

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ**

Ордена Трудового Красного Знамени

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Московский технический университет связи и информатики»

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №2

Основные концепции ООП

Выполнил: Студент группы

БВТ2303

Афанасьев Александр

Москва

2024

Цель работы:

Разобраться в основных концепциях ООП (абстракция, наследование, полиморфизм, инкапсуляция) и выполнить задание своего варианта лабораторной работы (Вариант №1)

Задачи:

Создать иерархию классов в соответствии с вариантом.

Иерархия должна содержать:

- 1) абстрактный класс - 2 уровня наследуемых классов (классы должны содержать в себе минимум 3 поля и 2 метода, описывающих поведение объекта)
- 2) демонстрацию реализации всех принципов ООП (абстракция, модификаторы доступа, перегрузка, переопределение)
- 3) наличие конструкторов (в том числе по умолчанию)
- 4) наличие геттеров и сеттеров
- 5) ввод/вывод информации о создаваемых объектах
- 6) В одном из классов счетчик созданных объектов с использованием статической переменной

Ход работы:

- 1) Первым делом я хорошо изучил задание, так как условие довольно обширное, и не сразу было понятно, с чего начинать. В итоге на этом этапе я пришел к тому, что создал пустые классы без методов, статических переменных и атрибутов. Только сами названия классов и их структуру уровней. Получился базовый абстрактный класс Animals, классы первого наследуемого уровня Mammals (млекопитающие), Birds (птицы) и Fishes (Рыбы). От этих классов было создано ещё по одному наследнику – Cats, Parrot и Fingerling (рыбка)
- 2) Далее я открывал задачи лабораторной работы и постепенно их выполнял. Создал 3 атрибута, а также конструкторы с параметрами и без в базовом классе Animal (выполнено

условие создание конструкторов и перегрузки методов),
создал абстрактную функцию acceleration, которая требует её
реализации у наследников, а также геттеры и сеттеры для
всех величин, чтобы избежать повторения кода

- 3) Далее в каждом из трех наследников я создавал
конструкторы, которые наследовались от базового класса, а
также имели один свой параметр – color, к которому был
добавлен свой геттер и сеттер
- 4) Дальше в наследниках второго уровня я определил
конструкторы, переписал метод acceleration, добавил метод
plusFat(), а в классе Cats создал static переменную, которая
считает кол-во котов

Вывод: познакомился с концепциями ООП в Java, практиковался
на примере решения задачи данной лабораторной работы и в
целом улучшил своё владение языком