



Язык программирования Java. Начальный уровень

Мультимедийный курс

автор: Васильев А.Н.

www.vasilev.kiev.ua

Киев 2017



Лекция 2. Базовые типы данных и основные операторы



- Базовые типы данных
- Литералы и автоматическое приведение типов
- Арифметические операторы
- Логические операторы
- Операторы сравнения
- Побитовые операторы
- Оператор присваивания

*Показывай свою гравицанпу.
Если фирменная вещь -
возьмём!
из к/ф "Кин-дза-дза"*

Базовые типы Java



Целые числа

byte	8 бит
short	16 бит
int	32 бит
long	64 бит

Логический тип

boolean	значения true и false
---------	--------------------------

Числа с плавающей точкой

float	32 бит
double	64 бит

Литералы

125	int
125L	long
0b101	2-ичные
012	8-ричное
0x12	16-ричное
2.0	double
2.0F	float

Символы

char	16 бит
------	--------

Используем

int double char boolean

Переменные



Объявление:

- идентификатор типа и имя переменной;
- объявляется в любом месте, но до первого использования;
- область доступности определяется блоком, в котором объявлена переменная (блок выделяется скобками { и }).

Примеры объявлений:

```
int num;  
int a=123,b,c=321;  
int d=a+c;  
char symb='Q',s;  
double x,y=12,z=12.0;  
boolean A,B=true;
```

Автоматическое приведение типов

- Типы должны быть совместимы
- Целевой тип "больше" исходного

Основные правила:

- Типы `byte` и `short` преобразуются в `int`
- Если есть `long` - все преобразуется в `long`
- Если есть `float` - все преобразуется в `float`
- Если есть `double` - все преобразуется в `double`

Переменные: примеры объявлений



ОШИБКА!!!

```
byte a=50;  
a=a+1;
```



Литерал 1 имеет тип `int`

ОШИБКА!!!

```
byte a=5,b=1,c;  
c=a+b;
```



Сумма `byte`-переменных
вычисляется как `int`-значение

Правильно!!!

```
byte a=50;  
a=(byte) (a+1);
```



Явное приведение типа

Правильно!!!

```
byte a=5,b=1,c;  
c=(byte) (a+b);
```



Явное приведение
типа



Операторы в Java



Основные группы операторов:

- Арифметические;
- Логические;
- Операторы сравнения;
- Побитовые.

А также:

- Оператор присваивания;
- Сокращенные формы операторов;
- Тернарный оператор.

Арифметические операторы



Оператор	Действие
+	сложение
-	вычитание
*	умножение
/	деление
%	остаток от деления
++	инкремент
--	декремент

Для целочисленных операндов выполняется целочисленное деление: результат выражения $8/5$ есть 1. Для обычного деления используют команду `(double)8/5` (результат равен 1.6)

При выполнении команды вида `a++` (или `++a`) переменная `a` увеличивает значение на 1. При выполнении команды вида `a--` (или `--a`) переменная `a` уменьшает значение на 1

```
int a=10,b;  
b=a++;  
// a=11, b=10
```

```
int a=10,b;  
b=++a;  
// a=11, b=11
```


Логические операторы



Оператор	Действие
&	логическое и
&&	логическое и (сокращенная форма)
	логическое или
	логическое или (сокращенная форма)
^	исключающее или
!	логическое отрицание

Логическое и: команда вида A&B

A \ B	true	false
true	true	false
false	false	false

Логическое или: команда вида A|B

A \ B	true	false
true	true	true
false	true	false

Исключающее или: команда A^B

A \ B	true	false
true	false	true
false	true	false

Операторы сравнения



Оператор	Действие
<	меньше
<=	меньше или равно
>	больше
>=	больше или равно
==	равно
!=	не равно

Системы счисления



- Количество используемых цифр определяет систему счисления

- Если используется h цифр, то в числе $\overline{b_m b_{m-1} \dots b_2 b_1 b_0}$ параметры $b_k = 0, 1, 2, \dots, h - 1$

- В десятичной системе

$$\overline{b_m b_{m-1} \dots b_2 b_1 b_0} = b_0 \cdot h^0 + b_1 \cdot h^1 + b_2 \cdot h^2 + \dots + b_m \cdot h^m$$

- Для двоичной системы $h = 2$

$$\overline{b_m b_{m-1} \dots b_2 b_1 b_0} = b_0 \cdot 2^0 + b_1 \cdot 2^1 + b_2 \cdot 2^2 + \dots + b_m \cdot 2^m$$

параметры $b_k = 0, 1$

- Двоичный код 1101 соответствует числу

$$1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13$$

- Для восьмеричной системы $h = 8$

$$\overline{b_m b_{m-1} \dots b_2 b_1 b_0} = b_0 \cdot 8^0 + b_1 \cdot 8^1 + b_2 \cdot 8^2 + \dots + b_m \cdot 8^m$$

параметры $b_k = 0, 1, 2, \dots, 7$

- Восьмеричный код 123 соответствует числу

$$1 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0 = 64 + 16 + 3 = 83$$



Шестнадцатеричная система



- В шестнадцатеричной системе ($h = 16$) кроме десяти цифр от 0 до 9 еще используются буквы от A (обозначает число 10) до F (обозначает число 15)

- Перевод в десятичную систему выполняется в соответствии с соотношением

$$\overline{b_m b_{m-1} \dots b_2 b_1 b_0} = b_0 \cdot 16^0 + b_1 \cdot 16^1 + b_2 \cdot 16^2 + \dots + b_m \cdot 16^m$$

параметры $b_k = 0, 1, 2, \dots, 15$

- Например, шестнадцатеричный код A1B соответствует числу

$$10 \cdot 16^2 + 1 \cdot 16^1 + 11 \cdot 16^0 = 2560 + 16 + 11 = 2587$$

(в шестнадцатеричном представлении символ A соответствует числу 10, а символ B соответствует числу 11)

Двоичное кодирование



- В двоичном представлении число записывается в виде последовательности нулей и единиц.

- Используется n битов (значение 0 или 1).

- Бинарный код числа $\overline{a_{n-1}a_{n-2} \dots a_2a_1a_0}$.

- В десятичной системе

$$a_0 \cdot 2^0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + \dots + a_{n-2} \cdot 2^{n-2} + a_{n-1} \cdot 2^{n-1}$$

- Операции с числами в двоичной системе:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 10$$

Отрицательные числа



- Имеется некоторое положительное число x
- Необходимо сгенерировать код для числа $-x$
- По определению $-x + x = 0$
- К числу x применяем побитовую инверсию: единицы поменяем на нули, а нули поменяем на единицы. Результат обозначим как $\sim x$
- Результатом выражения $\sim x + x$ является код $\underbrace{111 \dots 11}_n$ из n единиц
- Прибавим число 1 (вычисляем сумму $\sim x + x + 1$): получим код $1 \underbrace{000 \dots 00}_n$, в котором n нулей и в старшем бите 1
- Компьютер запоминает n битов и старший единичный бит теряется
- Получается код из n нулей: имеет место соотношение $\sim x + x + 1 = 0$
- Выражение $\sim x + 1$ дает код отрицательного числа $-x$

Отрицательные числа



- В двоичном побитовом представлении числа старший бит называется знаковым и определяет знак числа: у положительных чисел старший бит нулевой, а у отрицательных чисел старший бит единичный.
- Чтобы по коду положительного числа получить код противоположного ему отрицательного числа, необходимо инвертировать код положительного числа и прибавить к результату единицу.
- Чтобы по коду отрицательного числа определить это число в десятичной системе счисления, следует выполнить инверсию кода отрицательного числа, прибавить к коду единицу, перевести полученный код в десятичное значение (как это делается для положительных чисел) и "дописать" перед числом знак "минус".

Побитовые операторы



Оператор	Действие
&	побитовое и
	побитовое или
^	побитовое исключающее или
~	побитовая инверсия
<<	сдвиг влево
>>	сдвиг вправо
>>>	сдвиг вправо с заполнением старшего бита нулями

Побитовые операторы



Побитовое и: $a \& b$

$a \backslash b$	1	0
1	1	0
0	0	0

Побитовое или: $a | b$

$a \backslash b$	1	0
1	1	1
0	1	0

Исключающее или: $a \wedge b$

$a \backslash b$	1	0
1	0	1
0	1	0

Инверсия: $\sim a$

a	0	1
$\sim a$	1	0

Побитовые операторы



Сдвиг влево <<

`A << n`



Побитовое представление переменной **A** сдвигается влево на **n** позиций. Старшие биты теряются, младшие заполняются нулями

Сдвиг вправо >>

`A >> n`



Побитовое представление переменной **A** сдвигается вправо на **n** позиций. Младшие биты теряются, старшие заполняются значением знакового бита

Сдвиг вправо >>>

`A >>> n`



Побитовое представление переменной **A** сдвигается вправо на **n** позиций. Младшие биты теряются, старшие заполняются нулями

Оператор присваивания



Переменная = Значение

Оператор присваивания возвращает значение!!!

Следующие команды имеют смысл:

```
int x,y,z;  
x=y=z=100;
```



Все три переменные получают значение 100

Переменная x получает значение 16, переменная y получает значение 4, а переменная z получает значение 6.



```
int x,y,z;  
x=(x=(y=3)-(z=7))+(x=x+(y=y+1)*(z=z-1));
```

Сокращенные формы операции присваивания



Команда вида

$$A = A \text{ ☺ } B$$

может быть записана как

$$A \text{ ☺ } = B$$

Здесь ☺ – бинарный арифметический или побитовый оператор

Например

$A = A + B$



$A += B$

$A = A * B$



$A *= B$

$A = A ^ B$



$A ^= B$

Тернарный оператор



условие?значение_1:значение_2

Логическое
значение

Это значение
возвращается,
если условие
истинно

Это значение
возвращается,
если условие
ложно

```
num = (x < 0) ? 0 : 1;
```

Переменная `num` получает значение 0
если `x` меньше 0, и 1 в противном случае

Приоритет операторов



1. () [] .

2. ++ -- ~ !

3. * / %

4. + -

5. >> << >>>

6. > >= <= <

7. == !=

8. &

9. ^

10. |

11. &&

12. ||

13. ?:

14. = op=

Проверка числа на четность/нечетность



```
import static javax.swing.JOptionPane.*;
class OddEvenDemo{
    public static void main(String[] args){
        int number,remainder;
        number=Integer.parseInt(
            showInputDialog("Введите целое число:")
        );
        // Остаток от деления на 2:
        remainder=number%2;
        String txt="Вы ввели ";
        // Использован тернарный оператор:
        txt+=(remainder==0?"четное":"нечетное")+" число!";
        showMessageDialog(null,txt);
    }
}
```


Проверка числа: результат выполнения



Input dialog box titled "Input" with a question mark icon. The text "Введите целое число:" is followed by a text field containing "124". Below the text field are "OK" and "Cancel" buttons. A mouse cursor is pointing at the "OK" button.



Message dialog box titled "Message" with an information icon. The text "Вы ввели четное число!" is displayed. Below the text is an "OK" button.

Input dialog box titled "Input" with a question mark icon. The text "Введите целое число:" is followed by a text field containing "123". Below the text field are "OK" and "Cancel" buttons. A mouse cursor is pointing at the "OK" button.



Message dialog box titled "Message" with an information icon. The text "Вы ввели нечетное число!" is displayed. Below the text is an "OK" button.

Количество сотен в числе



```
import static javax.swing.JOptionPane.*;
class HundredsDemo{
    public static void main(String[] args){
        int number,hundreds;
        number=Integer.parseInt(
            showInputDialog("Введите целое число:")
        );
        // Количество сотен в числе (для целочисленных
        // операндов деление выполняется нацело):
        hundreds=( (number/10)/10)%10;
        String txt="В этом числе "+hundreds+" сотен!";
        showMessageDialog(null,txt);
    }
}
```

Количество сотен: результат выполнения



Input dialog box with a question mark icon. Text: "Введите целое число:". Input field contains "123567". Buttons: "OK", "Cancel".



Message dialog box with an information icon. Text: "В этом числе 5 сотен!". Button: "OK".

Input dialog box with a question mark icon. Text: "Введите целое число:". Input field contains "12". Buttons: "OK", "Cancel".



Message dialog box with an information icon. Text: "В этом числе 0 сотен!". Button: "OK".

Домашнее задание



- Определите число по двоичному коду 1010110.
- Определите двоичный код числа 15.
- Определите число по шестнадцатеричному коду A03D.
- Сложите два двоичных числа 1011 и 1101. Что это за числа и какое получилось число?
- Напишите программу, которая проверяет, делится ли введенное пользователем число на 3.
- Напишите программу, которая проверяет, сколько тысяч во введенном пользователем числе.