**Домашняя работа Abstraction**

(\* - не значит сложно, просто дополнительно, кто хочет)

### **Задача 1: Медиа-проигрыватель**

Создайте интерфейс MediaPlayer, который будет определять методы для проигрывания и остановки медиа-файлов.

1. Создайте интерфейс MediaPlayer с методами play() и stop().
2. Реализуйте классы AudioPlayer и VideoPlayer, которые реализуют интерфейс MediaPlayer и предоставляют собственные реализации методов play() и stop() для аудио и видео соответственно.
3. Создайте объекты для проигрывания аудио и видео и вызовите методы play() и stop() для каждого из них.

### **Задача 2\*: Файловый менеджер**

Создайте интерфейс FileManager, который определяет методы для копирования и удаления файлов.

1. Создайте интерфейс FileManager с методами copyFile() и deleteFile().
2. Реализуйте классы LocalFileManager и CloudFileManager, которые реализуют интерфейс FileManager и предоставляют свои собственные реализации для копирования и удаления файлов на локальном устройстве и в облаке соответственно.
3. Создайте объекты для управления файлами на локальном устройстве и в облаке, и выполните копирование и удаление файлов.

### **Задача 3: Школа**

Смоделируйте школьную систему управления учебными предметами с использованием абстрактного класса Subject.

1. Создайте абстрактный класс Subject с методами для получения информации о предмете (например, getSubjectName(), getTeacher()) и абстрактным методом study(), который будет реализован в подклассах.
2. Реализуйте подклассы, такие как Math, History, Science, которые наследуются от Subject и предоставляют свои собственные реализации метода study() для каждого учебного предмета.
3. Создайте объекты различных учебных предметов и вызовите метод study() для каждого из них.

### **Задача 4\*: Ресторан**

Смоделируйте систему заказа блюд в ресторане с использованием абстрактного класса Dish.

1. Создайте абстрактный класс Dish с общими характеристиками блюда (например, name, price, ingredients) и абстрактным методом prepare().
2. Реализуйте подклассы, такие как Salad, Pasta, которые наследуются от Dish и предоставляют свои собственные реализации метода prepare() для приготовления каждого блюда.
3. Создайте объекты различных блюд из меню ресторана и вызовите метод prepare() для каждого из них.

**Справка:**

<https://drive.google.com/drive/folders/1tYlH5CJyISK3BvSUe3p1B5BQ5EwDnuhe?usp=drive_link>

В классах лежит файл «Абстрактный класс и интерфейс»