

# Классы и объекты



ПРЕПОДАВАТЕЛЬ



# Артем Гордийчук

**Full-stack software engineer**

- Более 8 лет опыта работы
- Java, Spring, Hibernate, AWS, Oracle, PostgreSQL
- Проекты связанные с банковской, финансовой деятельностью, e-commerce

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

[www.linkedin.com/in/artem-g-48071a61](https://www.linkedin.com/in/artem-g-48071a61)



# 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. Задание для закрепления
5. Задание для закрепления
6. Задание для закрепления
7. Вопросы по основному блоку
8. Практическая работа
9. Оставшиеся вопросы

1

# ПОВТОРЕНИЕ ИЗУЧЕННОГО

# Повторение

- Определение цикла
- Синтаксис: while
- Как работает while, подробности
- цикл do-while
- Как работает do-while, подробности
- Разница между while и do-while





TEL-RAN  
by Starta Institute

# 2

## ВОПРОСЫ ПО ПОВТОРЕНИЮ

# Введение

- Модификаторы доступа
- Класс
- Объект
- Класс Object





3

# ОСНОВНОЙ БЛОК

# Класс

Class — это набор объектов, которые имеют общие характеристики и общие свойства.

Class не является сущностью реального мира, это просто шаблон, план или прототип, из которого создаются объекты.

Class не занимает памяти.

Class – это группа переменных разных типов данных и группа методов.

Синтаксис:

```
<access modifier> class <class name> {  
    data member;  
    method;  
    constructor;  
    nested class;  
}
```



# Класс

Class в Java может содержать:

- элементы данных
- методы
- конструкторы
- вложенные классы



# Объявления классов

Объявления классов могут включать компоненты:

Модификаторы

Класс может быть общедоступным или иметь доступ по умолчанию.

Ключевое слово `class`

Ключевое слово `class` используется для создания класса.

Имя класса

Имя должно начинаться с заглавной буквы.

Тело

Тело класса заключено в фигурные скобки `{}`.

# Модификаторы доступа

В Java используются следующие модификаторы доступа:

- **public**  
публичный, общедоступный класс или член класса. Поля и методы, объявленные с модификатором `public`, видны другим классам из текущего пакета и из внешних пакетов.
- **private**  
закрытый класс или член класса, противоположность модификатору `public`. Закрытый класс или член класса доступен только из кода в том же классе.
- **protected**  
такой класс или член класса доступен из любого места в текущем классе или пакете или в производных классах, даже если они находятся в других пакетах
- **модификатор по умолчанию**  
отсутствие модификатора у поля или метода класса предполагает применение к нему модификатора по умолчанию. Такие поля или методы видны всем классам в текущем пакете.



TEL-RAN  
by Starta Institute

# 4

## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ

1. Создайте пакет main
2. Создайте класс Main и метод main
3. Создайте новый пакет taks
4. Создайте класс Test
5. Создайте переменную с модификатором public
6. Создайте внутри пакета task подпакет sub\_task
7. Создайте в пакете sub\_task класс SubTest
8. Создайте переменную с модификатором private
9. Создайте экземпляры классов Test и SubTest в классе Main
10. Попробуйте изменить модификаторы классов и переменных
11. Проанализируйте написанный код

# Внутренние классы

В Java внутренний класс относится к классу, который объявлен внутри класса.

Существует четыре типа вложенных классов:

- Вложенный внутренний класс
- Локальные внутренние классы метода
- Статические вложенные классы
- Анонимные внутренние классы







TEL-RAN  
by Starta Institute

5

# ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ

1. Создайте класс Zoo
2. Создайте внутренний класс Monkey
3. Внутри Monkey создайте класс Banana
  - a. Создайте поле String sort = "Yellow"
  - b. Создайте метод getSort();
4. Создайте класс Test
5. Создайте экземпляр Zoo
6. Попробуйте распечатать сорт банана.
7. Проанализируйте написанный код

# Объект

Это базовая единица объектно-ориентированного программирования, представляющая объекты реальной жизни.

Объект состоит из:

## Состояние (State)

Представлено атрибутами объекта. Он также отражает свойства объекта.

## Поведение (Behavior)

Представлено методами объекта. Он также отражает реакцию объекта с другими объектами.

## Идентификатор (Identity)

Дает уникальное имя объекту и позволяет одному объекту взаимодействовать с другими объектами.



# Пример объекта

Пример объекта: Dog, Student, Car

- Identity: имя собаки
- State: возраст, цвет, порода
- Behaviors: лай, спит, ест



# Пример объекта

Объекты соответствуют вещам, найденным в реальном мире:

- графическая программа может иметь такие объекты, как «круг», «квадрат» и «меню».
- система онлайн-покупок может иметь такие объекты, как «корзина», «покупатель» и «продукт».





TEL-RAN  
by Starta Institute

# 6

## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

## ЗАДАНИЕ

1. Создайте любой объект из жизни
2. Наделите его свойствами (полями)
3. Определите поведение объекта

Например:

объект Собака, по имени “Шарик”, умеет лаять

# Экспресс-опрос

- **Вопрос 1.**

Объясните различие между: `class Cat` и `object Cat` .

- **Вопрос 2.**

Сколько памяти выделяется после написания класса `Dog` ?





# Класс Object

Все классы наследуются от класса Object.

Все классы могут реализовать те методы, которые определены в классе Object.



# Класс Object

**Object clone()**

Создает новый объект, не отличающийся от клонируемого.

**boolean  
equals(Object obj)**

Определяет, равен ли один объект другому.

**void finalize()**

Вызывается перед удалением неиспользуемого объекта.

**Class<?>  
getClass()**

Получает класс объекта во время выполнения.

**int hashCode()**

Возвращает хеш-код, связанный с вызывающим объектом.

**void notify()**

Возобновляет выполнение потока, который ожидает вызывающего объекта.

**void notifyAll()**

Возобновляет выполнение всех потоков, которые ожидают вызывающего объекта.

**String toString()**

Возвращает строку, описывающий объект.

**void wait()**

Ожидает другого потока выполнения.

**void wait(long  
millis)**

Ожидает другого потока выполнения.

**void wait(long  
millis, int nanos)**

Ожидает другого потока выполнения.

# Экспресс-опрос

- **Вопрос 1.**

Что делает метод toString() из класса Object?

- **Вопрос 2.**

Модификатор доступа: private - определите видимость?



7

# ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ



TEL-RAN  
by Starta Institute

# 8

# ПРАКТИЧЕСКАЯ РАБОТА

# Практическое задание

1. Создайте класс `Person`
2. Определите свойство класса `name` с модификатором `private`
3. Определите конструктор с параметром `name`
4. Переопределите `toString()`, таким образом, чтобы выводилось `"Name is ..."`
5. Переопределите `hashCode()`, таким образом, чтобы `hashCode` всегда был равен 10
6. Переопределите `equals()`, таким образом, чтобы сравнение было только с входящим объектом



# Реализация задания

```
public class Person {  
    private String name;  
  
    public Person(String name){  
        this.name=name;  
    }  
}
```

```
@Override  
public String toString() {  
    return "Name is " + name;  
}  
  
@Override  
public boolean equals(Object o) {  
    return this == o;  
}  
  
@Override  
public int hashCode() {  
    return 10;  
}
```

9

# ОСТАВШИЕСЯ ВОПРОСЫ



# Полезные ссылки

- [Object \(Java Platform SE 7 \) \(oracle.com\)](#)
- [Defining Methods \(The Java™ Tutorials > Learning the Java Language > Classes and Objects\) \(oracle.com\)](#)

# ЗАКЛЮЧЕНИЕ

