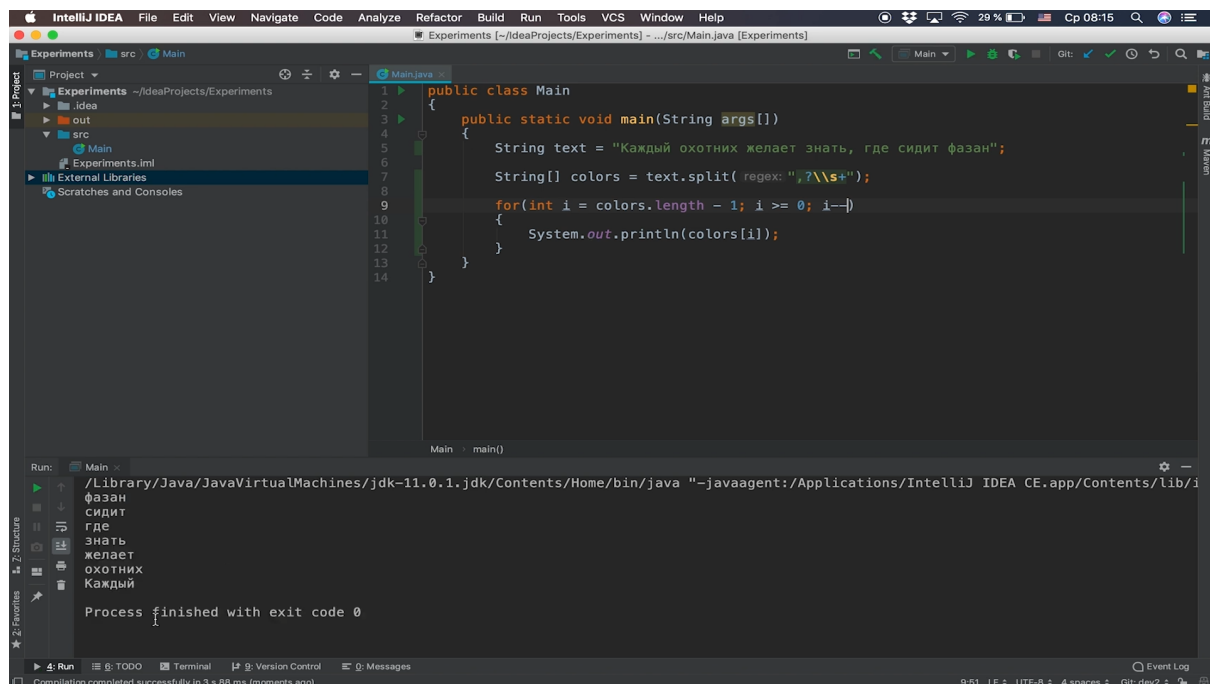


10.3

Обход массивов в обратном порядке

Массивы можно перебирать и в обратном порядке. То есть если мы хотим вывести в консоль фразу «каждый охотник желает знать, где сидит фазан» в обратном порядке, нужно начинать с последнего элемента массива. В данном случае элемент будет иметь вот такой номер: `colors.length - 1`.

То есть мы берём длину массива (здесь она будет 7) и, поскольку нумерация начинается с нуля, идём с шестого элемента по нулевой. Также в этой строке мы пишем не `i++`, а `i--` и выводим эти элементы в обратном порядке в консоль:



```
1 public class Main
2 {
3     public static void main(String args[])
4     {
5         String text = "Каждый охотник желает знать, где сидит фазан";
6         String[] colors = text.split(" ");
7         for(int i = colors.length - 1; i >= 0; i--)
8         {
9             System.out.println(colors[i]);
10        }
11    }
12 }
13
14
```

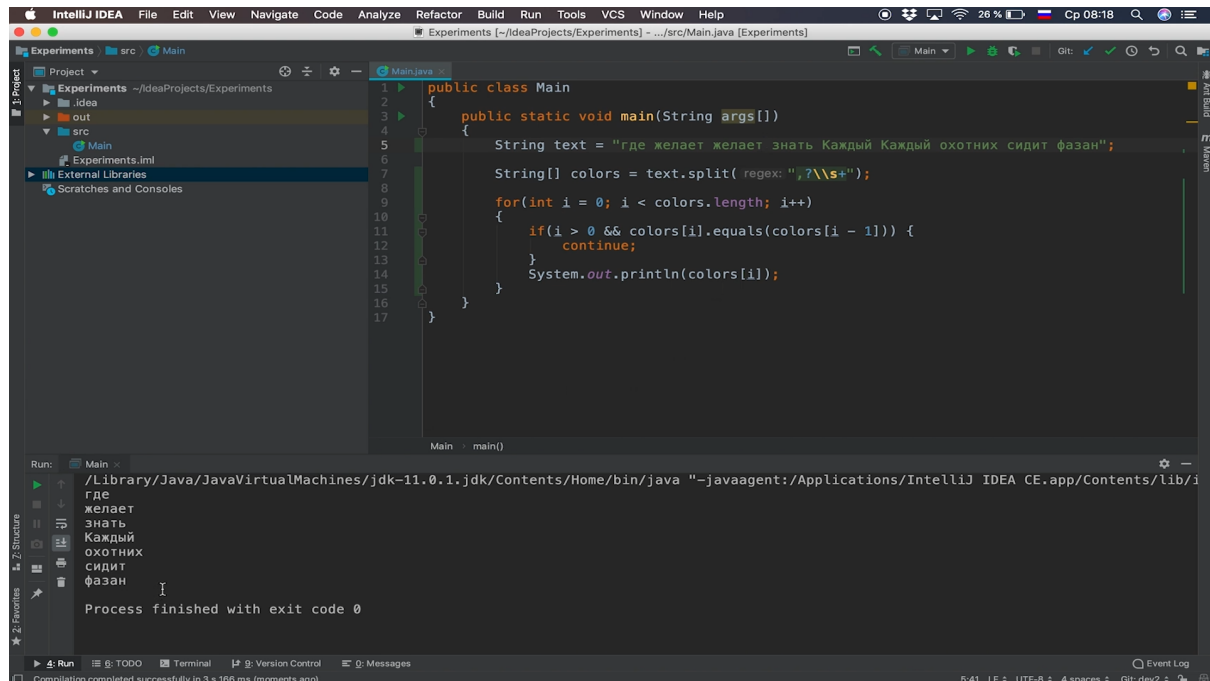
Run: Main -> /Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/i...

фазан
сидит
где
знает
желает
охотник
Каждый

Process finished with exit code 0

Использование индексов массивов и номеров элементов бывает очень полезно. Есть задачи, где необходимо сравнить элемент со следующим после него элементом. Например, у нас есть текст с несколькими повторяющимися одинаковыми словами, отсортируем их в алфавитном порядке. Нам необходимо вывести в консоль каждое слово только один раз, соответственно, нужно сравнивать каждое слово со следующим после него словом. Добавляем проверку: если `i` больше нуля (добавляем этот пункт в проверку,

чтобы не возникла ошибка — не произошло сравнение нулевого элемента с минус первым, поскольку его не существует) и текущий элемент массива равен предыдущему элементу, то цикл продолжается (continue). Выводим код в консоль:



```
1 public class Main
2 {
3     public static void main(String args[])
4     {
5         String text = "где желает желает знать Каждый Каждый охотник сидит фазан";
6         String[] colors = text.split( regex: ",?\\s+" );
7         for(int i = 0; i < colors.length; i++)
8         {
9             if(i > 0 && colors[i].equals(colors[i - 1])) {
10                 continue;
11             }
12             System.out.println(colors[i]);
13         }
14     }
15 }
16
17 }
```

Run: Main
/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/i
где
желает
знать
Каждый
ОХОТНИК
СИДИТ
фазан
Process finished with exit code 0

Как видите, каждый элемент массива выведен только один раз.

Это простой пример использования обратного перебора массива, существуют и более сложные варианты, которых следует избегать. Частая ошибка в написании поддерживаемого кода — чрезмерное использование вложенных друг в друга конструкций. Созданный нами код уже достаточно тяжёлый для восприятия, но если вы добавите в него дополнительные циклы, такой код будет невозможно читать или поддерживать, особенно если эти дополнительные элементы будут называться одной буквой. Этого надо избегать, однобуквенных индексов должно быть не больше двух в одном коде.

Если вы всё-таки хотите использовать сложные вложенные циклы, лучше выносить их в отдельные методы и раскрывать.