**Логические значения. Тип boolean**

Примитивный тип boolean предназначен для хранения логических значений. Данный тип может принимать одно из двух возможных значений: true (истина) или false (ложь).

Значения boolean возвращаются во всех логических операциях (например, операции сравнения). Применяется при построении циклов.

**Символьный тип данных**

Символьный тип данных представляет собой один 16-битный Unicode символ. Unicode — это объединение десятков кодировок символов, он включает в себя латинский, греческий, арабский алфавиты, кириллицу и многие другие наборы символов.

Диапазон допустимых значений - от 0 (‘\ u0000’)  до 65536 (‘\ uffff’). Отрицательных значений для типа char не существует. Это единственный беззнаковый целочисленный тип данных в Java.

Символы char  можно задавать при помощи символа, заключенного в одинарные кавычки, а можно с помощью соответствующего кода-числа из таблицы Unicode-символов. Например символ ‘Ы’ соответствует числу  1067.

**Преобразование типов**

Иногда возникают ситуации, когда у вас есть величина какого-то определенного типа, а вам нужно ее присвоить переменной другого типа. Для некоторых типов это можно проделать и без приведения типа, в таких случаях говорят об автоматическом (неявном) преобразовании типов. В других случаях программист должен сам провести преобразование типа к нужному — это называется явным преобразованием типов.

**Автоматические (неявные) преобразования**

При изучении типов данных уже говорилось о том, что целочисленные типы данных byte и short при вычислении Java всегда приводит к типу int, а тип float к типу double. Фактически это и есть неявное преобразование типов.

Стрелками на рисунке показано, какие преобразования типов могут выполняться автоматически без потери точности. Пунктирными стрелками показаны автоматические преобразования с потерей точности.

Преобразования от меньшего типа к большему называются расширяющими и именно эти преобразования проходят без потери данных и без потери точности.

Можно сказать, что существуют следующие правила повышения типов

Если в выражении используются переменные типов byte, short и int, то во избежание переполнения тип всего выражения автоматически повышается до int. Если же в выражении тип хотя бы одной переменной - long, то и тип всего выражения тоже повышается до long.

Если выражение содержит операнды типа float, то и тип всего выражения автоматически повышается до float. Если же хотя бы один из операндов имеет тип double, то тип всего выражения повышается до double. По умолчанию Java рассматривает все литералы с плавающей точкой, как имеющие тип double.

Преобразования от большего типа к меньшему называются сужающими и в этих случаях вся ответственность по переносу данных ложится на программиста.

Явные преобразования

Все остальные преобразования типов, не рассмотренные ранее, осуществляются явным образом, а именно указанием нужного типа, заключенного в круглые скобки. Обычно это сужающие преобразования от типа с большей разрядностью к типу с меньшей разрядностью. При явном преобразовании программист должен внимательно оценить все риски потери данных и выбрать тип, который сведет эти риски к нулю.

```java

//Пример кода

byte b = 50;

b = b\* 2;

b = (byte) (b\* 2);

Ошибка: Error:(5, 14) java: incompatible types: possible lossy conversion from int to byte

```

При преобразовании значений с плавающей точкой к целочисленным значениям, происходит усечение дробной части.

```java

//Пример кода

double a = 56.9898;

int b = (int)a;

System.out.println(b); // 56

```

## Арифметические операции

Большинство операций в Java являются стандартными математическими операциями. Есть унарные операции (выполняются над одним операндом), бинарные - над двумя операндами, а также тернарные - выполняются над тремя операндами. Операндом является переменная или значение (например, число), участвующее в операции.

### Унарные арифметические операции

Унарные арифметические операции — это операции, которые производятся над одним числом:

++ (инкремент) и -- (декремент). Инкремент подразумевает под собой увеличение значения переменной на единицу, декремент — наоборот, уменьшение значения переменной на единицу.

Каждая из операций имеет две разновидности: префиксная и постфиксная. Префиксный тип операции означает, что сначала значение переменной увеличивается на 1, а затем это значение переменной передается дальше. Постфиксный тип операции означает, что сначала значение переменной передается дальше, а затем увеличивается на 1.

```java

//Пример кода

int a = 8;

int b = ++a;

System.out.println(a);// 9

System.out.println(b);// 9

a = 8;

b = a++;

System.out.println(a);// 9

System.out.println(b);// 8

a = 8;

b = --a;

System.out.println(a);// 7

System.out.println(b);// 7

a = 8;

b = a--;

System.out.println(a);// 7

System.out.println(b);// 8

```

### Бинарные арифметические операции

К бинарным арифметическим операциям относятся:

\* вычитание («-»)

\* сложение («+»)

\* умножение («\\*»)

\* деление («/»)

\* остаток от деления («%»)

```java

//Пример кода

int a = 10;

int b = 7;

int c = a + b;// 17

int d = 4 - a;// -6

int k = b \* 5;// 35

int i = a / b;// 1

double k = 10 / 4;// 2

double k = 10.0 / 4;// 2.5

int j = a % b;// 3

```

### Приоритет арифметических операций

Одни операции имеют больший приоритет чем другие и поэтому выполняются вначале. Операции в порядке уменьшения приоритета:

\* ++ (инкремент), -- (декремент)

\* \\* (умножение), / (деление), % (остаток от деления)

\* \+ (сложение), - (вычитание)

Приоритет операций следует учитывать при выполнении набора арифметических выражений. Чтобы задать свой порядок вычислений, нужно использовать ().

```java

//Пример кода

int a = 8;

int b = 7;

int c = a + 5 \* ++b;

System.out.println(c);// 48

int d = (a + 5) \* ++b;

System.out.println(d);// 104

```

### Операции с присваиванием

Операцию присваивания можно скомбинировать с любой арифметической операцией и записать сокращенно, в виде рядом стоящих знаков присваивания и используемой арифметической операции.

```java

//Пример кода

int c = 3;

c += 5; //8

c -= 2; //6

c /= 3; //2;

```

Для выполнения различных математических операций в Java в пакете java.lang определен класс Math.

В классе определены две константы типа double: E и PI.

И большое многообразие математических функций:

| Тип | Метод | Описание |

|---------|-------------------------|-------------------------------------------------------------------------|

| double | abs(double a) | Возвращает абсолютное значение (модуль) числа типа double |

| double | exp(double a) | Возвращает экспоненту аргумента |

| double | max(double a, double b) | Возвращает больший из аргументов |

| double | random() | Возвращает случайное число от 0.0 (включительно) до 1 (не включительно) |

| double | sin(double a) | Возвращает синус аргумента |

| double | toRadians() | Преобразует градусы в радианы |