

ПЕРЕМЕННЫЕ

ПРИМИТИВНЫЕ ТИПЫ ДАННЫХ

ПРИВЕДЕНИЕ ТИПОВ

Каждая программа в конечном итоге сводится к манипулированию данными.

Для того, чтобы осуществлять манипуляции с данными, используя возможности языка, программа должна сначала где-то сохранить эти данные.

Данные хранятся в переменных.

Переменная - поименованная выделенная область памяти.

В языке Java все переменные должны быть объявлены, перед тем, как они будут использоваться.

Объявление переменных:

1. тип данных, которые будут в ней храниться
2. имя переменной

По имени переменной и осуществляется доступ к данным, которые в ней хранятся.

Объявление переменных, присваивание значений

```
1  int messageId; // объявили переменную типа int, имя - messageId
2
3  int a, b; // объявили несколько переменных одного типа
4  a = 4; // переменной a присвоили значение 4
5  a = 500000; // переопределили значение переменной a
6
7  double pi = 3.14; // объявили переменную и присвоили ей значение
8
9  int size = 20, page = 3; // объявили переменные и присвоили значения
10 size = 10; // переопределили значение переменной size
11 page = 5; // переопределили значение переменной page
12
13 // обратились к переменной pi по имени, вывели его в консоль
14 System.out.println(pi);
```

Требования к именам переменных (обязательные):

1. имя переменной должно начинаться с буквы (технически может начинаться со \$ или _)
2. имя переменной должно состоять из букв (Unicode), цифр и символа _
3. пробелы при именовании переменных не допускаются
4. имя переменной не должно быть ключевым или зарезервированным словом
5. имена переменных чувствительны к регистру

Требования к именам переменных (по соглашению Java Code Conventions):

1. по соглашению запрещено начинать имя переменной со _ и \$
2. символ доллара \$, по соглашению, никогда не используется
3. имя переменной должно начинаться с маленькой буквы
4. при выборе имен переменных, следует использовать полные слова
5. если имя переменной состоит из более чем одного слова, то отделяйте каждое последующее слово в имени переменной заглавной буквой

Типы данных, используемые в Java:

- Примитивные типы данных
- Ссылочные типы данных

Примитивные типы данных. Целые числа.

Тип	Размер	Диапазон
byte	1 байт	от -128 до 127
short	2 байта	от -32768 до 32767
int	4 байта	от -2147483648 до 2147483647
long	8 байт	от -9223372036854775808 до 9223372036854775807

Примитивные типы данных.
Числа с плавающей точкой.

Тип	Размер	Диапазон
float	4 байта	от $-3.4E+38$ до $3.4E+38$
double	8 байт	от $-1.7E+308$ до $1.7E+308$

Примитивные типы данных.

Символ юникода.

Тип	Размер	Диапазон
<code>char</code>	2 байта	от '\u0000' до '\uffff' от 0 до 65535

Примитивные типы данных.

Логические значения.

Тип	Размер	Значения
<code>boolean</code>	зависит от реализации JVM 4 байта / 1 байт в массивах	<code>true</code> / <code>false</code>

Некоторые особенности примитивных типов данных:

- результат работы арифметических операторов над `byte` и `short` преобразуется к `int`

```
1 byte n = 12, m = 34;  
2 // byte multiRes = n * m; - ошибка  
3 int multiRes = n * m;
```

- рекомендуется добавлять символ `L` или `l` для обозначения типа `long`

```
1 long population = 300000000000L;
```

- при целочисленном делении остаток отбрасывается

```
1 int a = 8, b = 3;  
2 int res = a / b; // значение res будет равно 2
```

- деление целого числа на ноль приведет к ошибке

```
1 int a = 8, b = 0;  
2 int res = a / b; // получим java.lang.ArithmeticException
```

- деление числа с плавающей точкой на ноль

```
1 double a = 8.1, b = 0;  
2 double res = a / b; // значение res будет равно Infinity
```


- рекомендуется добавлять символ F или f для обозначения типа float

```
1 float average = 5.9f;  
2 average = 98.31F;
```

- для точных вычислений над double и float необходимо использовать класс `java.math.BigDecimal`

Литерал — это представление исходного кода с фиксированным значением. Они представлены непосредственно в коде без каких-либо вычислений.

```
1 byte a = 68;  
2  
3 int price = 1_000_000;  
4  
5 int decimal = 100; // десятичная сс  
6 int octal = 0144; // восьмиричная сс, начинается с 0  
7 int hex = 0x64; // шестнадцатеричная сс  
8  
9 boolean done = false;
```


Приведение типов в Java

1. Автоматическое преобразование
2. Явное приведение типов

Автоматическое преобразование типов

осуществляется, если оба типа совместимы и длина целевого типа больше длины исходного типа.

В этом случае происходит преобразование с расширением.

```
1 byte a = 15;  
2 int b = a;    // a автоматически преобразовано к int
```

Если длина целевого типа меньше длины исходного типа, необходимо использовать **явное приведение ТИПОВ**.

При явном приведении типов перед целевым значением необходимо в круглых скобках указать целевой тип данных.

```
1 long bigCount = 4230;  
2 int count = (int) (bigCount);
```

Правила приведения типов:

1. При приведении `float` или `double` к целочисленным типам, дробная часть не округляется, а просто отбрасывается.
2. Тип `boolean` не приводится ни к одному из типов.
3. Тип `char` приводится к числовым типам, как код символа в системе UNICODE.
4. Если число больше своего контейнера, результат будет непредсказуемым.

Сплошные линии обозначают преобразования, выполняемые без потери данных.

Штриховые линии говорят о том, что при преобразовании может произойти потеря точности.

