Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №9

на тему:

**«Множественное наследование»**

БГУИР 6-05-0612-02 52

|  |
| --- |
| Выполнил студент группы 353505  Камышев Степан Владиславович |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

1 Описать семейство классов, имеющих общий функционал (), при этом в каждом классе присутствует дополнительно свой функционал. Набор дополнительных функций в разных классах может быть произвольным.

2 Дополнительный функционал описать в виде набора интерфейсов

3 Одна из общих функций должна быть реализована по-своему в каждом классе.

4 Одна из общих функций должна быть реализована в других классах (например, изменение скорости, использование оружия, доставка груза). При этом должно быть несколько вариантов реализации (несколько классов), например, персонажам игры доступны разные инструменты – каждый инструмент может использоваться разными персонажами. Конкретный вариант реализации выбирается при создании объекта (применить шаблон проектирования «Мост» («Bridge»)).

5 Для создания объектов использовать шаблон проектирования «Абстрактная фабрика» (Abstract factory) или «Построитель» (Builder)

6 В классе Program создать коллекцию разных объектов (см. п.1). Затем для каждого элемента коллекции вызвать все методы, доступные для данного объекта.

Предметная область: Гостиница.

# 2 Выполнение работы

Для выполнения данной лабораторной работы первым делом был создан проект, к которому позже добавлялись классы и интерфейсы, реализующие данное задание.

Для проектирования был выбран шаблон проектирования «Abstract factory», так как он позволяет создавать различные объекты одного семейства классов, а шаблон «Builder» применяется для создания объекта одного и того же класса, но с различными характеристиками.

Был реализован абстрактный класс «AbstractFactory», который объявил методы для создания объектов абстрактных классов «Hotel», «Hostel» и «Farmstead». У этих трех классов имеются по два наследника, которые подразделяют эти классы на два типа: Польский и Токийский. У класса «AbstractFactory» также имеется два наследника, которые реализуют его методы. Методы этих классов создают объекты классов-потомков, но возвращают их как объекты класса-родителя. Это позволяет хранить их в объекте List типа данных класса-родителя, но использовать при этом методы классов-потомков.

Для применения шаблона проектирования «Bridge» был создан интерфейс «IPoll», который объявляет метод для создания бассейна. Этот интерфейс реализуется в двух классах, каждый из которых имеет свою реализацию. Для выбора реализации в классе “Hotel” и его классах-наследниках было создано поле, которое может принимать объект любого класса, который реализует интерфейс «IPool». Это позволяет вызывать любую желаемую из существующих реализацию.

Оба эти шаблона проектирования применяются для повышения гибкости кода и уменьшения зависимости абстракции от реализации.

### Также был создан интерфейс «IMusic», который был реализован в классе «TokioHotel». Методы интерфейса «IMusic», реализованные в классе «TokioHotel» позволяют воспроизводить аудиофайлы расширения “wav” (для этого было необходимо подключить к проекту некоторые nuget-пакеты).

После реализации всех классов было создано меню взаимодействия с пользователем, которое позволяет создавать различные объекты реализованных классов, собирать их в коллекции и вызывать их методы, к примеру выводить информацию о каждом объекте.



Рисунок 1 – Пользовательский интерфейс программы

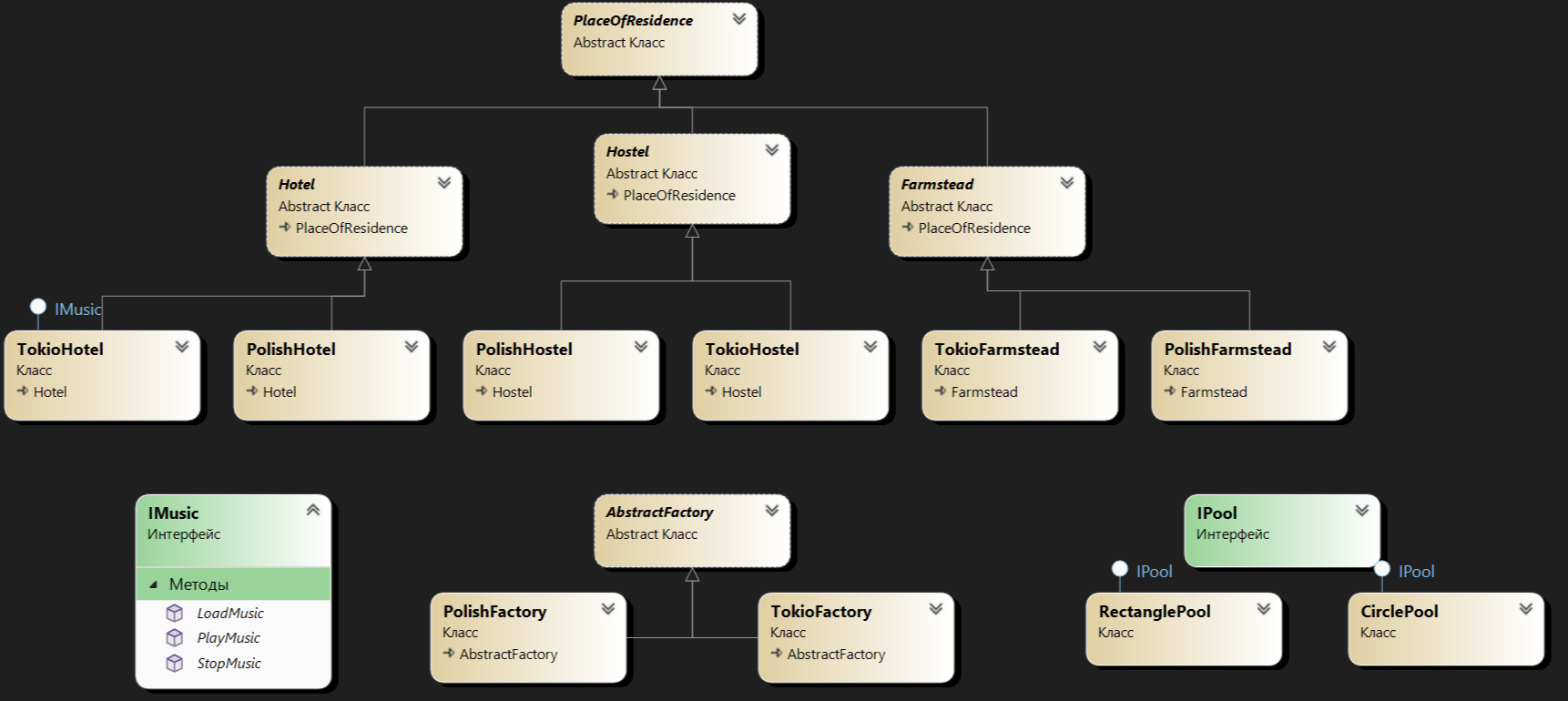


Рисунок 2 – Диаграмма классов, используемых в проекте



Рисунок 3 – Вывод программой информации о различных отелях, которые ранее были созданы пользователем



Рисунок 4 – Создание нового отеля

# Вывод

В ходе лабораторной работы были углублены знания работы с интерфейсами в языке программирования C#. Были изучены шаблон проектирования «Bridge», «Builder» и «Abstract factory», благодаря которым повышается гибкость кода, а также независимость абстракции от реализации, что значительно упрощает расширение и обновление существующего программного обеспечения. Было на практике отработано делегирования конструкторов, а также значительно углублены знания о наследовании и полиморфизме в языке C#. В ходе выполнения этой работы мною были изучены nuget-пакеты, которые предоставляют классы, позволяющие работать с аудиофайлами, к примеру включать музыку.