Условные операторы if и else

Привет!

Вы уже видели условные операторы if и else, а в этом уроке мы разберём их ещё раз и чуть более детально.

Напомню, что операторы в языке программирования — это некие команды, которые выполняют операции.

Условные операторы — это операторы, которые проверяют выполнение условий и выполняют или не выполняют какой-то код.

Посмотрим на механизм работы основных условных операторов if и else.

Оператор If

00:40–01:34

Начнём с самого простого оператора if («если»). После этого оператора пишутся сначала круглые скобки, а затем фигурные.

В круглых скобках пишется условие, которое этот оператор должен проверить. В качестве условия может быть как просто переменная, так и выражение.

В фигурных скобках пишется код, который должен выполниться, если выполнилось условие. Фигурные скобки в Java обозначают различные блоки кода, в том числе блоки кода после условных операторов.

Если результат выполнения выражения, которое стоит в круглых скобках, либо значение указанной в них переменной равно true, то условие будет считаться выполненным и код в фигурных скобках выполнится.

Если же это значение будет равно false, код в фигурных скобках не выполнится.

```
if (condition) {

}
```

Оператор else

01:34–02:36

Второй условный оператор, else («иначе»), позволяет выполнить код в случае, если условие после оператора if не выполнилось..

Такой код можно прочитать как: «если condition, далее — некий код, иначе — другой код». Обратите внимание, что оператор else может быть в коде только после оператора if, в качестве его дополнения.

```
if (condition) {

} else {

}
```

Если первое условие не выполнилось и вы хотите проверить выполнение другого условия, то его можно дописать сразу после оператора else.

Получится цепочка операторов: если condition равен true, то выполнится один код, иначе и если condition2 равен true, некий второй код, а иначе — некий третий код.

```
if (condition) {

} else if (condition2) {

} else {

}
```

Цепочки условий могут быть и длиннее, в зависимости от решаемой задачи, но пишутся они по одним и тем же принципам.

```
if (condition) {

} else if (condition2) {

} else if (condition3) {

```

```
} else {

}
```

Частые ошибки

02:36–05:03

Теперь разберём частые ошибки, с которыми сталкиваются начинающие разработчики при использовании операторов if и else.

В Java все основные строки кода должны оканчиваться на точку с запятой. Точка с запятой означает конец выполняемой команды, и она никогда не ставится внутри конструкций.

Иногда её по ошибке ставят после оператора if и круглых скобок, и это приводит к тому, что выполнение оператора if на этом заканчивается. Код в фигурных скобках выполнится в любом случае, независимо от оператора if:

```
if (condition); {

}
```

Такая ошибка не подсвечивается средой разработки, поскольку не является ошибкой синтаксиса. Её можно не заметить с непривычки, особенно если открывающую фигурную скобку переносить на следующую строку. Обратите внимание, открывающую фигурную скобку в Java после операторов принято писать на той же строке, на которой находится соответствующий оператор.

```
if (condition);
{

}
```

Вторая ошибка, с которой можно столкнуться, — это написание кода после оператора if или оператора else без фигурных скобок. Если писать код без фигурных скобок, то оператор будет влиять только на первую строку, и когда строка одна, никаких ошибок не будет. Но если вы допишете ещё одну строку и забудете поставить фигурные скобки, то эта новая строка будет выполняться всегда, независимо от оператора:

```
if (condition)
System.out.println("YES");
           value = value + 1;
```

Для простоты, лучшей читабельности кода и во избежание ошибок рекомендуем фигурные скобки после операторов писать всегда:

```
if (condition) {
System.out.println("YES");
value = value + 1;
}
```

Посмотрим на условные операторы в коде проекта FillingStation. Здесь используется и оператор if, и оператор else, и их комбинация else if.

Если тип топлива у нас 92, то переменная с ценой устанавливается равной цене 92-го бензина. В противном случае проверяется второе условие, и если тип равен 95, то цена приравнивается к стоимости 95-го бензина.

```
if (fuelType == 92) {
fuelPrice = fuel92price;
}

else if (fuelType == 95) {
fuelPrice = fuel95price;
}

else {
System.out.println(wrongFuelTypeMessage);
}

if (hasDiscount) {
fuelPrice = (1 - discount) * fuelPrice;
}

if (amount < 1) {
System.out.println("Указано слишком малое количество топлива");
amount = 0;
}
```

Если ни одно из двух условий не сработало, то выполняется блок кода после последнего оператора else: в консоль выводится сообщение о том, что указан неверный тип топлива.

Итоги

с 05:03 до конца

- Вы познакомились с условными операторами if и else, с их комбинацией else if, с тем, как их можно записывать в коде, какие типичные ошибки при этом возможны и как их избегать.
- Выражения или переменные в круглых скобках, выполнение которых проверяются при помощи оператора if, должны быть равны true или false — одному из двух вариантов переменной типа boolean.
- На практике часто бывает необходимо проверить сразу несколько условий или их комбинацию. О том, как комбинировать условия, вы узнаете в следующей теме.