

Модуль 1. Основы программирования

Тема 1.4. Логические выражения и тип boolean

1 час

Оглавление

1.4. Логические выражения и тип boolean	2
1.4.1. Логические выражения.....	2
Операторы сравнения.....	2
Логические операторы.....	2
1.4.2. Тип boolean	3
1.4.3. Тернарная операция	4
Упражнение 1.4.1.	5
Упражнение 1.4.2.	5
Задание 1.4.1.....	6
Задание 1.4.2.....	6
Благодарности	6

1.4. Логические выражения и тип boolean

1.4.1. Логические выражения

С логическими операторами программисты работают значительно чаще, чем с поразрядными. Это связано с тем, что в Java из них строятся логические выражения, которые повсеместно используются в операторах ветвления и цикла, которые будут рассмотрены в следующих темах.

Операторы сравнения

> строго больше	<= меньше или равно
< строго меньше	== равно
>= больше или равно	!= не равно

Логические операторы

&& И (конъюнкция или логическое умножение)	Результат ИСТИНА, если оба операнда ИСТИНА, в остальных случаях - ЛОЖЬ
 ИЛИ (дизъюнкция или логическое сложение)	Результат ЛОЖЬ, если оба операнда ЛОЖЬ, в остальных случаях - ИСТИНА
! НЕ (отрицание)	Унарная операция. Если операнд ИСТИНА, то результат - ЛОЖЬ, и наоборот

Чтобы правильно использовать несколько логических операторов в одном выражении, нужно учитывать приоритет операций, который и определяет порядок вычисления результата логического выражения. Среди логических операций наибольший приоритет у операции **!**, дальше **&&** и наименьший приоритет имеет операция **||**.

В выражениях, где встречаются операторы разных типов, сначала выполняются арифметические операции, затем операции сравнения, затем логические операции и в последнюю очередь присваивание.

В Java нельзя выстраивать цепочки операций сравнения - это строго бинарные операции. Это значит, что двойные условия типа $1 < x < 6$ записываются в виде пар логических операций. Для этого нужно разделить такое сложное условие двумя простыми и связать их операцией И:

`1 < x && x < 6`

С учетом приоритетов такое выражение будет вычисляться следующим образом:

1. $1 < x$
2. $x < 6$
3. Операция `&&` между результатами, полученными на шаге 1 и 2.

Пример 1

Выражение `x < 5 || x > 5 && x < 7` является истинным, для всех значений `x`, меньших 5 и (строго говоря «или») числа 6, а не для всех значений, меньших 7.

1.4.2. Тип boolean

Для хранения логических величин в Java имеется специальный тип **boolean**. Размер типа **boolean** зависит от виртуальной машины.



Из официальной документации: "This data type represents one bit of information, but its "size" isn't something that's precisely defined."

<http://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

Т.е. размер типа **boolean** зависит только от реализации JVM. Чаще всего размер совпадает с машинным словом (32 или 64 бита). Но если завести массив из **boolean**, то каждый элемент будет занимать 1 byte.

<http://stackoverflow.com/questions/1907318/java-boolean-primitive-type-size/>

Вообще говоря можно было бы реализовать тип **boolean** всего при помощи одного бита, но это сделано для производительности.

Дело в том, что "байт" это минимальная единица адресации в компьютерной памяти. Для того, чтобы работать с отдельными битами, используют поразрядные операции, которые применяются ко всем битам байта. Окончательные сравнения происходят, естественно, тоже в целых байтах.

Переменные типа **boolean** могут принимать всего два специальных значения **true** и **false** (ИСТИНА и ЛОЖЬ). Значения этого типа можно сравнивать на равенство и производить с ними логические операции, но с ними нельзя производить арифметические вычисления.

Например, корректны все строки фрагмента:

```
int x = 5;
boolean b1 = x > 0
boolean b2 = b1 && (x < 20)
```

Переменная `b2` принимает значение `true`, потому что первый и второй операнды конъюнкции истинны.

1.4.3. Тернарная операция

Очень интересной является тернарная операция (операция с тремя операндами) **?:**

Синтаксис тернарной операции следующий:

```
//ПСЕВДОКОД
<условие> ? <значение, если условие истинно> :
<значение, если ложно>
```

Например, удобно записать поиск максимума из двух чисел так:

```
max = (a > b ? a : b);
```

Причем значения могут быть разных типов. Например, если нужно определить площадь квадрата по длине стороны или сообщить о некорректности данных, то можно уложиться в одну строку:

```
out.println(a > 0 ? a * a : "WRONG");
```

Для отработки навыков работы с логическими операциями очень хорошо подходят задачи informatics №№112165-112173.

По формату все задачи одинаковы. В каждой закрашена область на декартовой плоскости. В программу вводятся координаты точки, и она должна определить, принадлежит ли точка закрашенной области или нет.

Поэтому и все программы-решения могут выглядеть совершенно одинаково:

```
import java.io.PrintStream;
import java.util.Scanner;

public class MyProgram {
    public static Scanner in = new Scanner(System.in);
    public static PrintStream out = System.out;

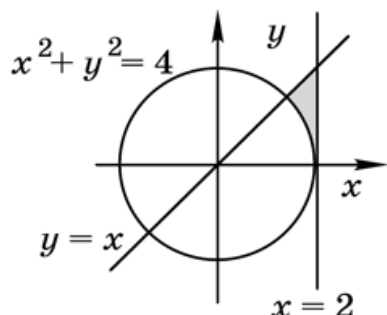
    public static void main(String[] args)
    {
        in.useLocale(Locale.US);

        double x = in.nextDouble();
        double y = in.nextDouble();
        out.println ((...) ? "YES" : "NO");
    }
}
```

и различаться только выделенным условием в скобках.

Упражнение 1.4.1.

Решим задачу **112165. Точка-1.**



Для вертикальной прямой условие очевидно: $x < 2$

Если дан график функции $y = f(x)$, то условие

«точка лежит **ниже** графика» записывается $y < f(x)$,

«точка лежит **выше** графика»: $y > f(x)$

В нашем случае, график – прямая с уравнением $y = x$, соответственно требуемое условие можно записать как $y < x$.

Наконец, принадлежность точки кругу («**внутри окружности**») записывается $x^2 + y^2 < R^2$,

«**вне окружности**» - $x^2 + y^2 > R^2$,

в нашем случае $x * x + y * y > 4$.

Точка принадлежит закрашенной области на рисунке, если она лежит левее вертикальной прямой **И** ниже графика функции $y=x$ **И** вне окружности.

К этим условиям надо добавить условие $y > 0$, иначе условию будут удовлетворять и точки под окружностью.

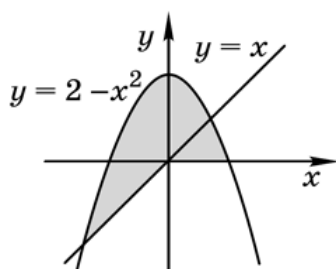
Соответственно, полностью условие для этой задачи будет выглядеть так:

$x < 2 \ \&\& \ y < x \ \&\& \ x * x + y * y > 4 \ \&\& \ y > 0$

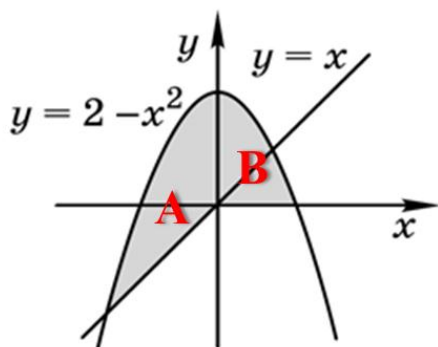
Для составления правильных условий, обычно надо разбивать сложную область на простые и связывать их логическими операциями.

Упражнение 1.4.2.

Задача 112167. Точка-3.



Удобно разбить область на две: левую (А) и правую (В):



Область А: точка лежит **под** параболой, **над** диагональной прямой и **слева** от оси ординат,

Таким образом, условие для нее будет $y < 2 - x * x \ \&\& \ y > x \ \&\& \ x < 0$.

Область В: точка лежит **под** параболой, **над** осью абсцисс и **справа** от оси ординат,

Таким образом, условие для нее будет $y < 2 - x * x \ \&\& \ y > 0 \ \&\& \ x \geq 0$.

Обратите внимание на последний знак \geq . Если написать просто знак $>$ программа будет считать точки, которая лежат на оси абсцисс, лежащими вне области.

Условие - ответ на задачу: (точка **лежит в области А**) **ИЛИ** (точка **лежит в области В**)

В принципе, скобки благодаря более высокому приоритету операции $\&\&$ ставить необязательно.

Задание 1.4.1

Для того, чтобы «почувствовать» битовые операции, нужно решить несколько задач. Попробуйте решить задачи <http://informatics.msk.ru> №№121, 123, 124, 125, 127

Задание 1.4.2

1. Дорешайте задачи, предложенные на занятии.

2*. Решите задачу 3060 (определить, является ли число точной степенью двойки) **без использования циклов**.

Благодарности

Компания Samsung Electronics выражает благодарность за участие в подготовке данного материала преподавателю ИТ ШКОЛЫ SAMSUNG Ильину Владимиру Владимировичу.