**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление информатика и вычислительная техника

**Отчет**

по лабораторной работе №4

по дисциплине

**«Технологии программирования»**

Выполнил:

Студент группы 8В31 И.М. Болбин

Проверил

Программист ОИТ ИШИТР В.Д Тарасенко

**Цель работы:** познакомится с разбиением приложения на слои. Освоить основные принципы объектно-ориентированного проектирования. Научиться применять шаблон Factory Method.

**Ход работы.**

Ознакомится с поставленной задачей, выполнить поставленную задачу согласно всем требованиям, оформить отчёт.

**Задание:**

Необходимо создать логику приложения в соответствии с вариантом задания. При выполнении лабораторной работы необходимо соблюдать следующие правила:

1. Необходимо создавать логику приложения в отдельных классах;
2. Классы    логики    приложения    не    должны    зависеть     от    интерфейса пользователя, их необходимо выделить в отдельный пакет либо библиотеку классов;
3. Для    использования    логики   приложения    в    интерфейсе    пользователя необходимо вызывать методы классов, содержащих логику.
4. Для создания объектов предметной области необходимо использовать шаблон Factory Method.
5. Программу необходимо реализовать в виде консольного приложения на языке C++

**Листинг кода:**

Листинг 1: Header класса AudioFile.

class AudioFile{

    public:

    virtual ~AudioFile() = default;

    virtual std::string play() const = 0;

};

Класс AudioFile является абстрактным, создан как шаблон для дочерних классов.

Листинг 2: Header класса MP3File.

class MP3File: public AudioFile{

    public:

    ~MP3File();

    std::string play() const override;

};

Класс MP3File является конкретным примером класса AudioFile.

Листинг 3: Header класса AudioPlayer.

std::string AudioPlayer::playAudioFile() const{

    AudioFile\* file = this->createAudioFile();

    std::string output = file->play();

    delete file;

    return output;

};

Класс AudioPlayer является абстрактным классом конструктором для создания конкретных объектом классов AudioFile.

Листинг 4: Header класса MP3Player.

AudioFile\*  MP3Player::createAudioFile() const{

    return new MP3File();

};

Класс AudioPlayer является классом сборником для объектов классов MP3File.

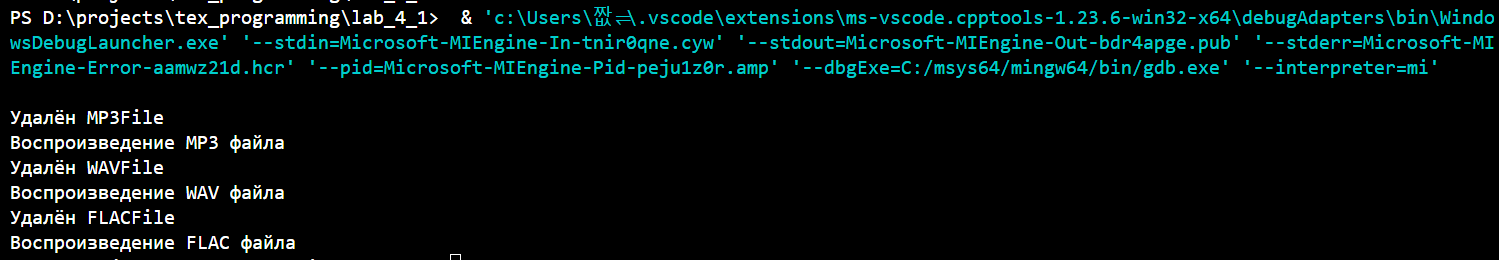


Рисунок 1 – Результат работы программы.

**Вывод.**

В результате лабораторной работы был получен опыт разработки архитектуры проекта с использование Factory Method.

**Приложение.**

**Cсылка на репозиторий Github:**

**https://github.com/Eretik1/tex\_programming\_lab\_4.git**