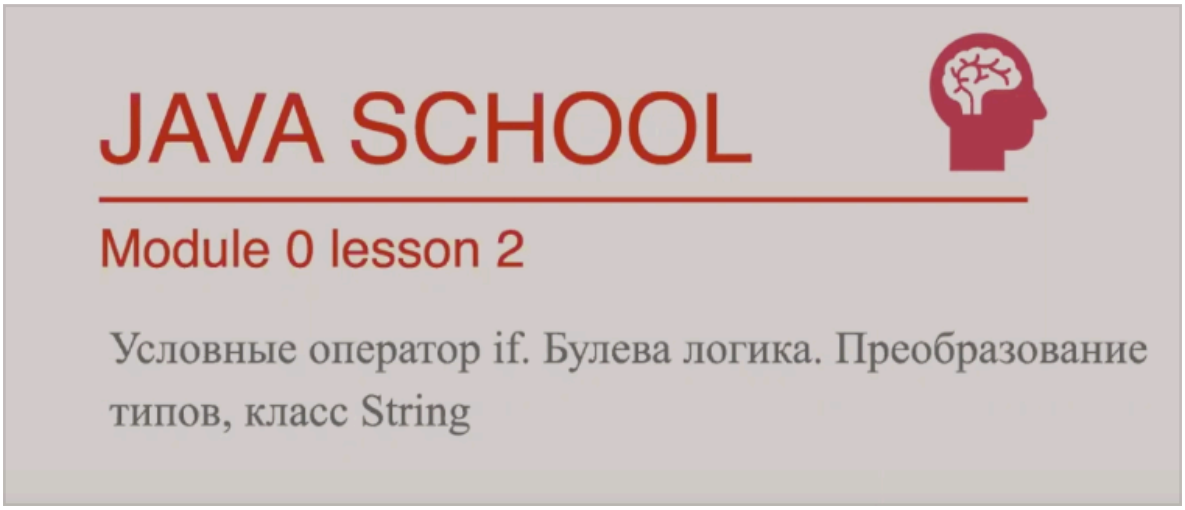
Java - Условный оператор if. Булава логика. Преобразование типов, класс String



Условные операторы

В **java**, как и в других языках программирования используется условный оператор **if**

Оператор **if** позволяет программе в зависимости от условия выполнить тот или иной код, который заключен в оператор

Форма оператора **if** выглядит так:

```
if (/*условие*/) {
    // код при выполнении условия
}
```

Если условие внутри скобок оператора не выполняется, то при такой конструкции выполнение программы продолжится, а код внутри оператора не выполнится

Условные операторы

Однако, можно управлять программой в случае если условие не выполнилось

Для этого форма оператора **if** немного изменится

```
if (/*условие*/) {
    // код при выполнении условия
} else {
    // код при не выполнении условия
}
```

Оператор ***else*** необязательный, поэтому его стоит добавлять когда это действительно нужно

Булевы операции

Рассмотрим какими бывают условия и как их задавать

Оператор **if** принимает переменные или операции результатом которы будет являться **boolean**

Рассмотрим тип **boolean**

boolean это примитивный тип данных. Имеет два значения **true** или **false**

Если рассматривать **boolean** в операторе **if**, то при условии равном **true** выполнится блок кода внутри оператора

```
if (true) {
    // код при выполнении условия
} else {
    // код при не выполнении условия
}
```

Булевы операции

Таблица основных булевых операций

Оператор	Описание
&&	Сокращенное логическое И
&	Логическое И
||	Сокращеноне логическое ИЛИ
|	Логическое ИЛИ
!	Отрицание
==	Равенство
!=	Не равенство

Булевы операции

Таблица примеров некоторых операций

A	B	A || B	A && B	A == B	A!=B	!A	!B
true	false	true	false	false	true	false	true
false	false	false	false	true	false	true	true
true	true	true	true	true	false	false	false
false	true	true	false	false	true	true	false

Булевы операции

Примеры операций с числами

Предположим у нас есть три переменные:

int a = 10; int b = 17; и int c = 10;

Операция	Результат
a > b	false
a == c	true
a + 7 == b	true
b > c	true
a != c	false
c < a	false
a >= c	true

Булевы операции

Попробуем использовать логические операции вместе с оператором **if**

Запускаем IntelliJ IDEA, далее нажимаем create new project, выбираем java (вверху), жмем next

Project name: booleanlogic

Project location: C:\projects\booleanlogic

И наконец finish

Далее в папе src создаем пакеты с именем com.javastart.booleanlogic и класс Main

С методом **public static void main(String[] args)**

Ввод с консоли

Теперь мы познакомимся с вводом с консоли, как он работает мы разберем чуть позже, пока будем просто пользоваться

Для этого объявим переменную типа **Scanner** и назовем ее **inputFromLine**

```
public class Main {

    public static void main(String[] args) {
        Scanner inputFromLine = new Scanner(System.in);
    }
}
```

Ввод с консоли

Попробуем считать числа с консоли, сложить их и вывести на экран

Для этого нужно объявить три переменные типа **int**, *firstNumber*, *secondNumber* и *result*

Присвоить число введенное с консоли можно с помощью конструкции **inputFromLine.nextInt()**

```
public static void main(String[] args) {
    Scanner inputFromLine = new Scanner(System.in);
    int firstNumber;
    int secondNumber;
    int result;

    System.out.println("Введите первое число");
    firstNumber = inputFromLine.nextInt();

    System.out.println("Введите второе число");
    secondNumber = inputFromLine.nextInt();

    result = firstNumber + secondNumber;
    System.out.println("Сумма: " + result);
```

Оператор if

Вернемся к оператору **if**

Давайте выведем наибольшее из введенных чисел

Для этого удалим строки со сложением, выводом суммы и с объявлением переменной ***result***

Добавим оператор **if**

```
if (firstNumber > secondNumber) {
    System.out.println("Первое число больше");
}
```

Оператор if

Добавим обработку ситуации, в которой первое число меньше, для это нужно дописать оператор **else** и вывести в консоль, что второе число больше

```
if (firstNumber > secondNumber) {
    System.out.println("Первое число больше");
} else {
    System.out.println("Второе число больше");
}
```

Оператор if

Однако если числа равны между собой, то мы всегда будем идти во ветки **else**

Можно предотвратить эту ситуацию разными способами

Можно сделать три проверки, то есть три отдельные блока **if**, Так же можно использовать немного другую конструкцию с **if**

```
if (firstNumber > secondNumber) {
    System.out.println("Первое число больше");
} else if (firstNumber < secondNumber){
    System.out.println("Второе число больше");
} else if (firstNumber == secondNumber) {
    System.out.println("Числа равны");
}
```

Давайте попробуем эти варианты

Класс String

Вернемся к примитивным типам данных

В java из примитивных типов связанных со строкой есть только **char**, который может хранить только один символ

```
char c = 'a';
```

Для хранения строки существует класс **String**

Объявление строки выглядит следующим образом:

```
String line = "Hello world!";
```

Тип данных **String**, это не примитивный тип, а полноценный ссылочный тип, с ссылочными типами мы познакомимся на следующих занятиях

Класс String

Давайте преобразуем все строки которые мы писали и объявим их как тип **String**

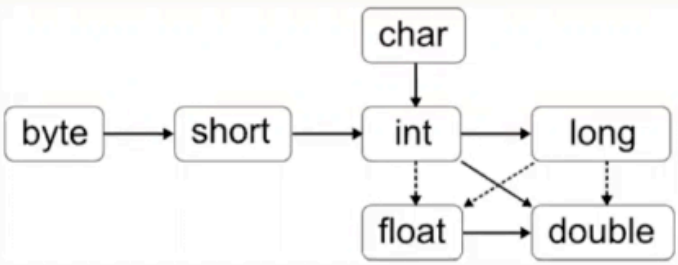
```
String enterFirstNumber = "Введите первое число";
String enterSecondNumber = "Введите второе число";
String firstIsBigger = "Первое число больше";
String secondIsBigger = "Второе число больше";
String numbersAreEquals = "Числа равны";
```

Преобразование типов

Немного о преобразовании типов

Вообще, **java** - это строготипизированный язык, где у каждой переменной должен быть тип

Но возможно и преобразование типов



Это схема преобразования примитивных типов. Сплошные линии обозначают преобразования, выполняемые без потери данных. Штриховые линии говорят о том, что при преобразовании может произойти потеря точности

Преобразование типов

Давайте попробуем преобразовать типы

Преобразование типов выглядит следующим образом:

```
int a = 10;
byte b;
b = (byte) a;
```

Для того что бы привести один тип данных к другому, перед присвоением в скобочках пишется тип переменной в которую будут записываться значение

Итог

Сегодня мы познакомились с условным оператором **if** научились его использоваться и посмотрели какие конструкции можно создавать с его помощью

Узнали что такое тип **boolean** и какие выражения могут иметь результат такого типа

Познакомились с типом **String** и узнали где можно хранить строки, в следующих занятиях мы часто будем говорить про класс **String** и его возможности

Посмотрели что такое преобразование типов, к этому мы еще тоже будем возвращаться
