

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное
бюджетное учреждение высшего образования
«Московский технический университет связи и информатики»

Лабораторная работа №2
по дисциплине
«Информационные технологии и программирование»

Выполнил: студент гр. БВТ2403
Толибов Р.Ш.

Проверил:

Москва, 2025 г.

Цель работы: познакомиться с принципами ООП на языке программирования Java.

Ход работы

Создадим абстрактный класс `Animal`, содержащий в себе поля `name`, `age`, `color` и абстрактные методы `makeSound()` и `showInfo()`, а также переменную `count`, которая будет подсчитывать количество созданных объектов. Также в классе реализованы 2 конструктора (`public Animal`), геттеры и сеттеры.

```
abstract class Animal {
    protected String name;
    protected int age;
    protected String color;
    private static int count = 0;

    public Animal() {
        this.name = "Безымянный";
        this.age = 0;
        this.color = "Бесцветный";
        count++;
    }

    public Animal(String name, int age, String color) {
        this.name = name;
        this.age = age;
        this.color = color;
        count++;
    }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }

    public String getColor() { return color; }
    public void setColor(String color) { this.color = color; }

    public abstract void makeSound();
    public abstract void showInfo();

    public static int getCount() { return count; }
}
```

Далее создадим классы с животными (кошкой, попугаем и рыбой). Они будут наследовать класс `Animal`.

Создав дочерний класс `Cat`, добавим ему дополнительное поле `breed` (порода). Далее реализуем в нём конструкторы (`public Cat`). Используя ключевое слово `super()`, вызываем конструкторы класса `Animal`. Также необходимо переопределить абстрактные методы.

```
class Cat extends Animal {
    private String breed;

    public Cat() {
        super();
        this.breed = "Без породы";
    }

    public Cat(String name, int age, String color, String breed) {
        super(name, age, color);
        this.breed = breed;
    }

    @Override
    public void showInfo() {
        System.out.println("Кошка: " + name + ", Возраст: " + age + ", Цвет: "
+ color + ", Порода: " + breed);
    }

    @Override
    public void makeSound() {
        System.out.println("Кошка " + name + " мяукает");
    }
}
```

Повторим те же действия с классами Parrot и Fish.

Новое поле – canTalk (говорящий)

```
class Parrot extends Animal {
    private boolean canTalk;

    public Parrot() {
        super();
        this.canTalk = true;
    }
    public Parrot(String name, int age, String color, boolean canTalk) {
        super(name, age, color);
        this.canTalk = canTalk;
    }
    @Override
    public void makeSound() {
        if (canTalk) { System.out.println(name + " говорит: 'привет!'"); }
        else { System.out.println(name + " щебечет"); }
    }
    @Override
    public void showInfo() {
        System.out.println("Имя: " + name + ", Возраст: " + age + ", Цвет: " +
color + ", Говорящий: " + canTalk);
    }
}
```

Новое поле – waterType (тип воды)

```
class Fish extends Animal {
    private String waterType;

    public Fish() {
        super();
        this.waterType = "Любая";
    }
    public Fish(String name, int age, String color, String waterType) {
        super(name, age, color);
        this.waterType = waterType;
    }
    @Override
    public void showInfo() {
        System.out.println("Имя: " + name + ", Возраст: " + age + ", Цвет: " +
color + ", Тип воды: " + waterType);
    }
    @Override
    public void makeSound() {
        System.out.println(name + " булькает");
    }
}
```

Финальным штрихом будет реализация класса Main.

```
public class Main {  
    public static void main(String[] args) {  
        Fish fish = new Fish("Немо", 1, "Красная", "Пресная");  
        fish.showInfo();  
        Parrot parrot = new Parrot("Кеша", 5, "Красный", true);  
        parrot.showInfo();  
        System.out.println("Создано объектов: " + Animal.getCount());  
    }  
}
```

Вывод программы:

```
Имя: Немо, Возраст: 1, Цвет: Красная, Тип воды: Пресная  
Имя: Кеша, Возраст: 5, Цвет: Красный, Говорящий: true  
Создано объектов: 2
```

Вывод: в ходе выполнения работы я познакомился с ООП на Java, разобрался с основными терминами (геттеры, сеттеры, конструкторы, переопределение).

<https://github.com/TRuslan666/ITiP/tree/main/лаб2>