

Конспект лекций по java

Александр Титилин

Содержание

1 Шапочка	2
2 Hello World	2
3 Целочисленный тип данных int	2
4 Ввод с клавиатуры	3
4.1 Получение целого числа с клавиатуры.	3
5 Действия с целыми числами	3
5.1 Замечание. Деление отрицательных чисел	3
6 Целочисленные типы данных в java	4
7 Ерунда про двоичное представление, про знак и прочее	4
8 15.09.2022	4
8.1 Вещественный тип данных	4
8.2 Ввод	4
8.3 Вывод на экран	4
8.4 Операции с вещественными числами	5
8.5 Приведения действительных чисел к целым	5
9 Математические функции в java.	5
9.1 Символьный тип данных char	5
9.2 Ввод с клавиатуры	5
10 22.09.2022	6
10.1 Логический тип данных	6
10.2 Операции сравнения	6
10.3 Логические операции	6
10.4 Условный оператор	6
10.5 Область видимости переменных.	7

11 Цикл с счетчиком <u>for</u>	7
11.1 Задача	7
11.1.1 Первый способ, формулой	7
11.1.2 Второй способ, дополнительная переменная	8
11.2 Задача	8
11.3 Правило	8

1 Шапочка

```

1      public class Main {
2          public static void main(String[] args){
3              write here
4          }
5      }

```

2 Hello World

```

1      System.out.print("Hello World");
2      System.out.println();
3      System.out.print("How are you?");

```

print - вывести в консоль без переноса строки.
println - вывести в консоль с переносом строки.

3 Целочисленный тип данных int

Создание переменной с именем variableName с типом dataType

```

1      dataType variableName;

```

Переменная sum с типом данных int;

```

1      int sum;

```

Положим в sum число 12.

```

1      sum = 12;

```

Создаем переменную sum и сразу присваиваем ей значение 12

```

1      int sum = 12;

```

Описываем несколько переменных и некоторые инициализируем

```

1      int a , b , c = 5, d , e = 1234;

```

Вывод значение переменной sum на экран.

```

1      System.out.println(sum);

```

Выводить можем и значения выражений.

```

1      System.out.println(sum + 129291);

```

```
1 System.out.println("next after sum" + sum + 1);
```

Напишет "... 121"

```
1 System.out.println("next after sum" + ( sum + 1 ));
```

Напишет "... 13"

int - целое число, размером 4 байта. Имена переменных начинаются с маленькой буквы, состоящие из нескольких слов используют camelCase (countOfEvenDigits).

4 Ввод с клавиатуры

Для ввода с клавиатуры нужен объект Scanner из шапки с сайта школы.

4.1 Получение целого числа с клавиатуры.

```
1 sum = in.nextInt();
```

Так тоже можно

```
1 int second = in.nextInt();
```

5 Действия с целыми числами

Арифметические действия как везде (*, +, -, /, %), деление целочисленное. Побитовый сдвиг влево

```
1 a << b;
```

Побитовый сдвиг вправо.

```
1 a >> b;
```

Знаковый побитовый сдвиг вправо.

```
1 a >>> b;
```

5.1 Замечание. Деление отрицательных чисел

Делимое	Делитель	Целое	Остаток
23	5	4	3
-23	5	-4	-3
23	-5	4	3
-23	-5	4	-3

В питоне и математике не так.

6 Целочисленные типы данных в java

Тип Данных	Размер ячейки	Размер в битах	Диапазон
byte	1 байт	8	$-128 \dots 127$
short	2 байта	16	$-32768 \dots + 32767$
int	4 байта	32	$-2^{31} + \dots 2^{31} - 1$
long	8 байт	64	$-2^{63} + \dots + 2^{63} - 1$

7 Ерунда про двоичное представление, про знак и прочее

Суть в том, что старший разряд в двоичном представлении это знак. Про дополнительный двоичный код. Пока джавы нет идут байки про работу компа.

8 15.09.2022

8.1 Вещественный тип данных

Вещественные типы данных в java - float(4 байта) и double(8 байт). Если можно не использовать вещественные числа, то их не надо использовать. Прикол про $0.1 + 0.2 \neq 0.3$. Любые числа с вещественными числами будут приближенными.

```
1 float f = 1.7;  
2 double d = 1.7;
```

Первое работать не будет.

8.2 Ввод

```
1 double x = in.nextDouble();
```

8.3 Вывод на экран

```
1 out.printf("%.3f", x);
```

Вывод вещественного числа x с 3 числами после запятой.

```
1 out.printf("Answer: %d %.2f\n", a, x);
```

Вывод слова, целого числа a в десятичном представлении и вещественного числа x с двумя числами после запятой и перевод строки.

8.4 Операции с вещественными числами

Арифметические как в целых, кроме деления.

```
1 int b = 23, c=5;  
2 double y = b/c;
```

Y будет равен четырем. Надо делать так

```
1 double y = (double)b / c;
```

Надо какое нибудь число привести к вещественным.

8.5 Приведения действительных чисел к целым

```
1 int g = (int) x;
```

9 Математические функции в java.

Все такие функции лежат в библиотеке Math, подключать не надо.

Math.abs(x)	$ x $
Math.sqrt(x)	\sqrt{x}
Math.sin(x)	$\sin x$
Math.cos(x)	$\cos x$
Math.tan(x)	$\tan x$

9.1 Символьный тип данных char

```
1 char c = 'F';
```

Представляет собой целое беззнаковое число, занимает 2 байта.

9.2 Ввод с клавиатуры

```
1 char h = (char) System.in.read();
```

При этом компилятор ругнется. Нужно использовать альтернативную шапку

```
1 import java.io.IOException  
2 public class Main {  
3     public static void main(String[] args) throws  
4         IOException  
5     {  
6         write here  
7     }  
}
```

10 22.09.2022

10.1 Логический тип данных

```
1 boolean b = true;
```

10.2 Операции сравнения

Математика	Java
>	>
<	<
=	==
≥	>=
≤	<=
≠	!=

```
1 boolean c = 3 > 5;  
2 out.print(c);
```

Считывать boolean нельзя.

10.3 Логические операции

Операция	Обозначение	Смысл
Не (инверсия)	!	Меняет логическое значение на противоположное
И (конъюнкция)	&&	Истина, если оба операнда истина
Или (дизъюнкция)		Истина, если хотя бы один операнд истина
Хор	^	Истинна если операнды разные

В java нельзя использовать двойные сравнения.

10.4 Условный оператор

```
1 if (cond){  
2     operator-Yes;  
3 }
```

```
1 if (cond){  
2     operator-Yes;  
3 }  
4 else {  
5     operator-No;  
6 }
```

Примеры.

```
1 if (x > 10) {  
2     System.out.println("Too much");  
3 }
```

```

1  if (x > 10){
2      System.out.println("Too much");
3  }
4  else {
5      System.out.println("Good");
6  }

```

10.5 Область видимости переменных.

Создание переменной внутри блока

```

1  {
2      int a = 10;
3  }

```

Внутри скобок использовать можно, снаружи нет.

11 Цикл с счетчиком for

Цикл повторяющаяся последовательность действий, которые называются телом цикла.

```

1  for(start values; condition;change counter) {
2      body
3  }

```

В блоке начальных значений, можно описывать переменные. В блоке изменения счетчика можно менять несколько начальных значений.

```

1  for(int i = 0,k,j = ; i < 10 && j < 1000;i++,j+10)

```

Эти переменные пропадут, после окончания цикла.

```

1  for(int i = 5; i <= 8; i ++ )

```

Выведет

```

1  5
2  6
3  7
4  8

```

11.1 Задача

Дано натуральное число, нужно вывести первые n четных чисел.

11.1.1 Первый способ, формулой

```

1  for(int i = 0; i < n ; i++){
2      out.println(2*(i + 1));
3  }

```

11.1.2 Второй способ, дополнительная переменная

```
1 int a = 2
2 for(int i = 0; i < n; i++){
3     out.println(a);
4     a += 2;
5 }
```

Тоже самое, но короче

```
1 for (int i = 0, a = 2; i < n; i++, a+=2) {
2     out.println(a);
3 }
```

11.2 Задача

Дано число n (количество элементов последовательности), после этого даны n чисел. Надо найти \sum четных элементов данной последовательности

```
1 int n = in.nextInt();
2 int sum = 0;
3 for (int i = 0, a; i < n ; i++){
4     a = in.nextInt();
5     if (a % 2 == 0){
6         sum += a;
7     }
8 }
9 out.println(sum);
```

11.3 Правило

Начальные значения переменных (суммы, количества) нужно задавать непосредственно перед тем циклом, в котором они изменяются.