

Операторы

Арифметические операторы

+	Сложение
-	Вычитание
*	Умножение
/	Деление

Арифметические операторы записываются аналогично обычным математическим выражениям:

$2 + 2$

$3 * 15$

$3 - 5 * 4$

Отдельно следует упомянуть оператор % – остаток от целочисленного деления. В результате выполнения этого оператора получается остаток, например, $7\%5$ вернет остаток равный 2.

Операторы сравнения

Позволяют сравнить два значения, и получить результат в виде логического значения, то есть "истина" или "ложь".

Оператор == позволяет сравнить два значения на равенство. Если значения, стоящие по обе стороны от него, равны, результатом будет истина, если они не равны, результатом будет ложь. Например:

$x == 10$

$x == y$

$y * 2 == x - 10$

В первом случае результатом будет истина, если значение переменной x равно десяти. Во втором случае результатом будет истина, если значения в переменных x и y совпадают. В третьем случае результатом будет истина, если совпадают результаты выражений справа и слева.

Оператор "не равно" !=

$x \neq 10$

Операторы "меньше" $<$ и "больше" $>$

$x > 10$

$x < y$

Есть также операторы "больше либо равно" \geq и "меньше либо равно" \leq

Логические операторы

&& - "логическое и"

Пример:

$1 < x \ \&\& \ x < 5$

Только если оба выражения, стоящие слева и справа от него, являются истинными, тогда все выражение истинное.

|| - "логическое или"

Пример:

$5 < x \ || \ x < 1$

Если хотя бы одно из выражений, стоящее справа и слева от него, будет истинным, то все выражение будет истинным.

С помощью логических операторов можно соединять не только простые условия, но и другие логические выражения. В логических выражениях также можно ставить скобки, чтобы управлять последовательностью действий.

! - "не"

Пример:

$!(1 < x)$

Результат будет истиной, если x будет меньше либо равен 1.

Оператор присваивания

$x = y * 2 / z;$

$x = 3;$

$x = y;$

Также допустимо использовать переменную, стоящую слева в правой части выражения. Например, чтобы увеличить значение x в два раза, следует записать:

$x = x * 2;$

В этом случае в правом выражении используется старое значение переменной, а потом в нее же заносится результат.

Существуют сокращенные формы записи подобных операторов. Они выглядят как:

$+=$ Сложение с присваиванием

$-=$ Удаление с присваиванием

$*=$ Умножение с присваиванием

$/=$ Деление с присваиванием

$\%=$ Деление с остатком с присваиванием

Например, две следующие строчки идентичны по выполняемым действиям:

$x += 2;$

$x = x + 2;$

Унарные операторы

Обычные арифметические операторы являются бинарными, так как имеют две части – правую и левую.

Унарные записываются как два знака плюс для инкремента и два знака минус для декремента.

$i++;$

В данном примере переменная i увеличивается на единицу

Знаки $++$ и $--$ можно ставить как после переменной, так и до. Разница существует в том случае, если оператор используется внутри выражения. Допустим имеется следующий код:

```
int i = 10;
```

```
int j = 10;
```

```
int z = 10 + i++;
```

```
int y = 10 + ++j;
```

В двух последних выражениях выполняются аналогичные операции, значения переменных *i* и *j* равны 10. Но в результате в *z* попадет 20, а в *y* – 21.

Особенности операторов при работе со строками:

При сложении двух строк происходит их склеивание:

```
String str = "Hello ";
```

```
String newStr = str + "world!";
```

В результате в переменной *newStr* будет находиться значение "Hello world!".

Пример использования арифметических операторов

Имеется промежуток времени в секундах. Следует вывести его на страницу в виде часов минут и секунд.

```
public class HelloWorld {  
    public static void main(String[] args) { int s = 3700;  
        int sec = s % 60;  
        int m = (s - sec) / 60; int min = m % 60;  
        int h = (m - min) / 60;  
        System.out.println(h + " часов " + min + " минут " + sec + " секунд");  
    }  
}
```