom indicates 'Compilation completed successfully in 2 s 565 ms (moments ago)'.

Если же вам нужен каким-либо образом упорядоченный список, то вы можете воспользоваться коллекцией TreeSet.



```
1 import java.util.HashSet;
2 import java.util.TreeSet;
3
4 public class Main
5 {
6     public static void main(String args[])
7     {
8         ArrayList<String> list = new ArrayList<>();
9         TreeSet<String> words = new TreeSet<>();
10
11         words.add("Первое");
12         words.add("Второе");
13         words.add("Третье");
14         words.add("Второе");
15
16         for(String word : words)
17         {
18             System.out.println(word);
19         }
20
21         System.out.println(words.contains("Второе"));
22     }
23 }
24
25
```

Run: Main

/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/i

Второе
Первое
Третье
true

Process finished with exit code 0

Compilation completed successfully in 2 s 93 ms (moments ago)

Здесь список упорядочен по алфавиту. Элементы в данном случае сравниваются и располагаются таким образом, чтобы их было легко искать. Список имеет на выходе древесную структуру, откуда и идёт название TreeSet.

Таким образом, HashSet — неупорядоченный набор элементов, а TreeSet — упорядоченный. В одном из следующих модулей мы поговорим о том, как задавать определённый порядок для TreeSet, когда будем изучать интерфейсы.