Java — Примитивные и ссылочные типы данных, литералы

Переменные не что иное как зарезервированные места памяти для хранения значений. Это означает, что при создании переменной Вы резервируете некоторое пространство в памяти.

Основываясь на типе данных, который присвоен переменной, операционная система выделяет память и решает, что может быть сохранено в зарезервированную памяти. Поэтому, назначая различные типы данных для переменных, в Java можно хранить целые числа, десятичные дроби или символов в этих переменных.

Существует два типа данных в Java:

- простые или примитивные типы данных;
- ссылочные типы данных (ссылка/объект).

Содержание [скрыть]

Примитивные типы данных

Есть **восемь типов данных**, поддерживаемых Java. Основные типы данных предопределены языком и названы по ключевому слову. Теперь давайте посмотрим в деталях эти восемь базовых типов данных существующих в языке программирования Java.

Тип byte

- Тип данных byte является 8-разрядным знаковым целым числом.
- Минимальная величина равна -128 (-2^7).
- Максимальное значение равно 127 (включительно) (2^7-1).
- По умолчанию – 0.
- byte предназначен для экономии места в больших массивах, главным образом вместо целых чисел, поскольку byte в четыре раза меньше, чем int.
- Пример:

```
byte a = 100;
byte b = -50;
```

Тип short

- Тип данных short является 16-разрядным знаковым целым числом.
- Минимальное значение равно -32768 (-2^{15}).
- Максимальная величина равна 32 767 (включительно) (2^{15}-1).
- Тип short в Java может также использоваться для экономии памяти как byte. Размер short в 2 раза меньше, чем int.
- По умолчанию — 0.
- Пример:

```
short s = 10000;
short r = -20000;
```

Тип int

- В языке Java тип данных int является 32-разрядным знаковым целым числом.
- Минимальный размер int — 2 147 483 648 (-2^{31}).
- Максимальная величина равна 2,147,483,647 (включительно)(2^{31}-1).

- Тип int обычно используется для целых значений. Если нет озабоченности по поводу памяти.
- По умолчанию равно 0.
- Пример:

```
int a = 100000;
int b =-200000;
```

Тип long

- Тип данных long является 64-разрядным знаковым целым числом.
- Минимальное значение равно — 9,223,372,036,854,775,808 (-2^{63}).
- Максимальная величина — 9,223,372,036,854,775,807 (включительно). (2^{63}-1).
- В Java Применяется когда требуется более широкий диапазон, чем int.
- По умолчанию — 0L.
- Пример:

```
long a = 100000L;
long b =-200000L;
```

Тип float

- Тип данных float является с одинарной точностью 32-битный IEEE 754 с плавающей точкой.
- Тип float используется главным образом для сохранения памяти в больших массивах чисел с плавающей точкой.
- По умолчанию — 0.0f.
- Тип float никогда не должен применяется для точного значения, например, валюты.
- Пример:

```
float f1 = 234.5f;
```

Тип double

- Тип данных double является с двойной точностью 64-битный IEEE 754 с плавающей точкой.
- Обычно используется для десятичных значений.
- Тип double никогда не должен применяется для точного значения, например, валюты.
- По умолчанию — 0.0d.
- Пример:

```
double d1 = 123.4;
```

Тип boolean

- Тип данных boolean представляет собой один бит информации.
- Существует только два возможных значения: true и false.
- Предназначен для простых признаков, которые позволяют отслеживать условия true или false.
- По умолчанию — false.
- Пример:

```
boolean one = true;
```

Тип char

- Тип данных char является одним 16-разрядным символом Юникода.
- Минимальная величина равна «\u0000» (или 0).
- Максимальная величина равна «\uffff» (или 65535 включительно).
- В Java char нужен для хранения любого символа.
- Пример:

```
char letterA ='A';
```

Ссылочные типы данных

- Ссылочные переменные создаются с использованием определенных конструкторов классов. Они предназначены для доступа к объектам. Эти переменные объявляются с определенным типом, который не может быть изменен. Например, Employee, Puppy и т.д.
- Объекты класса и различные виды переменных массива подпадают под **ссылочный тип данных**.
- По умолчанию в Java значение любой переменной ссылки - null.
- Ссылочная переменная может применяться для обозначения любого объекта, объявленного или любого совместимого типа.
- Пример:

```
Animal animal = new Animal("giraffe");
```

Литералы в Java

Литерал — представление исходного кода как фиксированное значение. Оно представлено непосредственно в коде без каких-либо вычислений.

Литерал в Java можно назначить любой переменной из основного типа. Например:

```
byte a = 68;
char a = 'A';
```

Byte, int, long, и short может выражаться десятичной (основание 10), шестнадцатеричной (основание 16) или восьмеричной (основание 8) системой исчисления.

При использовании литералов в Java, префикс 0 применяться для указания восьмеричной системы, а префикс 0x указывает на шестнадцатеричную систему. Например:

```
int decimal = 100;
int octal = 0144;
int hexa =  0x64;
```

Строковые литералы в языке Java задаются как и в большинстве других языков, заключив последовательность символов между парой двойных кавычек. Примеры строковых литералов:

```
"Hello World"
"two\nlines"
"\"This is in quotes\""
```

Типы литералов String и char могут содержать любые символы Юникода. Например:

```
char a = '\u0001';
String a = "\u0001";
```

Язык Java поддерживает несколько специальных управляющих последовательностей для литералов String и char:

Обозначение	Представление
\n	Новая строка (0x0a)
\r	Возврат каретки (0x0d)
\f	Прогон страницы (0x0c)
\b	Возврат на шаг (0x08)
\s	пробел (0x20)
\t	Табуляция
\"	Двойная кавычка
\'	Апостроф
\\	Обратная косая черта
\ddd	Восьмеричный символ (ddd)
\uxxxx	Шестнадцатеричный символ UNICODE (xxxx)

Более подробно управляющие последовательности с примерами рассмотрим в следующих уроках.

Следующий урок объясняет различные типы переменных и их использование. Это даст Вам хорошее представление о том, как они могут использоваться в java-классах, интерфейсах и т.д.