

# Примитивные типы данных.

Группа: ИО14-1

Подготовил: Мокану Г.С.

Проверил(а): Строганова С.М.

# Введение

Тип данных определяется тремя составляющими:

1. Набором значений или объектов;
2. Набором операций, которые могут быть применены ко всем значениям в наборе;
3. Представлением данных, определяющим их хранение.

В Java типы данных делятся на примитивные и ссылочные.

«Примитивный» означает, что дальнейшее его деление невозможно. Расширять или изменять его язык программирования не позволяет. Такой тип данных описывается примитивными и другими пользовательскими типами.

Переменная примитивного типа содержит значение, а ссылочная – адрес объекта в памяти.

## Введение

Язык программирования содержит некоторые заранее определенные встроенные типы и позволяет программистам определять свои собственные, пользовательские.

# Типы данных: Ссылочная

## Время и дата

Составные данные делятся на классы, интерфейсы и массивы. Членами интерфейсного типа являются абстрактные методы и константы. В языке Java типы данных дата и время задаются конструктором `Date()`:

```
d = new Date();
```

Примером ссылки также является строка.

## String

Объявление ссылочной String-переменной следующее: `String str`.

```
str = new String ("Привет").
```

```
String str1;
```

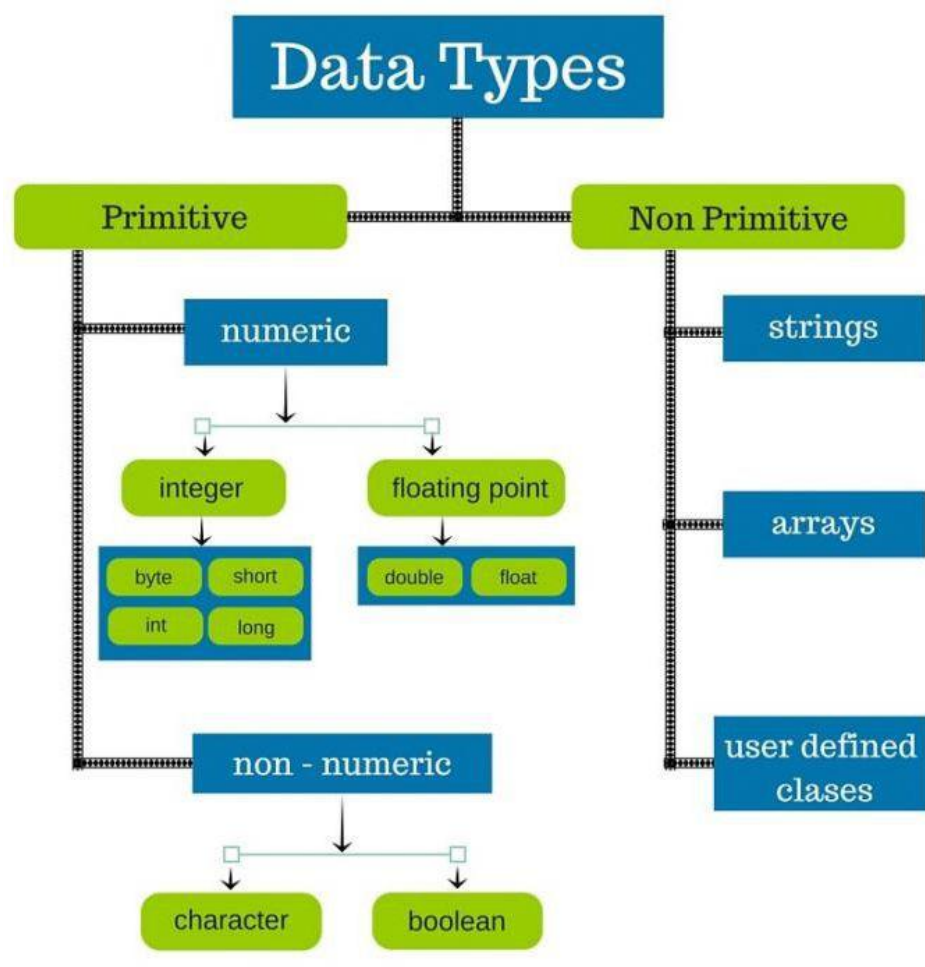
```
String str2;
```

```
str1 = new String  
("Привет");
```

```
str2 = str1;
```

# Типы данных: Примитивные

Целые типы данных Java представляют собой числовые типы, значения которых являются целыми числами. Их пять: **byte**, **short**, **int**, **long** и **char**.



# Типы данных: Примитивные

## Int

Int представляет собой 32-разрядный знаковый примитивный тип данных. Переменная занимает 32 бита памяти. Допустимый диапазон – от -2147483648 до 2147483647 (-231 до 231 - 1). Все целые числа в этом диапазоне являются целочисленными литералами или константами. Например, 10, -200, 0, 30, 19 являются литералами int. Их можно присвоить переменной int:

[Obj]

```
int num1 = 21;
```

Целые литералы могут быть выражены в виде двоичного, восьмеричного, десятичного и шестнадцатеричного числа.

Когда литерал начинается с нуля и имеет, по крайней мере, две цифры, он считается записанным в восьмеричном формате. 0 и 00 представляют одинаковое значение – ноль.

## Long

Это 64-разрядный знаковый примитивный тип. Используется, когда результат вычислений может превышать диапазон int. Диапазон long – от -263 до 263- 1. Все целые числа этого диапазона являются литералами длинного типа.

Чтобы отличать в языке Java типы данных int и long, литерал последнего типа всегда заканчивается L или l.

Поскольку диапазон int меньше, чем у long, значение int-переменной всегда может быть присвоено переменной типа long. Но обратное присвоение невозможно даже в пределах диапазона int. Для этого используется явное указание:

```
num1 = (int) num2;
```

# Типы данных: Примитивные

## Byte

Byte представляет собой 8-разрядный целый примитивный тип. Его диапазон - от -128 до 127 ( $-2^7$  до  $2^7 - 1$ ). Это наименьший целый тип, имеющийся в Java. Как правило, байтовые переменные используются, когда в программе задействовано много значений в диапазоне от -128 до 127, или при работе с бинарными данными. В отличие от литералов `int` и `long`, `byte`-литералы отсутствуют. Тем не менее можно назначить `byte`-переменной любой `int`-литерал, так как он перекрывает диапазон байта.

- Если значение переменной выходит за пределы диапазона, Java выдаст ошибку компилятора.
- Кроме того, можно присвоить только `int`-литерал, но не значение, хранящееся в переменной `int`, так как при этом возможна потеря точности. Для этого потребуется явное приведение типа.

`b1 = (byte) num1.`

## Short

Представляет собой 16-разрядный знаковый целый примитивный тип данных. Его диапазон – от -32768 до 32767 (или  $-2^{15}$  до  $2^{15} - 1$ ).

Как правило, необходимость в коротких переменных возникает, когда программа использует большое количество значений, которые не превышают указанный диапазон. `Short`-литерал отсутствует, но возможно присвоение любого литерала `int` в пределах диапазона `short`. Значение `byte`-переменной может быть присвоено всегда. Остальные правила присвоения `int` или `long` `short`-переменной такие же, как для `byte`.

Символ может быть выражен путем заключения его в одинарные кавычки:

```
char C1 = 'A'.
```

Двойные кавычки обозначают строковый литерал, который не может быть присвоен char-переменной, даже если строка состоит только из одного символа. Это недопустимо, так как ссылка примитивной переменной не присваивается.

Все строковые литералы являются объектами класса String и, следовательно, ссылками, тогда как символьные относятся к примитивному типу.

Литерал, выраженный управляющей последовательностью, записывается в виде обратной косой черты с символом в одиночных кавычках. Всего их 8: '\n', '\r', '\f', '\b', '\t', '\\', '\"', '\\'.

Управляющая юникод-последовательность имеет вид '\uxxxx', где \u (обратный слеш, за которым следует строчная u) обозначает ее начало, а xxxx представляет ровно четыре шестнадцатеричные цифры кода символа в системе Юникод. Например, 'A' имеет значение 65 в десятичной системе и 41 в шестнадцатеричной. Таким образом, этот символ может быть представлен как '\u0041'.

Восьмеричная управляющая последовательность записывается в виде '\nnn', где n – восьмеричная цифра (0-7). Диапазон значений – от '\000' до '\377', где 377<sub>8</sub> соответствует 255<sub>10</sub>. Поэтому она используется для представления символов с кодом от 0 до 255, необходимого для совместимости с другими языками программирования. В отличие от юникод-последовательности, где необходимы все четыре шестнадцатеричные цифры, здесь можно использовать 1, 2 или 3 восьмеричные: '\n', '\nn' или '\nnn'.

# Типы данных: Примитивные

## Char

- СИМВОЛ, ЗАКЛЮЧЕННЫЙ В одинарные кавычки;
- последовательность управляющих символов;
- последовательность управляющих символов юникода;
- последовательность восьмеричных управляющих символов.



Спасибо за внимание