3\_6 сравнение строк по содержанию

**1. Сравнение строк**

Это все отлично. Но вы же видите, что строки s1 и s2 фактически одинаковые – они содержат одинаковый текст. Как сказать программе, чтобы при сравнении строк она смотрела не на адреса объектов String, а на их содержимое?

Для этого в Java у типа String есть специальный метод — equals. Выглядит его вызов так:

строка1.equals(строка2)

Сравнение двух строк

Этот метод возвращает true (истина), если строки одинаковые, и false (ложь), если они не одинаковые.

Пример:

|  |  |
| --- | --- |
| **Код:** | **Примечание** |
| String s1 = "Привет";  String s2 = "ПРИВЕТ";  String s3 = s1.toUpperCase();  System.out.println(s1.equals(s2));  System.out.println(s1.equals(s3));  System.out.println(s2.equals(s3)); | // Привет  // ПРИВЕТ  // ПРИВЕТ  false // разные  false // разные  true // одинаковые, хотя адреса разные |

Еще примеры:

|  |  |
| --- | --- |
| **Код** | **Пояснение** |
| "Привет".equals("ПРИВЕТ") | false |
| String s = "Привет";  "Привет".equals(s); | true |
| String s = "При";  "Привет".equals(s + "вет"); | true |
| String s = "П";  (s + "ривет").equals(s + "ривет"); | true |

**2. Сравнение строк без учета регистра**

В последнем примере вы увидели, что сравнение "Привет".equals("ПРИВЕТ") дает false. Действительно, строки-то не равны. Хотя...

Строки, конечно, не равны, но ведь в них написано одно и то же, только размер (регистр) букв отличается. Можно ли как-нибудь их так сравнить, чтобы размер букв при этом не учитывался? Ну чтобы "Привет".equals("ПРИВЕТ") дало true?

И ответ на этот вопрос — да. В Java у типа String есть еще один специальный метод — equalsIgnoreCase. Выглядит его вызов так:

строка1.equalsIgnoreCase(строка2)

Название метода переводится как сравнить, игнорируя регистр. В названии метода есть две вертикальные линии: первая — это L маленькая, а вторая — это i, только большая. Не перепутайте.

Пример:

|  |  |
| --- | --- |
| **Код:** | **Примечание** |
|  |  |
| String s1 = "Привет";  String s2 = "ПРИВЕТ";  String s3 = s1.toUpperCase();  System.out.println(s1.equalsIgnoreCase(s2));  System.out.println(s1.equalsIgnoreCase(s3));  System.out.println(s2.equalsIgnoreCase(s3)); | | | // Привет  // ПРИВЕТ  // ПРИВЕТ  true  true  true |

**3. Пример сравнения строк**

Приведем только один простой пример: нужно ввести с клавиатуры две строки и определить, одинаковые они, или нет. Вот как будет выглядеть этот код:

Scanner console = new Scanner(System.in);

String a = console.nextLine();

String b = console.nextLine();

String result = a.equals(b) ? "Одинаковые" : "Разные";

System.out.println(result);

**4. Интересный нюанс сравнения строк**

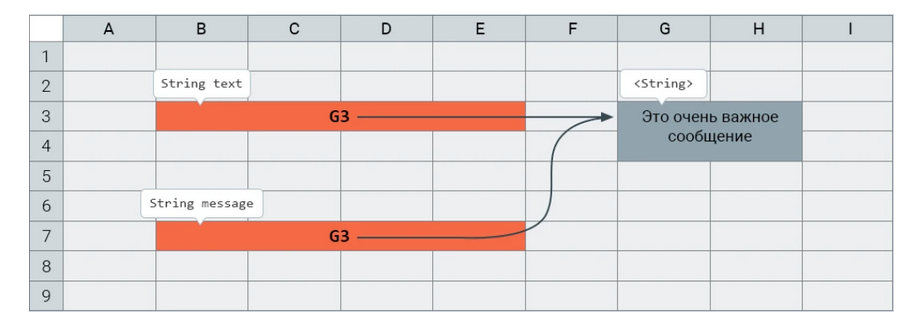
Есть один важный нюанс, о котором вам нужно знать.

Если Java-компилятор в вашем коде (именно в коде) найдет несколько одинаковых строк, для экономии памяти он создаст для них только один объект.

String text = "Это очень важное сообщение";

String message = "Это очень важное сообщение";

А вот что в результате будет в памяти:



И если в данном случае вы сравните между собой text == message, получите true. Так что не удивляйтесь.

А если вам вдруг очень нужно, чтобы ссылки были разные, вы можете написать так:

String text = "Это очень важное сообщение";

String message = new String ("Это очень важное сообщение");

Или так:

String text = "Это очень важное сообщение";

String message = new String (text);

В обоих этих случаях переменные text и message указывают на разные объекты, содержащие одинаковый текст.