



- ❖ Типы данных
- ❖ Арифметические операции
- ❖ Условные выражения
- ❖ Условные конструкции



Типы данных

Java

Одной из основных особенностей Java является то, что данный язык является строго типизированным. А это значит, что каждая переменная и константа представляет определенный тип и данный тип строго определен. Тип данных определяет диапазон значений, которые может хранить переменная или константа.



Типы данных в Java

- ❖ boolean хранит значение true или false
- ❖ byte хранит целое число от -128 до 127 и занимает 1 байт
- ❖ short хранит целое число от -32768 до 32767 и занимает 2 байта
- ❖ int хранит целое число от -2147483648 до 2147483647 и занимает 4 байта
- ❖ long хранит целое число от $-9\ 223\ 372\ 036\ 854\ 775\ 808$ до $9\ 223\ 372\ 036\ 854\ 775\ 807$ и занимает 8 байт
- ❖ double хранит число с плавающей точкой от $\pm 4.9 \cdot 10^{-324}$ до $\pm 1.8 \cdot 10^{308}$ и занимает 8 байт
- ❖ float хранит число с плавающей точкой от $-3.4 \cdot 10^{38}$ до $3.4 \cdot 10^{38}$ и занимает 4 байта
- ❖ char хранит одиночный символ в кодировке UTF-16 и занимает 2 байта, поэтому диапазон хранимых значений от 0 до 65535
- ❖ String



Арифметические операции в Java

Операции в Java	Описание	Пример
+	операция сложения двух чисел	<code>int d = 4 + 7;</code>
-	операция вычитания двух чисел	<code>int d = 4 - 7;</code>
*	операция умножения двух чисел	<code>int d = 4 * 7;</code>
/	операция деления двух чисел	<code>int d = 7 / 4; // 2</code>
%	получение остатка от деления двух чисел	<code>int d = 22 % 4;</code> <code>// 2 (22 - 4*5 = 2)</code>



Практические задания

```
public static void main(String[] args) {  
  
    int x1 = 100;  
    int x2 = 65;  
  
    double y1 = n1 + n2 + n3 + n4 + x1;  
    double y1 = n1 + n2 + n3 + n4 + x2;  
  
    System.out.println("symm = " + symm);  
    System.out.println("min = " + min);  
    System.out.println("avg = " + avg);  
}
```



Унарные арифметические операции

Операции в Java	Пример
++ Префиксный инкремент	<pre>int a = 8; int b = ++a; System.out.println(a); // 9 System.out.println(b); // 9</pre>
++ Постфиксный инкремент	<pre>int a = 8; int b = a++; System.out.println(a); // 9 System.out.println(b); // 8</pre>
-- Префиксный декремент	<pre>int a = 8; int b = --a; System.out.println(a); // 7 System.out.println(b); // 7</pre>
-- Постфиксный декремент	<pre>int a = 8; int b = a--; System.out.println(a); // 7 System.out.println(b); // 8</pre>



Приоритет арифметических операций

++ (инкремент), -- (декремент)

* (умножение), / (деление), % (остаток от деления)

+ (сложение), - (вычитание)

```
int a = 8;  
int b = 7;
```

```
x = a + b * ++b;  
x = (a + b) * b++;
```

```
int a = 9;  
int b = 3;
```

```
x = a / b + ++b;  
x = a + (b - --b);
```

```
int a = 10;  
int b = 5;
```

```
x = a2 * b * b++;  
x = (a2 + b) * a++;
```



Практические задания

```
public static void main(String[] args) {  
  
    int n1 = 100;  
    int n2 = 65;  
    double n3 = 48.0;  
    double n4 = 27.0;  
  
    double symm = n1 + n2 + n3 + n4;  
    min = ...  
  
    System.out.println("symm = " + symm);  
    System.out.println("min = " + min);  
    System.out.println("avg = " + avg);  
}
```




Условные выражения

Операции в Java	Пример
==	<pre>boolean c = 3 == 4; // false boolean c = 3 == 3; // true</pre>
!=	<pre>boolean c = 3 != 3; // false boolean c = 3 != 4; // true</pre>
<	<pre>boolean c = 3 < 3; // false boolean c = 3 < 4; // true</pre>
>	<pre>boolean c = 3 > 3; // false boolean c = 4 > 3; // true</pre>
<=	<pre>boolean c = 4 <= 3; // false boolean c = 3 <= 4; // true</pre>
>=	<pre>boolean c = 3 >= 4; // false boolean c = 3 >= 3; // true</pre>



Логические операции

Оператор	Описание	Пример
&& (логическое и)	Вызывается логическим оператором И. Если оба операнда не равны нулю, условие становится истинным.	(A && B) ложно
(логическое или)	Вызывается логическим оператором ИЛИ. Если любой из двух операндов отличен от нуля, условие становится истинным.	(A B) верно
! (логическое нет)	Вызывается оператором логического НЕ. Используется для изменения логического состояния операнда на обратное. Если условие истинно, то оператор логического НЕ сделает ложным.	!(A && B) верно



Практика

```
int a = 8;
```

```
int b = 7;
```

```
boolean check = (a > 10) && (b < 5);
```

```
boolean check = (a != 10) || (b < 5);
```

```
boolean check = (a != 8) || (b < 5);
```

```
boolean check = (a >= 8) && (b <= 5);
```

```
boolean check = (a <= 10) && (b <= 7);
```

```
int a = 10;
```

```
int b = 20;
```

```
boolean check = (a > 10) && (b < 5);
```

```
boolean check = (a != 10) || (b < 5);
```

```
boolean check = (a != 8) || (b < 5);
```

```
boolean check = (a >= 8) && (b <= 5);
```

```
boolean check = (a <= 10) && (b <= 7);
```



Условные конструкции

Конструкция if/else

```
int num1 = 6;
int num2 = 4;
if(num1>num2){
    System.out.println("Первое число больше второго");
}
```

```
int num1 = 6;
int num2 = 8;
if(num1>num2){
    System.out.println("Первое число больше второго");
}
else if(num1<num2 && num1 != num2){
    System.out.println("Первое число меньше второго");
}
else{
    System.out.println("Числа равны");
}
```



Условные конструкции

Конструкция switch

```
int num = 8;
switch (num) {
    case 1:
        System.out.println("число равно 1");
        break;
    case 8:
        System.out.println("число равно 8");
        num++;
        break;
    case 9:
        System.out.println("число равно 9");
        break;
    default:
        System.out.println("число не равно 1, 8, 9");
}
```



Условные конструкции

Тернарная операция

Значение

=

Первый операнд

?

Второй операнд

:

Третий операнд

```
int x=3;  
int y=2;  
int z = x<y? (x+y) : (x-y);  
System.out.println(z);
```



Практика

```
public static void main(String[] args) {  
  
    int n1 = 100;  
    int n2 = 65;  
    double n3 = 48.0;  
    double n4 = 27.0;  
  
    double symm = n1 + n2 + n3 + n4;  
    min = ...  
  
    System.out.println("symm = " + symm);  
    System.out.println("min = " + min);  
    System.out.println("avg = " + avg);  
}
```



Циклы

Цикл for

```
for ([инициализация счетчика]; [условие]; [изменение счетчика])  
{  
    // действия  
}
```

```
for (int i = 1; i < 9; i++){  
    System.out.printf("Квадрат числа %d равен %d \n", i, i * i);  
}
```




ЦИКЛЫ

Цикл do

```
int j = 7;
do{
    System.out.println(j);
    j--;
}
while (j > 0);
```

Цикл while

```
int j = 6;
while (j > 0){
    System.out.println(j);
    j--;
}
```



Подведём итоги



Java – высокоуровневый компилируемый язык программирования



Java компилируется в байт-код, который исполняется JVM в любой среде