УСЛОВНЫЕ ОПЕРАТОРЫ

Конструкция if/else

Выражение if/else проверяет истинность некоторого условия и в зависимости от результатов проверки выполняет определенный код:

```
int num1 = 6;

int num2 = 4;

if(num1>num2){

    System.out.println("Первое число больше второго");

}
```

Но что, если мы захотим, чтобы при несоблюдении условия также выполнялись какие-либо действия? В этом случае мы можем добавить блок else:

```
int num1 = 6;

int num2 = 4;

if(num1>num2){

    System.out.println("Первое число больше второго");

}

else{

    System.out.println("Первое число меньше второго");

}
```

Но при сравнении чисел мы можем насчитать три состояния: первое число больше второго, первое число меньше второго и числа равны. С помощью выражения else if, мы можем обрабатывать дополнительные условия:

```
int num1 = 6;
int num2 = 8;
if(num1>num2){
    System.out.println("Первое число больше второго");
}
else if(num1<num2){
    System.out.println("Первое число меньше второго");
}
else{
    System.out.println("Числа равны");
}
```

Также мы можем соединить сразу несколько условий, используя логические операторы:

```
int num1 = 8;
int num2 = 6;
if(num1 > num2 && num1>7){
    System.out.println("Первое число больше второго и больше 7");
}
```

ОПЕРАТОР	ПРИМЕР ИСПОЛЬЗОВАНИЯ	ВОЗВРАЩАЕТ ЗНАЧЕНИЕ "ИСТИННО", ЕСЛИ...
>	a > b	a больше b
>=	a >= b	a больше или равно b
<	a < b	a меньше b
<=	a <= b	a меньше или равно b
==	a == b	a равно b
!=	a != b	a не равно b
&&	a && b	a и b истинны, b оценивается условно (если a ложно, b не вычисляется)
||	a || b	a или b истинно, b оценивается условно (если a истинно, b не вычисляется)
!	!a	a ложно
&	a & b	a и b истинны, b оценивается в любом случае
|	a | b	a или b истинно, b оценивается в любом случае

Конструкция switch

Конструкция **switch/case** аналогична конструкции if/else, так как позволяет обработать сразу несколько условий:

```
int num = 8;

switch(num){

    case 1:

        System.out.println("число равно 1");

        break;

    case 8:

        System.out.println("число равно 8");

        num++;

        break;

    case 9:

        System.out.println("число равно 9");

        break;

    default:

        System.out.println("число не равно 1, 8, 9");

}
```

После ключевого слова **switch** в скобках идет сравниваемое выражение. Значение этого выражения последовательно сравнивается со значениями, помещенными после операторов **case**. И если совпадение найдено, то будет выполняет соответствующий блок **case**.

В конце блока case ставится оператор **break**, чтобы избежать выполнения других блоков. Например, если бы убрали оператор break в следующем случае:

```
case 8:
    System.out.println("число равно 8");
    num++;
case 9:
    System.out.println("число равно 9");
    break;
```

то выполнился бы блок case 8, (поскольку переменная num равна 8). Но так как в этом блоке оператор break отсутствует, то начал бы выполняться блок case 9.

Если мы хотим также обработать ситуацию, когда совпадения не будет найдено, то можно добавить блок **default**, как в примере выше. Хотя блок **default** необязателен.

Также мы можем определить одно действие сразу для нескольких блоков case подряд:

```
    case 2:
    case 3:
    case 4:
        output = 6;
        break;
```

Тернарная операция

Тернарная операция имеет следующий синтаксис:

[первый операнд - условие] ? [второй операнд] : [третий операнд].

Задача

Треугольник существует только тогда, когда сумма любых двух его сторон больше третьей.

Дано: a, b, c – стороны предполагаемого треугольника.

Требуется сравнить длину каждого отрезка-стороны с суммой двух других. Если хотя бы в одном случае отрезок окажется больше суммы двух других, то треугольника с такими сторонами не существует.

Из двух чисел с разной четностью вывести на экран нечетное число.

Всем известна прямоугольная (декартова) система координат, в которой две перпендикулярные оси делят плоскость на четверти. В первую четверть попадают точки, у которых обе координаты (*x* и *y*) больше нуля. Во вторую: $x < 0, y > 0$; третью: $x < 0, y < 0$; четвертую: $x > 0, y < 0$.

Тнаписать программу, определяющую по координатам точки, в какой четверти она находится.

Определить какое из трех чисел максимальное и вывести его на экран.

Требуется написать программу, вычисляющую значение какой-либо функции $y = f(x)$. Допустим, такой:

$y = x - 2$, *если* $x > 0$,
$y = 0$, *если* $x = 0$,
$y = |x|$, *если* $x < 0$.

Чтобы получить модуль числа воспользуйтесь функцией abs в