



Урок № 2

Курс: «Разработка приложений под мобильные устройства на основе Android»

Тема: Переменные, типы данных, операторы

План

1. Типы данных
2. Переменные. Константы и литералы
3. Операторы
 1. Понятие оператор
 2. Типы операторов
 3. Оператор присваивания
 4. Арифметические операторы

1. Типы данных

Понятие типа данных

Язык Java является строго типизированным. Каждая переменная и каждое выражение имеет свой определенный тип. Всего в Java существует восемь простых типов данных, которые условно можно разделить на четыре группы:

- Целые – описывает целые числа с учетом знака (byte, short, int, long).
- Числа с плавающей запятой – описывает «дробные» числа (float и double).
- Символьный тип – это тип char, который предназначен для описания символов.
- Логический тип – это тип boolean, описывающий переменные, которые могут принимать значение «правда» (true) или «ложь» (false).

Размер и диапазон значений

| Тип | Разрядность | MIN | MAX |
|--------|-------------|--------------------------|-------------------------|
| byte | 8 бит | -128 | 127 |
| short | 16 бит | -32768 | 32767 |
| int | 32 бит | -2147483648 | 2147483647 |
| long | 64 бит | -922337203685477 5808 | 922337203685477 5807 |
| float | 32 бит | -3.4E+38 | 3.4E+38 |
| double | 64 бит | -1.7E+308 | 1.7E+308 |
| char | 16 бит | | |

Целые (Integers)

Тип позволяет хранить значения целых чисел, без дробной части.

Важной особенностью Java является возможность присвоения переменной не только значения в десятичной системе счисления, но а также в двоичной, восьмеричной и шестнадцатеричной.

Пример записи в разных системах счисления:

10 - значение 10 в десятичной системе счисления

0x10 - значение 10 в шестнадцатеричной системе счисления (16 в десятичной)

010 - значение 10 в восьмеричной системе счисления (9 в десятичной)

b10 - 10 в двоичной системе счисления (2 в десятичной)

Типы с плавающей запятой (Floating-point number)

Тип описывает число, состоящее из целой и дробной части. Соответствующие стандарту IEEE754-2008.

Пример записи: 10.5.

Указано значение 10 целых и 5 десятых

Следует помнить, что в Java, как и других языках программирования, дробная часть числа отделяется от целой точкой.

Символьный тип (Characters)

Используется для описания символьных констант и переменных. В Java занимает не 8, а 16 бит, что позволяет хранить в переменной этого типа любой символ Unicode. Значения данного типа необходимо заключать в одинарные кавычки ('').

Escape-последовательности:

`\xxx` – восьмиричный символ (xxx)

`\uxxxx` – шестнадцатеричный символ Юникод, где xxxx – номер символа.

`\'` – одиночная кавычка

`\''` – двойная кавычка

`\` – обратный слэш

`\r` – возврат каретки

`\n` – новая строка

`\f` – перевод страницы

`\t` – табуляция

`\b` – возврат на один символ (backspace)

Логический

Тип может принимать только 2 значения: `true` и `false` (англ. «правда» и «ложь»).

Используется в логических выражениях и условиях.

2. Переменные. Константы и литералы

Что такое переменная?

Переменная - это основной элемент хранения информации в Java- программе. Переменная характеризуется комбинацией идентификатора, типа и области действия. В зависимости от того, где объявлена переменная, она может быть локальной, например, для кода внутри метода, либо это может быть переменная экземпляра класса, доступная всем методам данного класса. Локальные области действия объявляются с помощью фигурных скобок.

Объявление переменной

Объявлению переменной происходит следующей последовательности:

- указание типа данных
- указание имя переменной
- инициализация переменной (присвоение значения, необязательно)

После каждой строки при объявлении переменных необходимо ставить точку с запятой «;».

Пример объявления переменных без инициализации:

```
int i;  
double d;
```

Пример объявления переменных с инициализацией:

```
int i = 5;  
double d = 5.25;
```

Область видимости переменной

Блоки составных операторов в Java отмечаются парой фигурных скобок `{}`. Переменные в Java начинают действовать с того места в программе, где они объявлены до конца содержащего их блока. Блоки могут быть вложены друг в друга, и у каждого может быть свой собственный набор локальных переменных.

Классы обертки

Если требуется создать ссылку на одни из примитивных типов данных, необходимо использовать соответствующий класс-обертку. Также в таких классах есть некоторые полезные методы и константы, например минимальное значение типа `int` можно узнать используя константу `Integer.MIN`.

Оборачивание примитива в объект называется боксингом (boxing), а обратный процесс анбоксингом (unboxing).

| Тип | Класс-обертка |
|--------|---------------|
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| char | Character |
| float | Float |
| double | Double |

Литералы

Литералы — это явно заданные значения одного из допустимых типов, которые могут использоваться в различных выражениях языка (например, как аргумент метода, выводящего что-либо на экран).

Примеры литералов:

2 - литерал типа int

1.2 - литерал типа double

2e5 - литерал типа double, записанный в научной нотации, число после e надо воспринимать как показатель степени десятки, на которую умножается число указанное до e. В данном случае записано число $2 \cdot 10^5 = 200000$

2L - литерал типа long, можно использовать строчную букву l

2F - литерал типа float, можно использовать строчную букву f

2D - литерал типа double, можно использовать строчную букву d

2e-2f - литерал типа float, численно равен 0.02

false - литерал типа boolean

'a' - литерал типа char — печатный символ, задаётся в одинарных кавычках

Константы

Ключевое слово final означает, что присвоить какое-либо значение данной переменной можно лишь один раз и навсегда. Использовать в именах констант только прописные буквы или только строчные необязательно.

3. Операторы

3.1 Понятие оператор

Операторы в языке Java — это специальные символы, которые сообщают транслятору о том, что вы хотите выполнить операцию с некоторыми операндами.

3.2 Типы операторов

Операторы можно классифицировать по нескольким показателям. Рассмотрим типы операторов по количеству операндов. Некоторые операторы требуют одного операнда, их называют унарными. Одни операторы ставятся перед операндами и называются префиксными, другие — после, их называют постфиксными операторами. Большинство же операторов ставят между двумя операндами, такие операторы называются infixными бинарными операторами. Существует тернарный оператор, работающий с тремя операндами.

С другой стороны, операторы можно разделить по таким критериям, как: типы данных операндов и возвращаемое значение оператора. Можно выделить: арифметические, логические и операторы ветвлений.

3.3 Оператор присваивания

Операторы присваивания устанавливают значение переменной или выражения в новое значение. Присваивание поддерживается рядом других операторов. Простое присваивание использует знак «`=`».

3.4 Арифметические операторы

Арифметические операторы используются для вычислений так же как в алгебре. Допустимые операнды должны иметь числовые типы. Использовать арифметические операторы для работы с логическими типами нельзя, а для работы с типом `char` можно, поскольку в Java тип `char` — это подмножество типа `int`.

Существуют, также, и сокращенные формы записи арифметических операций совместно и оператором присваивания.

| Оператор | Результат | Оператор (с присваиванием) |
|----------|---------------------------------|----------------------------|
| + | Сложение | + = |
| - | Вычитание (также унарный минус) | - = |
| * | Умножение | * = |
| / | Деление | / = |
| % | Деление по модулю | % = |
| ++ | Инкремент | -- |