**Напишите программу, вводящую три целых числа, и печатающую Yes в том случае, если среди введенных чисел есть одинаковые, и No — иначе.**

***Текст программы***

public class Equal3v1

{

public static void main(String[] args) throws Exception

{

int a = Xterm.inputInt("Введите первое число -> ");

int b = Xterm.inputInt("Введите второе число -> ");

int c = Xterm.inputInt("Введите третье число -> ");

if ( (a == b) || (a == c) || (b == c) )

Xterm.println("Yes");

else

Xterm.println("No");

}

}

Обратите внимание, что в данной программе использованы операторы условного **Или** ||, а не логического **Или** |. Это вполне типично — операторы логического **Или** | и логического **И** & на практике не используют — вместо них применяют условные операторы || и &&.

Как это следует из определений, если первый операнд дизъюнкции истинен, то независимо от значения второго операнда результатом будет истина. Аналогично в случае конъюнкции при ложном первом операнде значение второго операнда на результат не влияет — он всегда будет ложным. При выполнении условных операторов || и && исполняющая система Java не производит оценку второго операнда логического выражения, если результат ясен из значения первого операнда. Иногда это просто ускоряет вычисления, а иногда позволяет добиться и большего, как, например, в следующем программном фрагменте.

if (a == 0 || b / a > 0) x = y;

При **a** **= 0** второй операнд оператора || вычисляться не будет и деления на ноль не произойдет, как это было бы в случае использования логического оператора **Или** |.

Можно изменить проверяемое условие **среди чисел есть равные** в рассмотренной выше программе на его отрицание **среди чисел нет равных** и переписать программу, заменяя оператор **Или** на **И**.

**Фрагмент программы (*Equal3v2.java*)**

if ((a != b) && (a != c) && (b != c))

Xterm.println("No");

else

Xterm.println("Yes");

При решении рассматриваемой задачи можно обойтись без условных операторов, если вспомнить, что произведение чисел равно нулю тогда и только тогда, когда одно из них нулевое. Это позволяет написать следующую программу.

**Фрагмент программы (*Equal3v3.java*)**

if ((a - b) \* (b - c) \* (c - a) != 0)

Xterm.println("No");

else

Xterm.println("Yes");

И, наконец, заметим, что программа запишется короче, если заменить в ней оператор if-else на тернарный оператор условия ?:, общая форма записи которого имеет следующий вид:

выражение1 ? выражение2 : выражение3

Если результат вычисления первого выражения истинен, то выполняется выражение2 (второй операнд), а иначе — выражение3 (третий операнд). При использовании этой конструкции два последних ее выражения должны иметь один и тот же тип, в данном случае — строковый.

**Фрагмент программы (*Equal3v4.java*)**

Xterm.println( (a-b)\*(b-c)\*(c-a) != 0 ? "No" : "Yes" );