**Тип boolean**

[Java Syntax](https://javarush.ru/quests/QUEST_JAVA_SYNTAX)

[Уровень 4](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX&level=4), Лекция 7

— Привет, Амиго. Хочу рассказать тебе о новом типе данных. Это тип boolean. Переменные этого типа могут принимать всего два значения – **true** (истина/правда) и **false** (ложь).

— А как его использовать?

— Этот тип неявно используется во многих местах. Так же, как и результат любого сложения – число, то и результат любого сравнения – истина или ложь – тип **boolean**. Примеры:

|  |  |  |
| --- | --- | --- |
|  | **Код** | **Пояснение** |
| 1 | boolean m; | Два данных выражения эквивалентны. Значение переменой типа boolean по умолчанию **false** («ложь»). |
| 2 | boolean m = false; |
| 3 | if (a > b)  System.out.println(a); | В переменную m будет записан результат сравнения – **true** или **false**. Условие выполняется, если переданное в него выражение истинно – **true**. |
| 4 | **boolean** m = (a > b);  if (m)  System.out.println(a); |
| 5 | **boolean** m = (a > b);  if (m == true)  System.out.println(a); | Не нужно сравнивать логическую переменную (типа boolean) c **true** или **false**. Результат сравнения сам будет иметь тип boolean, и в точности будет совпадать со значением сравниваемой переменой: true == true – истина. Результат выражения – true. true == false – ложь. Результат сравнения – false. |
| 6 | **boolean** m = (a > b);  if (m)  System.out.println(a); |

— Еще примеры:

|  |  |  |
| --- | --- | --- |
|  | **Код** | **Пояснение** |
| 1 | public **boolean** isALessThanB (int a, int b)  {  if (a < b)  return true;  else  return false;  } | Данный метод проверяет, что число А меньше числа Б.  Показаны четыре эквивалентных записи сравнения. Внизу самая компактная и корректная. Всегда старайся использовать компактную запись. |
| 2 | public **boolean** isALessThanB (int a, int b)  {  boolean m = (a < b);  if (m)  return true;  else  return false;  } |
| 3 | public **boolean** isALessThanB (int a, int b)  {  **boolean m** = (a < b);  return m;  } |
| 4 | public boolean isALessThanB (int a, int b)  {  return a < b;  } |

2

Задача

Java Syntax,  4 уровень,  7 лекция

Набираем код Ӏ Java Syntax: 4 уровень, 7 лекция

Java Syntax: 4 уровень, 7 лекция. Иногда думать не надо, строчить надо! Как ни парадоксально звучит, порой пальцы «запоминают» лучше, чем сознание. Вот почему во время обучения в секретном центре JavaRush вы иногда встречаете задания на набор кода. Набирая код, вы привыкаете к синтаксису и зарабатываете немного материи. А ещё — боретесь с ленью.

— А что мне делать, если я хочу записать, выражение 0<a<b?

— Ну, выражений, которые включают три оператора, в Java нет, поэтому тут нужно воспользоваться такой конструкцией: (0<a) И (a<b)

— Так и писать И?

— Подожди, сейчас все объясню. В Java есть три логических оператора: AND (и), OR (или) и NOT (не). С помощью них можно строить условия различной сложности. Эти операторы можно применять только к выражению, имеющему тип boolean. Т.е. (a+1) AND (3) написать нельзя, а (a>1)AND (a<3) – можно.

— Оператор NOT – унарный – его действие распространяется только на выражение справа от него. Он больше похож на знак «минус» перед отрицательным числом, чем на знак умножить

— С переменными типа boolean (логический тип) можно выполнять различные операции.

— А какие?

— Вот сейчас мы их и рассмотрим:

|  |  |  |  |
| --- | --- | --- | --- |
| **Логический oператор** | **Обозначение в Java** | **Выражение** | **Результат** |
| AND (и) | && | true && true | true |
| true && false | false |
| false && true | false |
| false && false | false |
| OR (или) | || | true || true | true |
| true || false | true |
| false || true | true |
| false || false | false |
| NOT(не) | ! | ! true | false |
| ! false | true |
| Распространённые комбинации и выражения | | m && !m | false |
| m || !m | true |
| ! (a && b) | !a || !b |
| ! (a || b) | !a && !b |

— А можно побольше примеров?

— Конечно:

|  |  |  |
| --- | --- | --- |
| **Запись в Java** | **Логическая запись** | **Пояснение** |
| (a<3) && (a>0) | (a < 3) AND (a>0) | a<3 и a>0 |
| (a>10) || (a<100) | (a>10) OR (a<100) | a>10 или a<100 |
| (a<b) && (!(c<=d)) | (a<b) AND (NOT (c<=d)) | a<b и (не (c<=d)) |

— Реши пока немного задачек.

4

Задача

Java Syntax,  4 уровень,  7 лекция

Ярлыки и числа

Не все ярлыки одинаково полезны, но порой без них образуется хаос. Давайте решим задачу с раздачей ярлыков: пользователь вводит с клавиатуры целое число, а программа ему рассказывает, отрицательное это число или положительное (или нулевое), чётное или нечётное. Зачем нам это? Просто так, для тренировки. Мы же тут учимся, в конце концов!

8

Задача

Java Syntax,  4 уровень,  7 лекция

Описываем числа

Представим себе, что нам нужно разделить числа по каким-то признакам. Например, по чётности или по количеству разрядов. Представили? А теперь за дело: пишем программу, которая описывает введенное пользователем число (оно должно находиться в диапазоне от 1 до 999), указывая в строке вывода, чётное ли это число и сколько в нём цифр.

4

Задача

Java Syntax,  4 уровень,  7 лекция

Положительное число

В школе «Позитивчик» шкала оценки знаний не ограничена ни снизу, ни сверху. Это радует юные дарования (а вы бы не радовались, получив оценку 100500 за изложение?). Все оценки больше нуля считаются хорошими, меньше — плохими. На зачете ученики должны сдать три предмета. Нам нужно ввести три оценки с клавиатуры, и вывести количество позитивных оценок.

4

Задача

Java Syntax,  4 уровень,  7 лекция

Положительные и отрицательные числа

Анализ данных начинается с малого: мы должны понимать, какие данные нам важны и их нужно использовать, а какие нужно отбросить. Решим простую задачу на анализ данных. Введем с клавиатуры три целых числа, затем проанализируем их и выведем на экран количество положительных и отрицательных чисел в исходном наборе.