

# ОСНОВЫ СИНТАКСИСА: СТРУКТУРА ПРОГРАММЫ И УСЛОВНЫЕ ОПЕРАТОРЫ



**ИЛЬДАР ГАЛИМБИКОВ**



# ИЛЬДАР ГАЛИМБИКОВ

Android developer в Atlassian



[grenni2007@gmail.com](mailto:grenni2007@gmail.com)



[@grenni2007](https://t.me/grenni2007)



# ПЛАН ЗАНЯТИЯ

1. [О Java](#)
2. [Структура программы](#)
3. [Понятие переменной](#)
4. [Вывод данных в консоль](#)
5. [Понятие пакета](#)
6. [Считывание значения из консоли](#)
7. [Условные операторы](#)
8. [Пример деления чисел](#)



**0 JAVA**

---

# О ЯЗЫКЕ JAVA

Язык [Java](#) был создан канадским ученым [Джеймсом Гослингом](#), дата официального выпуска — 23 мая 1995 года).





# ХАРАКТЕРИСТИКИ ЯЗЫКА

- Объектно ориентированный;
- Статическая типизация;
- Платформо-независимый (байт-код);
- Работает на виртуальной машине JVM;
- Популярный язык — много доступной информации и готовых компонентов;
- Удобные и бесплатные инструменты разработки.



# СТРУКТУРА ПРОГРАММЫ

# СТРУКТУРА ПРОСТЕЙШЕЙ ПРОГРАММЫ

```
1  class Main { //Класс
2      public static void main(String[] args) { //Метод
3          System.out.println("Hello world!"); //Инструкция
4      }
5  }
```



# ПРОБЕЛЫ В ПРОГРАММЕ

- Обязательно нужно ставить пробелы между ключевыми словами и названиями переменных.
- Между буквой и скобкой (или знаком `=`) — не обязательно.
- Для улучшения читаемости программы блоки (код внутри фигурных скобок) выделяют отступами слева.
- Традиционно используют отступ длиной 4 пробела (или 1 таб).

\* Не все правила, перечисленные на этом и следующих слайдах, соблюдаются в коде на этой презентации. Это сделано в основном для удобства чтения кода.

# КЛЮЧЕВЫЕ СЛОВА ЯЗЫКА

Ключевые слова еще также называют зарезервированными. Их нельзя использовать как имя переменной, класса или метода. Вот их список:

- `abstract`, `continue`,  
`for`, `new`, `switch`;
- `assert`, `default`, `goto`\*,  
`package`, `synchronized`;
- `boolean`, `do`, `if`,  
`private`, `this`;
- `break`, `double`,  
`implements`, `protected`,  
`throw`;
- `byte`, `else`, `import`,  
`public`, `throws`;
- `case`, `enum`, `instanceof`,  
`return`, `transient`;
- `catch`, `extends`, `int`,  
`short`, `try`;
- `char`, `final`, `interface`,  
`static`, `void`;
- `class`, `finally`, `long`,  
`strictfp`, `volatile`;
- `const`\*, `float`, `native`,  
`super`, `while`.



# ПОНЯТИЕ ПЕРЕМЕННОЙ

# ПЕРЕМЕННЫЕ

— Объявление

```
int x;
```

— Присваивание

```
x = 3;
```

— Инициализация

```
int x = 3;
```


Основное назначение переменных — хранить значения, которые могут меняться в процессе работы программы.

# ИМЕНА ПЕРЕМЕННЫХ, МЕТОДОВ И Т.Д.

- Все ключевые слова в Java пишутся с маленькой буквы.
- Классы — с большой.
- Названия методов и идентификаторы переменных — camel case (например, `doSomething()` или `inputValue`).
- Идентификаторы и названия всегда должны начинаться с буквы.
- Если в вашей программе больше одной переменной или метода, то лучше выбирать осмысленные имена (а не `a`, `b` или `peremennaya1`).

# РАССМОТРИМ НА ПРИМЕРЕ

```
1  class Main {  
2      public static void main(String[] args) {  
3          String s = "Hello world!";    //Инициализация  
4          System.out.println(s);        //Выведет Hello world!  
5          s = "Bye!";  
6          System.out.println(s);        //Выведет Bye!  
7      }  
8  }
```



# **ВЫВОД ДАННЫХ В КОНСОЛЬ**

# ВЫВОД ДАННЫХ В КОНСОЛЬ

Для того чтобы вывести данные в консоль нужно использовать вызов следующего метода:

- Способ без перевода каретки на новую строку:

```
System.out.print("Всем привет");
```

- Способ с переводом каретки на новую строку:

```
System.out.println("Всем снова привет");
```

- Вывод не только текста, но и значения переменной (в кавычках можно писать любой текст):

```
int value = 100;  
System.out.println("Значение переменной value = " + value);
```



# ФОРМАТИРОВАННЫЙ ВЫВОД

```
1 System.out.printf("Целое число %d", 123);  
2 System.out.printf("Дробное число %f", 123.5);  
3 String s = "абв";  
4 System.out.printf("Строка и перевод на новую строку %s%n", s);
```



# ПОНЯТИЕ ПАКЕТА

# ПАКЕТЫ

Файлы с исходным кодом в Java могут храниться внутри пакетов (но это не обязательно). По сути пакет — это просто папка, в которой лежат java-файлы. Пакеты нужны для того, чтобы структурировать программу и упростить поиск нужных java-файлов.

Также пакеты можно импортировать внутрь класса, чтобы использовать классы, лежащие в этом пакете в своем коде.

Примеры пакетов:

- `java.util`
- `apache.commons.collection`

# ПАКЕТЫ

Чтобы указать, какому пакету принадлежит класс, используем:

```
package ru.netology.code;
```

Чтобы использовать класс из пакета, например `Scanner`, пишем:

```
import java.util.Scanner;
```

Сделав это вы сможете использовать `Scanner` в том классе, где использован `import`.



# **СЧИТЫВАНИЕ ЗНАЧЕНИЯ ИЗ КОНСОЛИ**

# СЧИТЫВАНИЯ ЗНАЧЕНИЯ ИЗ КОНСОЛИ

Для считывания данных из консоли используется специальный объект

Scanner:

```
1  import java.util.Scanner;
2
3  class Main {
4      public static void main(String[] args) {
5          System.out.println("Как вас зовут?");
6          Scanner scan = new Scanner(System.in);
7          String s = scan.nextLine();
8          System.out.println("Привет, " + s + "!");
9      }
10 }
```

# СЧИТЫВАЕМ ЧИСЛОВЫЕ ЗНАЧЕНИЯ ИЗ КОНСОЛИ

Для считывания данных из консоли в числовом виде, используется вызов метода `nextInt()`:

```
1  import java.util.Scanner;
2
3  class Main {
4      public static void main(String[] args) {
5          System.out.println("Введите 2 числа:");
6          Scanner scan = new Scanner(System.in);
7          int a = scan.nextInt();
8          int b = scan.nextInt();
9          int c = a + b;
10         System.out.printf("a + b = %d", c);
11     }
12 }
```

# ПРЕВРАЩАЕМ СТРОКУ В ЧИСЛО

Даже если введенная пользователем строка содержит только цифры, с ней нельзя производить математические операции. Сначала нужно привести ее к числовому типу. Для этого используется `Integer.parseInt()` или аналогичные методы других типов:

```
1  import java.util.Scanner;
2
3  class Main {
4      public static void main(String[] args) {
5          System.out.println("Введите число:");
6          Scanner scan = new Scanner(System.in);
7          String s = scan.nextLine();
8          int a = Integer.parseInt(s);
9          System.out.println(a + 1);
10     }
11 }
```





# УСЛОВНЫЕ ОПЕРАТОРЫ

# УСЛОВНЫЕ ОПЕРАТОРЫ

Оператор (или инструкция) — это наименьшая часть языка программирования. Обозначает какое-то действие, которое необходимо выполнить. Например:

- Оператор присваивания `=` (уже видели в примерах);
- Условный оператор `if ... then ...else` (увидим в следующем примере).

# УСЛОВНЫЙ ОПЕРАТОР[1]

Например, нам нужно узнать что пользователь ввел число 0:

```
1 //какой-то код
2 int a = scan.nextInt();
3 if(a==0){
4     System.out.println("Введено число 0");
5 }
6 //какой-то код
```

Если был введен 0, на экран будет выведено соответствующее сообщение.

## УСЛОВНЫЙ ОПЕРАТОР[2]

Чтобы вывести сообщение, используем ключевое слово `else` в обоих случаях:

```
1 //какой-то код
2 int a = scan.nextInt();
3 if(a==0){
4     System.out.println("Введено число 0");
5 } else {
6     System.out.println("Введено число НЕ 0");
7 }
8 //какой-то код
```

Теперь сообщение будет выведено в любом случае.

# ТЕРНАРНЫЙ ОПЕРАТОР

Как условный, но используется в правой части присваивания:

```
1 //какой-то код
2 int input = scan.nextInt();
3 int a = (input==0) ? 1 : input;
4 //какой-то код
```

Этот код позволит получить ввод цифры, и если ввели 0, использовать единицу. Далее это нам понадобится, чтобы избежать деления на 0.



# ПРИМЕР ДЕЛЕНИЯ ЧИСЕЛ

## ПРИМЕР ДЕЛЕНИЯ ЧИСЕЛ

```
1  import java.util.Scanner;
2
3  class Main {
4      public static void main(String[] args) {
5          System.out.println("Введите 2 числа:");
6          Scanner scan = new Scanner(System.in);
7          int a = scan.nextInt();
8          int b = scan.nextInt();
9          b = (b==0) ? 1 : b;
10         System.out.printf("a / b = %.2f", ((double)a / (double)b));
11     }
12 }
```

Эта программа делит два числа, которые вводит пользователь. Использует защиту от деления на 0, рассмотренную на предыдущем слайде. Выводит результат с двумя знаками после запятой.



## ЧТО БЫЛО ИЗУЧЕНО СЕГОДНЯ

- Структура программы — блоки, инструкции;
- Понятие переменной;
- Вывод в консоль информации;
- Чтение данных из консоли;
- Условные операторы;
- Тернарный оператор.





# ДОМАШНЕЕ ЗАДАНИЕ

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаем в чате Slack!
- Задачи можно сдавать по частям.
- Зачет по домашней работе проставляется после того, как приняты **все задачи**.



## ПОЛЕЗНЫЕ ССЫЛКИ

Самой важной ссылкой на протяжении работы с языком является спецификация языка java, рекомендую иногда обращаться к ней:

<https://docs.oracle.com/javase/specs/jls/se8/html/index.html>



**Задавайте вопросы и напишите отзыв о лекции!**

**ИЛЬДАР ГАЛИМБИКОВ**

 [grenni2007@gmail.com](mailto:grenni2007@gmail.com)

 [@grenni2007](https://t.me/grenni2007)