

# Метод, конструктор и Random class



ПРЕПОДАВАТЕЛЬ

**Фото  
преподавателя**

# Имя Фамилия

## Текущая должность

- Количество лет опыта
- Какой у Вас опыт - ключевые кейсы
- Самые яркие проекты
- Дополнительная информация по вашему усмотрению

[Корпоративный e-mail](#)

[Социальные сети \(по желанию\)](#)



# ВАЖНО:

- Камера должна быть включена на протяжении всего занятия.
- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

# ПЛАН ЗАНЯТИЯ

1. Повторение
2. Вопросы по повторению
3. Основной блок
4. Вопросы по основному блоку
5. Задание для закрепления
6. Практическая работа
7. Оставшиеся вопросы



TEL-RAN  
by Starta Institute

1

# ПОВТОРЕНИЕ ИЗУЧЕННОГО

# Повторение

- Переменные в Java
- Типы переменных
- Локальные переменные
- Переменные экземпляра
- Статические переменные
- Сравнение переменных
- Примеры типов переменных
- String в Java
- Сканер



2

# ВОПРОСЫ ПО ПОВТОРЕНИЮ

# Введение

- Основные понятия
- Объявление метода – 6 компонентов
- Типы методов в Java
- Как назвать метод?
- Random class
- Конструктор
- this





3

# ОСНОВНОЙ БЛОК

# Основные понятия

- Метод в Java или Java метод, представляет собой набор операторов, которые выполняют определенную задачу и **возвращают результат вызывающей стороне.**
- Метод Java может выполнять определенную задачу и **ничего не возвращать вызывающей стороне.**
- Методы в Java позволяют нам использовать код неограниченное количество раз без повторного написания кода.
- В Java каждый метод должен быть частью класса.



# Объявление метода – 6 компонентов

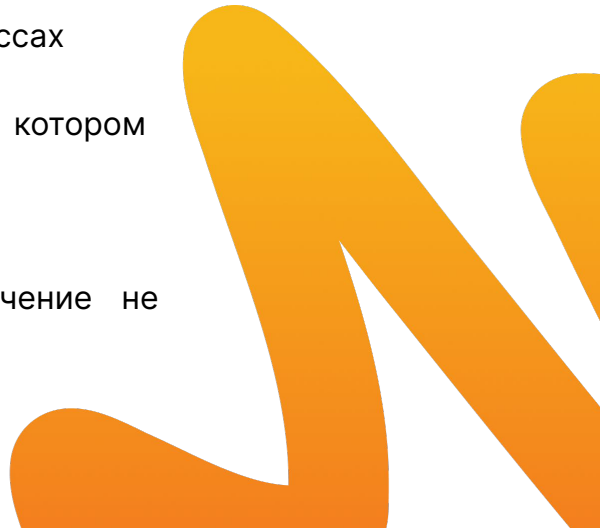
## 1. Модификатор:

определяет тип доступа к методу, т. е. откуда к нему можно получить доступ в вашем приложении. В Java существует 4 типа модификаторов доступа.

- **public:** доступен во всех классах вашего приложения
- **protected:** доступен в классе, в котором он определен, и в его подклассах
- **private:** доступен только внутри класса, в котором он определен.
- **default:** *не объявляется.* Доступен в том же классе и пакете, в котором определен.

## 2. Тип возвращаемого значения:

тип данных значения, возвращаемого методом, или `void`, если значение не возвращается.



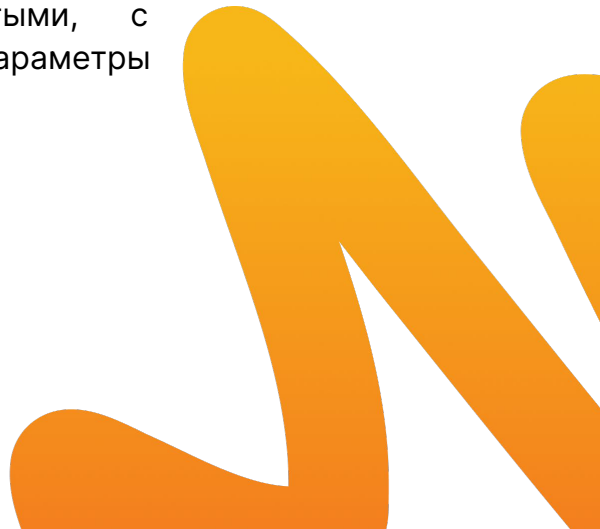
# Объявление метода – 6 компонентов

3. Имя метода:

правила для имен полей применяются и к именам методов.

4. Список параметров:

определяется список входных параметров, разделенных запятыми, с предшествующим типом данных в заключенных скобках. Если параметры отсутствуют, необходимо использовать пустые скобки ().



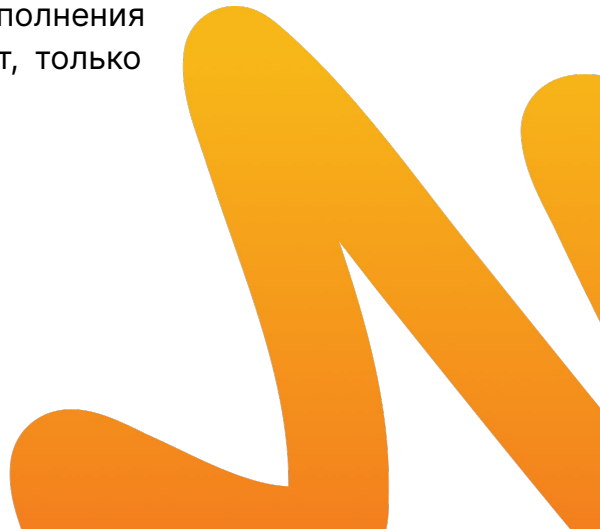
# Объявление метода – 6 компонентов

## 5. Список исключений:

исключения, которые вы ожидаете от метода, вы можете указать эти исключения.

## 6. Тело метода:

заключено в фигурные скобки. Код, который необходимо выполнить для выполнения намеченных операций, находится внутри фигурных скобок и принадлежит, только этому методу.



# Типы методов в Java

## Предопределенный метод

Метод, который уже определен в библиотеках классов Java, известный как предопределенные методы. Также известен как метод стандартной библиотеки или встроенный метод. Мы можем напрямую использовать эти методы, просто вызывая их в программе в любой момент.

## Пользовательский метод

Метод, написанный программистом, известен как пользовательский метод.

# Как назвать метод

Правила:

- имя метода должно быть глаголом и начинаться со строчной буквы.
- если состоит из более чем двух слов, первое слово должно быть глаголом, за которым следует прилагательное или существительное.
- первая буква глагола строчная, остальные прописные (camelCase).
- метод имеет уникальное имя в пределах класса, в котором он определен, но иногда метод может иметь то же имя, что и другие имена методов в том же классе, поскольку в Java разрешена перегрузка (overload) методов.

Например: `print(){}` , `printString(){}` , `validateUserName(){}`

overload: `print(int number){...}` , `print(String str){...}`



# Вызов метода

Метод должен быть вызван для использования его функциональности.

Метод всегда возвращается к коду, вызвавшему его.

При вызове метода могут быть три ситуации:

1. Он завершает все инструкции в методе
2. Он достигает инструкции возврата (return)
3. Выдает исключение







TEL-RAN  
by Starta Institute

# 4

## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ

1. Создайте метод `getSum(int a, int b)`, который ничего не возвращает, и печатает результат суммы двух полученных чисел в консоль.
2. Создайте метод `getReturnSum(int a, int b)`, который возвращает результат - сумма двух чисел, а печать в консоль производит вызывающий метод.

# Экспресс-опрос

- **Вопрос 1.**

Какие модификаторы доступа для метода вы знаете, объясните их область доступа.

- **Вопрос 2.**

Что такое перегрузка метода?

- **Вопрос 3.**

Сколько методов может быть у класса?



# Конструктор в Java

Конструкторы Java — это терминология, используемая для создания чего-либо в наших программах.

В Java, конструктор — это блок кода, аналогичный методу. Он вызывается при создании экземпляра класса.

- Конструкторы должны иметь тоже имя, что и класс.
- Конструкторы не возвращают никакого типа.
- Конструкторы вызываются только один раз во время создания объекта.



# this

Ключевое `this` слово относится к текущему объекту в методе или конструкторе.

```
// переменная
```

```
int age;
```

```
// переменная
```

```
String name;
```

```
// конструктор с двумя параметрами
```

```
Cat(int age, String name) {
```

```
    this.age = age;
```

```
    this.name = name;
```

```
}
```



5

# ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ



TEL-RAN  
by Starta Institute

1. Создайте класс Кот
2. Создайте конструктор по умолчанию
3. Создайте класс Тест
4. Создайте экземпляр класса Кот
5. Добавьте в класс Кот два конструктора с параметрами
6. Удалите конструктор по умолчанию
7. Удалите this из конструктора, переименуйте параметры

# Экспресс-опрос

- **Вопрос 1.**

Что такое `this` в конструкторе или методе класса?

- **Вопрос 2.**

Сколько конструкторов может быть создано в классе?





5

# ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ

# Random

Способы генерации случайного числа:

Способ 1: Использование Random класса

Способ 2: использование Math.random

Способ 3: использовать ThreadLocalRandom





TEL-RAN  
by Starta Institute

# 6

# ПРАКТИЧЕСКАЯ РАБОТА

# Практическое задание 1

Использование Random класса:

1. Импортировать класс `java.util.Random`
2. Создать экземпляр класса `Random` (`Random rand = new Random()`)
3. Вызовите один из следующих методов объекта `rand`:
  - a. `nextInt(upperbound)` генерирует случайные числа в диапазоне от 0 до `upper bound - 1`.
  - b. `nextFloat()` генерирует число с плавающей запятой от 0,0 до 1,0.
  - c. `nextDouble()` генерирует двойное число между 0.0 и 1.0.



# Реализация задания 1

```
public class GenerateRandom {  
    public static void main(String[] args) {  
        Random rand = new Random();  
        int upperBound = 25;  
        // generate random values from 0-24  
        int intRandom = rand.nextInt(upperBound);  
        double doubleRandom = rand.nextDouble();  
        float floatRandom = rand.nextFloat();  
  
        System.out.println("Random integer value from 0 to" + (upperBound - 1) + " : " + intRandom);  
        System.out.println("Random float value between 0.0 and 1.0 : " + floatRandom);  
        System.out.println("Random double value between 0.0 and 1.0 : " + doubleRandom);  
    }  
}
```

# Практическое задание 2

Использование Math.random

1. Объявить минимальное значение диапазона
2. Объявить максимальное значение диапазона
3. Используйте формулу  $\text{Math.random()} * (\text{max} - \text{min} + 1) + \text{min}$  для создания значений, включающих значение min и max

Этот метод можно использовать только в том случае, если вам нужно целое число или случайное значение с плавающей запятой



# Реализация задания 2

```
public static void generateNumber() {  
    int min = 50;  
    int max = 100;  
  
    //Generate random int value from 50 to 100  
    System.out.println("Random value in int from " + min + " to " + max + ":");  
    int randomInt = (int) (Math.random() * (max - min + 1)) + min;  
    System.out.println(randomInt);  
}
```



TEL-RAN  
by Starta Institute

7

# ОСТАВШИЕСЯ ВОПРОСЫ



# Домашнее задание

№1

Реализовать программу, выводящую на экран результаты:

- Сложения двух чисел
- Вычитания двух чисел
- Умножения двух чисел
- Деления двух чисел

Каждая из арифметических операций должна быть реализована как отдельный метод.



# Домашнее задание

№2

Создайте класс TestRandom, определите две переменные экземпляра num1, num2

Создайте объект random

Инициализируйте переменные рандом значениями в диапазоне от 1 - 100

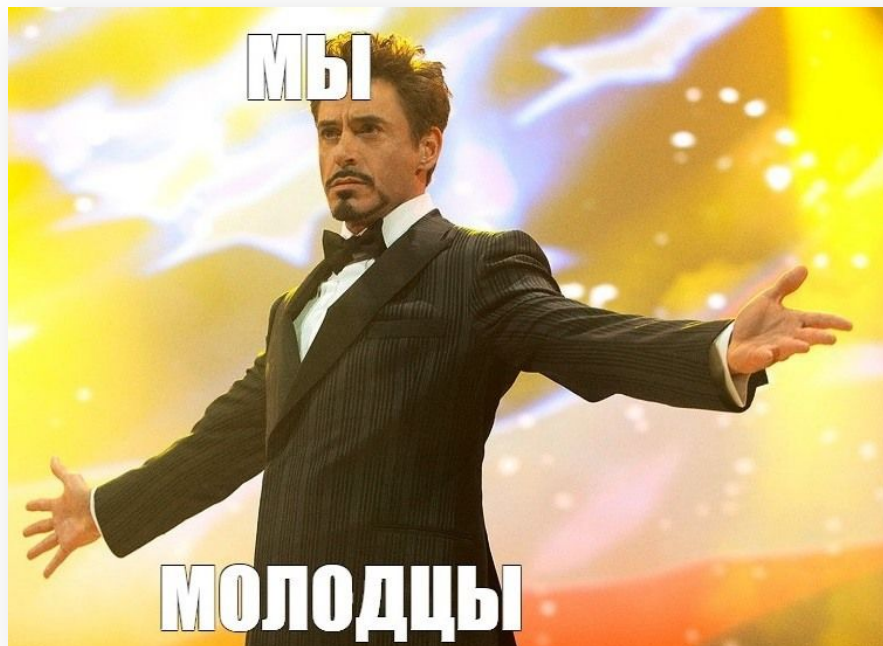
Выведите результат на консоль.



# Полезные ссылки

- [Defining Methods \(The Java™ Tutorials > Learning the Java Language > Classes and Objects\) \(oracle.com\)](#)

# ЗАКЛЮЧЕНИЕ



# Дополнительная практика

1. Программа запрашивает у пользователя сумму в Евро для конвертации
2. Реализовать метод, который конвертирует полученную сумму в эквивалент, в долларах США