



Курс Java-разработчик

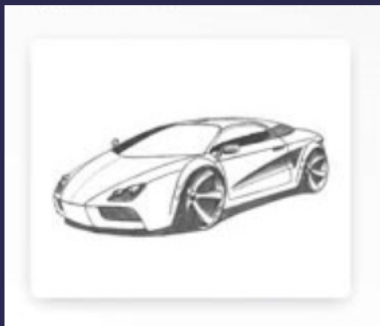


ООП



Объект

- Объект – это самоописывающая структура данных, обладающая внутренним состоянием и способная обрабатывать передаваемые ей сообщения.



Машина – это объект



Котёнок – это объект



Класс

- Класс (class) – это тип данных, значениями которого являются объекты, имеющие сходное внутреннее состояние и обрабатывающие одинаковый набор сообщений.

```
1 package com.example;
2
3 public class Cat {
4
5     public void say() {
6         System.out.println("Meow");
7     }
8
9 }
```



Класс

Класс можно рассматривать как шаблон для порождения объектов. Поэтому объекты называют также экземплярами класса (class instances).

```
1  package com.example;
2
3  public class Application {
4
5      public static void main(String[] args) {
6          var cat = new Cat();
7          cat.say();
8      }
9
10 }
```



Объявление классов в Java

В языке Java классы делятся на публичные и непубличные.

В каждом java-файле должен быть ровно один публичный класс, причём имя файла должно совпадать с именем класса.

Непубличных классов в файле может быть несколько. Непубличные классы видны только в пределах того файла, где они объявлены.



Инкапсуляция

- Инкапсуляция – принцип объектно-ориентированного программирования (ООП), заключающийся в том, что доступ к внутреннему состоянию объекта извне осуществляется только через механизм передачи сообщений.



Экземплярное поле

- Экземплярное поле – именованная составная часть внутреннего состояния объекта.

```
1  package com.example;
2
3  public class Cat {
4
5      private String name;
6      private Integer age;
7      private String color;
8
9  }
```



Экземлярный метод

- Экземлярный метод – это подпрограмма, осуществляющая обработку переданного объекту сообщения.

```
1 package com.example;
2
3 public class Cat {
4
5     public void say() {
6         System.out.println("Meow");
7     }
8
9 }
```



Экземлярный метод

- Доступ к внутреннему состоянию объекта обеспечивается за счёт передачи в экземплярный метод ссылки на объект. В Java эта ссылка передаётся неявно и имеет имя **this**.

```
1  package com.example;
2
3  public class Cat {
4
5      private String name;
6
7      public void setName(String name) {
8          this.name = name;
9      }
10 }
```



Модификаторы доступа

Модификаторы доступа применяются для управления доступом к членам класса (полям и методам):

- **private** доступ разрешён только из тела класса;
- **без модификатора** доступ разрешён для самого класса и для классов из того же пакета.
- **protected** доступ разрешён для самого класса, для классов из того же пакета, а также для наследников класса;
- **public** доступ возможен откуда угодно;



Getters и setters

При создании классов необходимо следовать инкапсуляции (т.е. необходимо как-то защитить данные от некорректного вмешательства извне).

Это можно решить с помощью специальных методов **getters** (метод для получения значения) и **setters** (метод для установки значения поля).



Getters и setters

```
1 package com.example;
2
3 public class Cat {
4
5     private String name;
6
7     public String getName() {
8         return this.name;
9     }
10
11     public void setName(String name) {
12         this.name = name;
13     }
14 }
```



Наследование

Наследование – это способ создания нового класса на основе уже существующего класса, сочетающий усложнение внутреннего состояния объектов, расширение ассортимента обрабатываемых сообщений и изменение реакции на некоторые сообщения.



Наследование

При наследовании производный класс получает все экземплярные поля базового класса и все экземплярные методы, кроме конструкторов.

```
1 package com.example;
2
3 class Animal {
4     private String name;
5
6     public String getName() {
7         return this.name;
8     }
9
10    public void setName(String name) {
11        this.name = name;
12    }
13
14    public void say() {
15        System.out.println("Я животное");
16    }
17 }
18
19 class Cat extends Animal {
20
21     public void say() {
22         System.out.println("Мяу");
23     }
24
25 }
```



Наследование

- В некоторых ситуациях применение наследования может быть нежелательным. Для этого необходимо указать ключевое слово **final**.

```
1 package com.example;
2
3 public final class Person {
4
5     // code here...
6
7 }
8
9 // данный код не скомпилируется :(
10 class Employee extends Person {
11
12     // code here..
13 }
```



Домашнее задание

Необходимо описать классы (всего должны быть семь) для зоопарка.

- Имя пакета для животных может быть например **com.zoo.animal**, а главный метод находится например в **com.zoo**
- Животным необходимо добавить возможность раздачи имен (как сделали в классе) и возможности разговора (как сделали в классе).
- Добавить еще два общих метода и два метода, который будет отличаться у каждого животного

