МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)

БГТУ.СМК-Ф-4.2-К5-02

Факультет О Естественнонаучный
шифр наименование

Кафедра О7 Информационные системы и программная инженерия
шифр наименование

Дисциплина Программирование на языке высокого уровня

КУРСОВАЯ РАБОТА
на тему

Объектно-ориентированная разработка программ с графическим пользовательским интерфейсом «сверху-вниз»: предварительное выявление классов, объектов и их отношений. Вариант: Создание многооконного приложения

Выполнил студент группы И501Б

Крылов И.О.
Фамилия И.О.

РУКОВОДИТЕЛЬ

Вальштейн К.В.
Фамилия И.О. Подпись

Оценка _____

«_____» 2022 г.

_____

САНКТ-
ПЕТЕРБУРГ 2021 г.

Содержание

ВВЕДЕНИЕ

Язык C# является наиболее известной новинкой в области создания языков программирования. Ввиду очень удобного объектно-ориентированного дизайна, C# является хорошим выбором для конструирования различных компонентов – от высокоуровневой бизнес логики до системных приложений, использующих низкоуровневый код. Также следует отметить, что C# является и Web ориентированным - используя встроенные конструкции языка компоненты, могут быть превращены в Web сервисы, к которым можно будет обращаться из Internet посредством любого языка на любой операционной системе.

В C# унифицирована система типов, можно рассматривать каждый тип как объект. Несмотря на то, используется класс, структура, массив или встроенный тип, можно обращаться к нему как к объекту. Объекты собраны в пространства имен (namespaces),которые позволяют программно обращаться к чему-либо. Это значит, что вместо списка включаемых файлов заголовков в своей программе необходимо написать какие пространства имен, для доступа к объектам и классам внутри них, будут использоваться. В C# выражение using позволяет не писать каждый раз название пространства имен, когда необходимо использовать класс из него. Например, пространство имен System содержит несколько классов, в том числе и Console.

Цель данной курсовой работы -- создание многооконного приложения. Задачи для достижения данной цели:

Создание иерархии классов;

Создание кнопочной формы;

Объединение иерархии классов и кнопочной формы;

Запись результатов в файл.

1 Постановка задачи

Создание иерархии классов было осуществлено с помощью языка c#.

Кнопочная форма реализована с помощью технологии Windows Forms.

Объединение двух этих компонентов было произведено благодаря средствам языка c#, например делегатам.

Запись результатов в файл осуществляется средствами языка c# с использованием функций сериализации и десериализации.

2 Описание разработанной программы

2.1 Иерархия классов

На рисунке 1 представлена диаграмма классов. В программе используется 7 классов, 1 из которых абстрактный и 3 интерфейса.

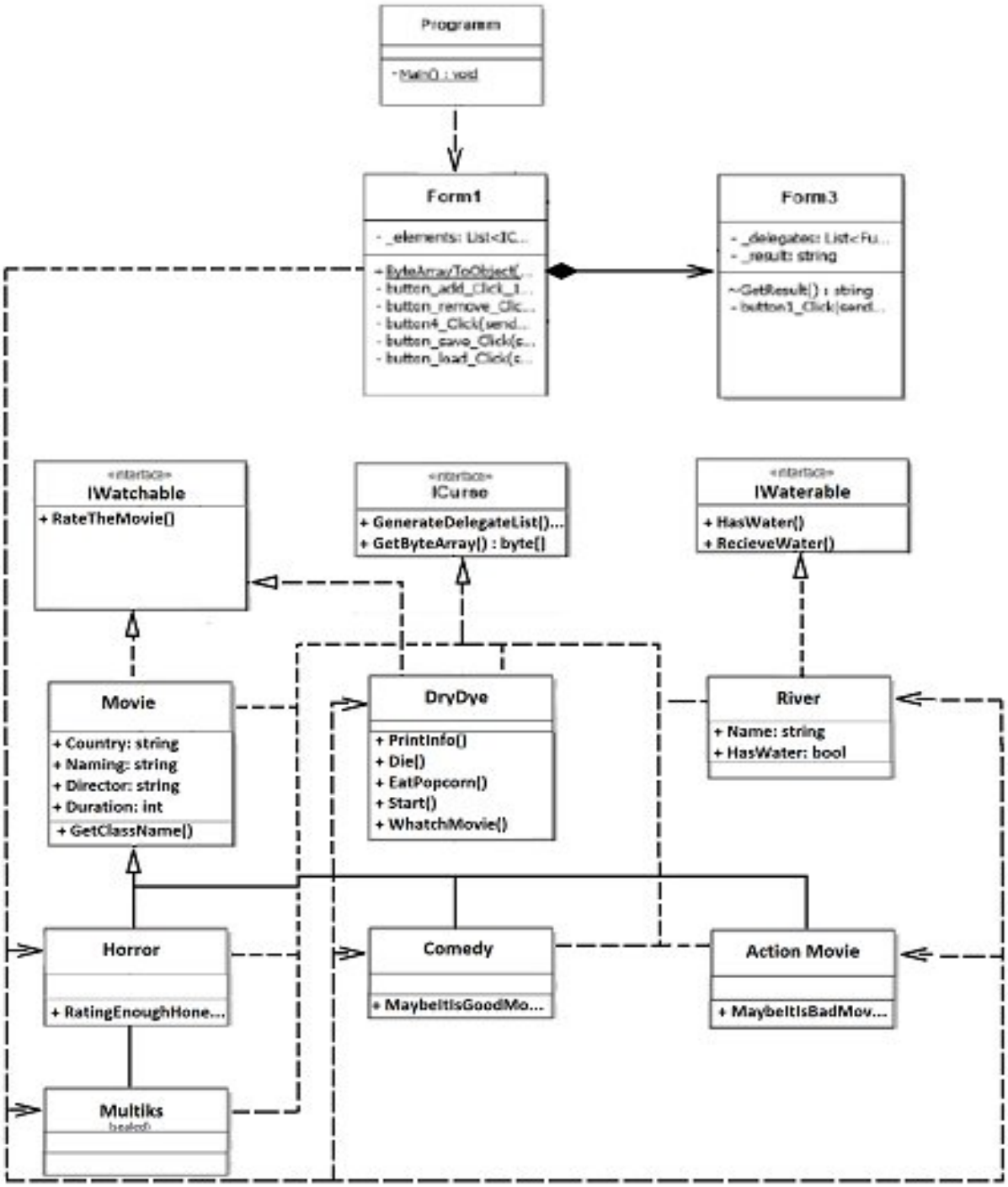

Рисунок 1 - Диаграмма классов

2.2 Класс “Programm”

Класс содержит метод Main, в котором находится главная точка входа в приложение.

2.3 Класс “Form1”

Кнопочная форма, с помощью которой пользователь взаимодействует с приложением.

Содержит поле:

List<ICurse> _elements; - Список классов, которые пользователь может использовать в приложении.

Методы:

private void button_add_Click_1(object sender, EventArgs e) - Кнопка, при нажатии на которую, в список добавляется класс, выбранный пользователем.

Object sender - ссылка на данную кнопку. [1]

EventArgs e - параметр, содержащий данные о событии. [2]

private void button_remove_Click(object sender, EventArgs e) - Кнопка, при нажатии на которую, из списка удаляется выбранный элемент.

Object sender - ссылка на данную кнопку.

EventArgs e - параметр, содержащий данные о событии.

private void button4_Click(object sender, EventArgs e) - Кнопка, при нажатии на которую вызывается Form3 - форма для выбора метода класса.

Object sender - ссылка на данную кнопку.

EventArgs e - параметр, содержащий данные о событии.

private void button_save_Click(object sender, EventArgs e) - Кнопка, при

нажатии на которую список сохраняется в файл.

Object sender - ссылка на данную кнопку.

EventArgs e - параметр, содержащий данные о событии.

private void button_load_Click(object sender, EventArgs e) - Кнопка, при нажатии на которую список загружается из выбранного файла.

Object sender - ссылка на данную кнопку.

EventArgs e - параметр, содержащий данные о событии.

2.4 Класс “Form3”

string _result - поле, в котором хранится строка-результат выбранного метода.

List<Func<string>> _delegates - список делегатов.

internal string GetResult() - метод, который возвращает результат.

private void button1_Click(object sender, EventArgs e) - Кнопка, при нажатии на которую, в окно вывода выводится результат выбранного метода.

Object sender - ссылка на данную кнопку,

EventArgs e - параметр, содержащий данные о событии.

2.5 Абстрактный класс “Movie”

public abstract string RateTheMovie() – Оценка фильма

public virtual string WhatchMovie() – Просмотр фильма

public virtual string EatPopcorn() – Поесть попкорн

public virtual string GetClassName() – Получить имя класса

public List<Func<string>> GenerateDelegateList() - Метод для добавления методов класса в список делегатов. [3]

public byte[] GetByteArray() - Метод, который сериализует экземпляр класса. [5]

2.6 Класс “ActionMovie”

public override string RateTheMovie() - Оценка

public String MaybeItIsGoodMovie() – Может это хороший фильм?

public override string GetClassName() – Имя класса

public new List<Func<string>> GenerateDelegateList() - Метод для добавления методов класса в список делегатов.

public byte[] GetByteArray() - Метод, который сериализует экземпляр класса. Возвращает массив байтов.

2.7 Comedy

public override string RateTheMovie() - Оценка

public string MaybeItIsBadMovie() - Может это плохой фильм?

public override string GetClassName() –Имя класса

public new List<Func<string>> GenerateDelegateList() - Метод для добавления методов класса в список делегатов.

public byte[] GetByteArray() - Метод, который сериализует экземпляр класса. Возвращает массив байтов.

2.8 Класс Horror

public override string RateTheMovie() - Оценка

public string RatingEnoughHonest() – Достаточно ли объективная оценка?

public override string GetClassName() - Имя класса

public new List<Func<string>> GenerateDelegateList() - Метод для добавления методов класса в список делегатов.

public byte[] GetByteArray() - Метод, который сериализует экземпляр класса. Возвращает массив байтов.

2.9 Класс Multiks (sealed)

public override string RateTheMovie() - Оценка

public string RatingEnoughHonest() – Достаточно ли объективная оценка?

public override string GetClassName() - Имя класса.

public new List<Func<string>> GenerateDelegateList() - Метод для добавления методов класса в список делегатов.

public byte[] GetByteArray() - Метод, который сериализует экземпляр класса. Возвращает массив байтов.

2.10 Класс DryDye

public override string RateTheMovie() - Оценка

public string WhatchMovie() – Просмотр

public string EatPopcorn() – Покушать попкорна

public new List<Func<string>> GenerateDelegateList() - Метод для добавления методов класса в список делегатов.

public byte[] GetByteArray() - Метод, который сериализует экземпляр класса. Возвращает массив байтов.

2.11 Класс River

public string RecieveWater() – получить воду

public new List<Func<string>> GenerateDelegateList() - Метод для добавления методов класса в список делегатов.

public byte[] GetByteArray() - Метод, который сериализует экземпляр класса. Возвращает массив байтов.

3 Демонстрация работы

На рисунке 2 показано главное меню многооконного приложения.

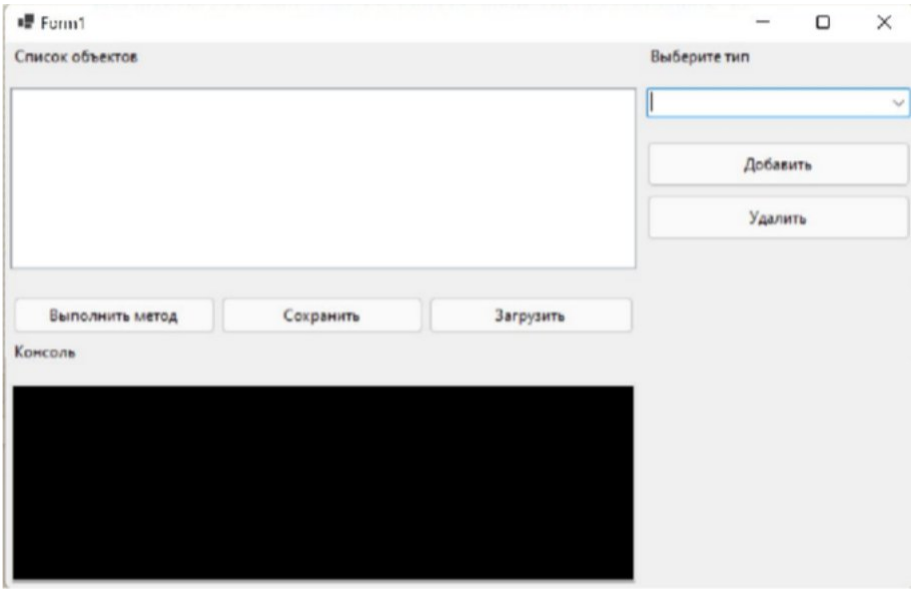

Рисунок 2 - Запуск приложения.

На рисунке 3 показана возможность выбора класса для добавления его экземпляра в список.

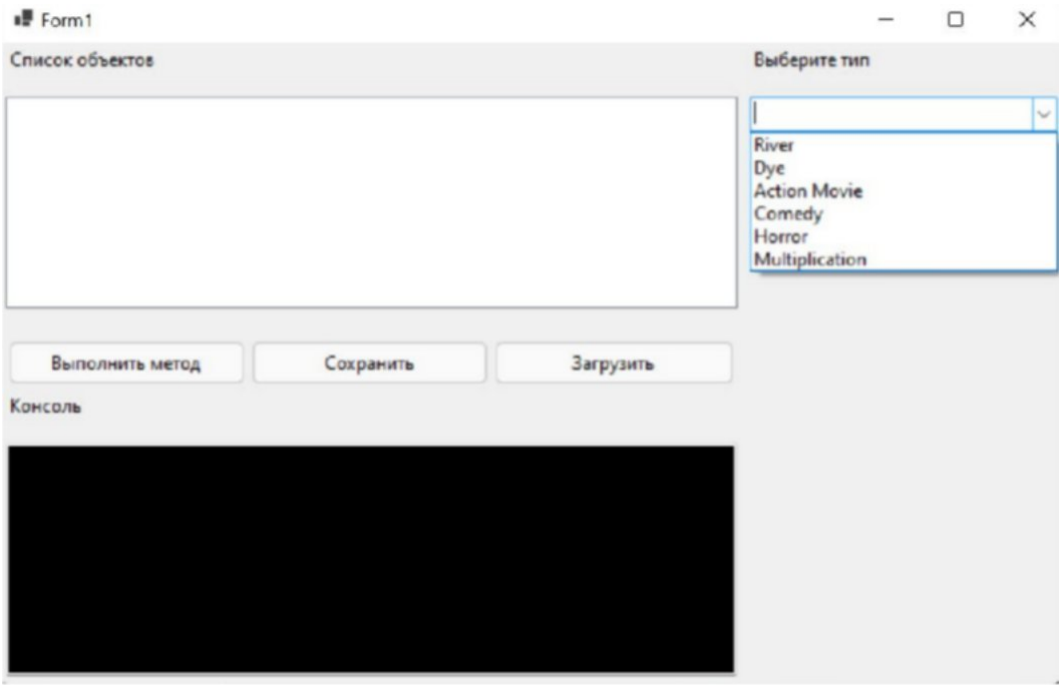

Рисунок 3 - Меню выбора типа.

На рисунке 4 пользователь заполнил список экземплярами всех классов.

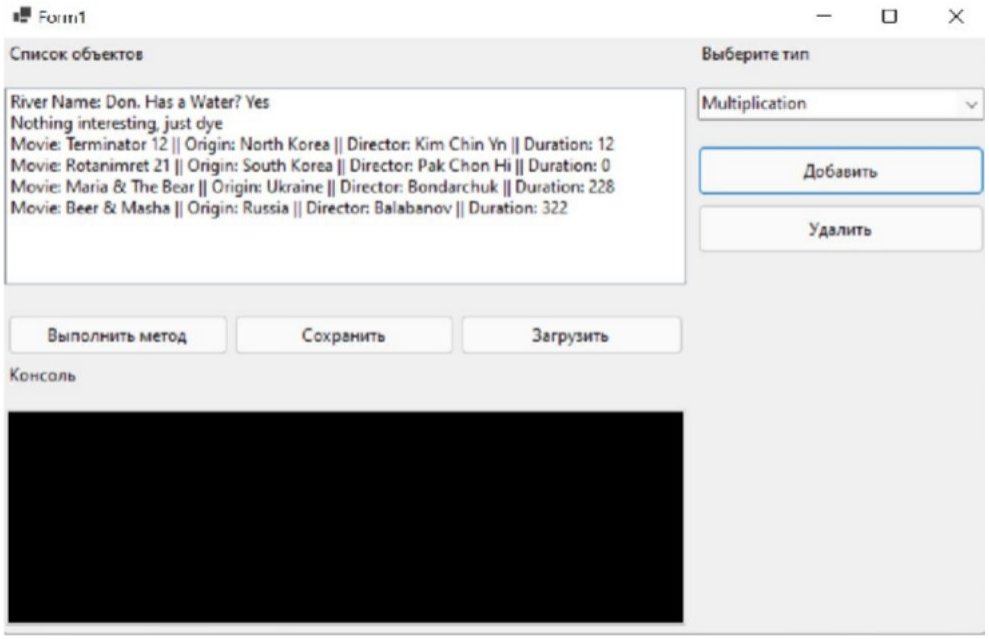

Рисунок 4 - Экземпляры всех классов добавлены в список.

На рисунке 5 отображено окно для выбора метода для экземпляра выбранного класса. У каждого класса свой уникальный набор методов.

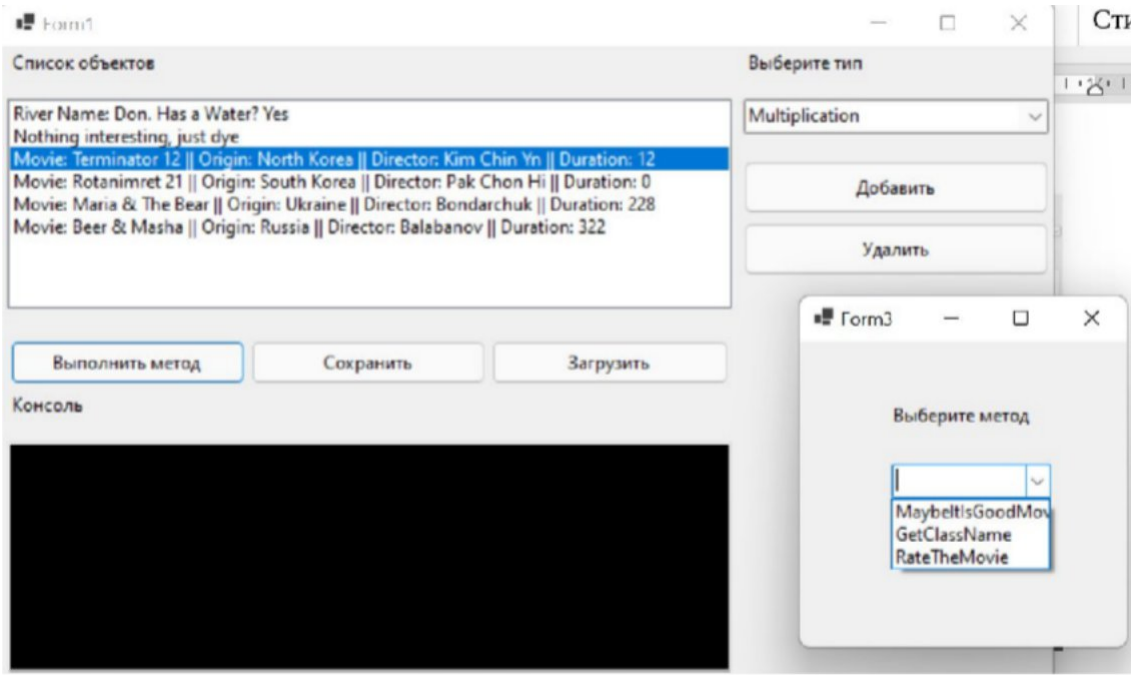

Рисунок 5 - Выбор метода выбранного элемента

На рисунке 6 пользователь выполнил метод выбранного элемента списка и результат отобразился в текстовом поле - “Консоль”. При этом в спискеэлемент изменился, так как метод изменяет поля класса.

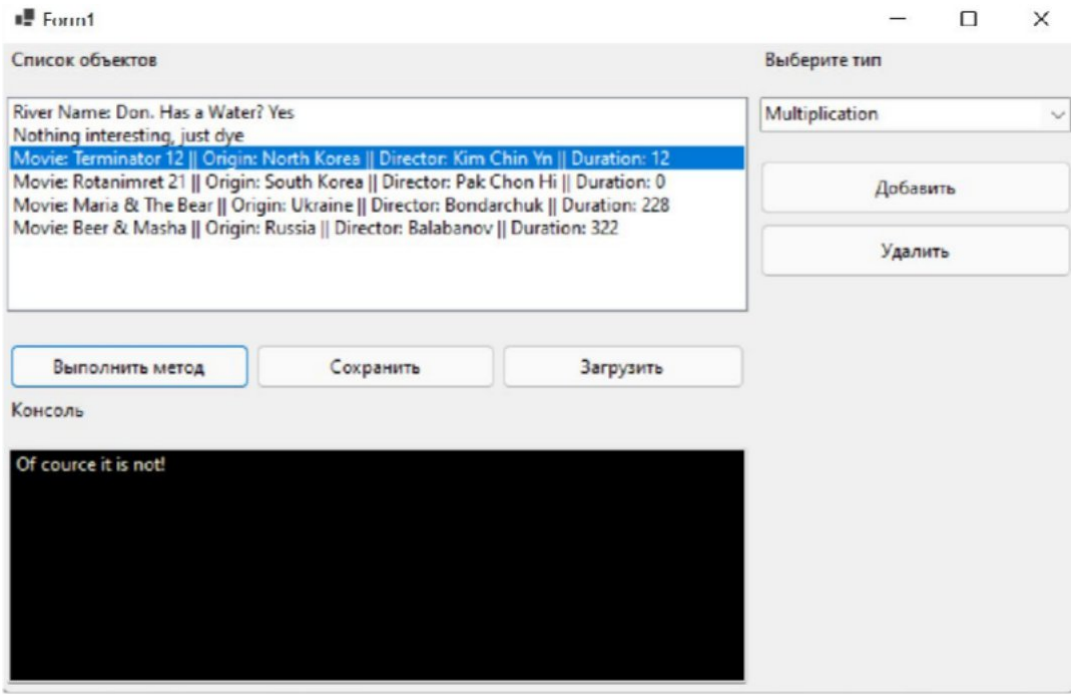

Рисунок 6 - Выполнение метода и вывод результата в консоль.

На рисунке 7 показан результат удаления элемента из списка.

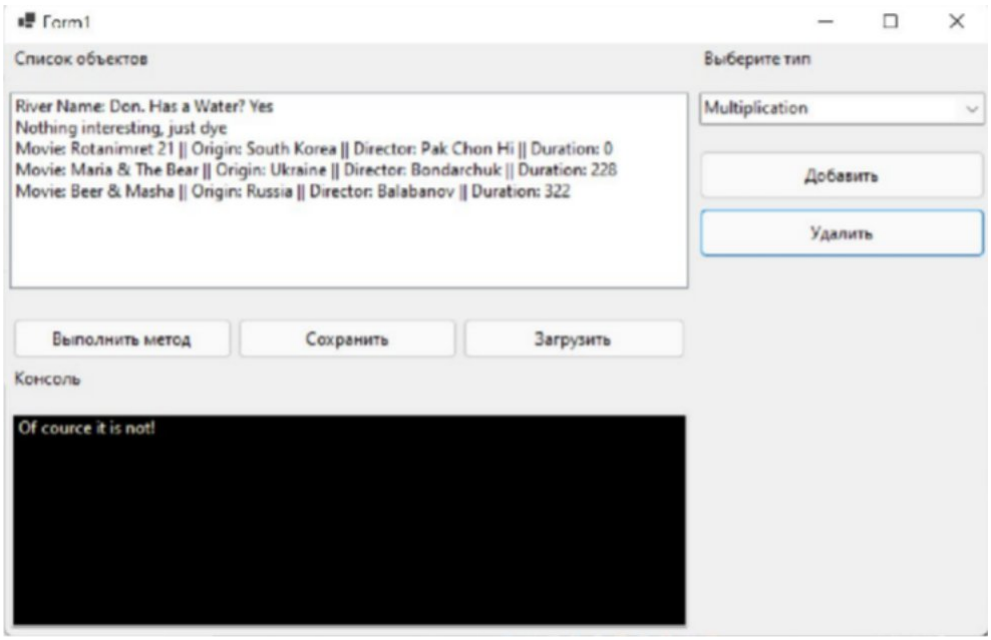

Рисунок 7 - Удаление элементов из списка.

На рисунке 8 пользователь нажал кнопку “Сохранить”. Эта кнопка вызывает системное окно для записи списка в файл.

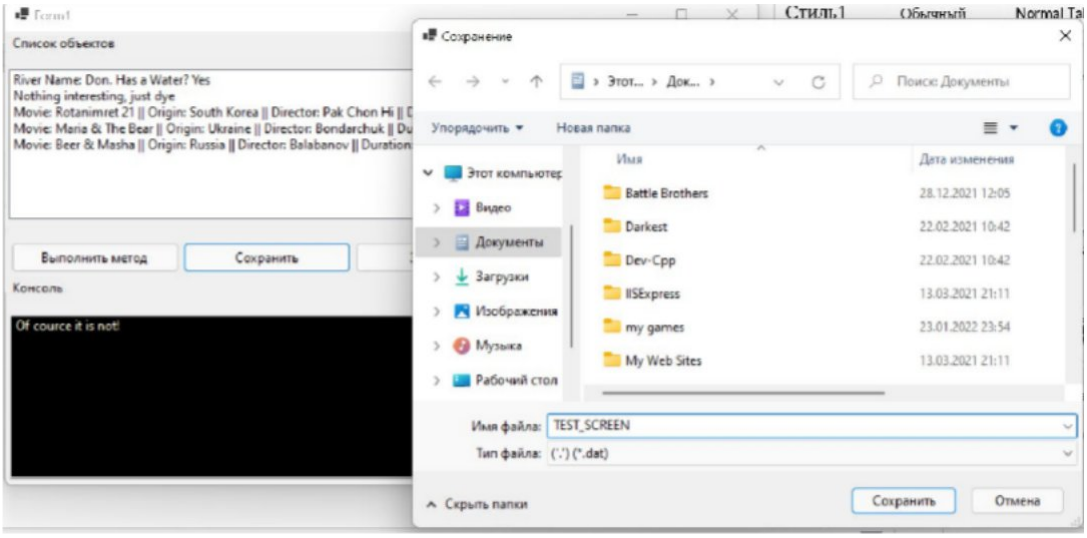

Рисунок 8 - Сохранение в файл.

На рисунке 9 показан результат нажатия кнопки “Загрузить” - открывается системное меню, в котором пользователь может выбрать какой файл открыть, и, соответственно, какой список загрузить в приложение.

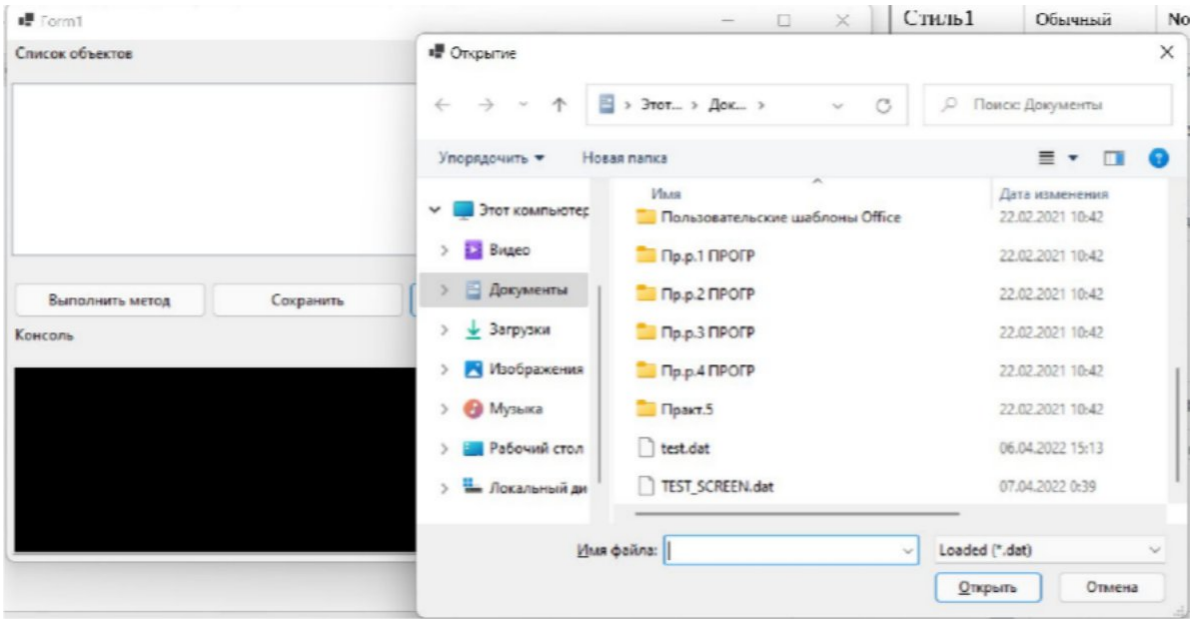

Рисунок 9 - Загрузка списка из файла.

На рисунке 10 отображён результат загрузки из файла. В пустой список добавились элементы, которые находились в файле.

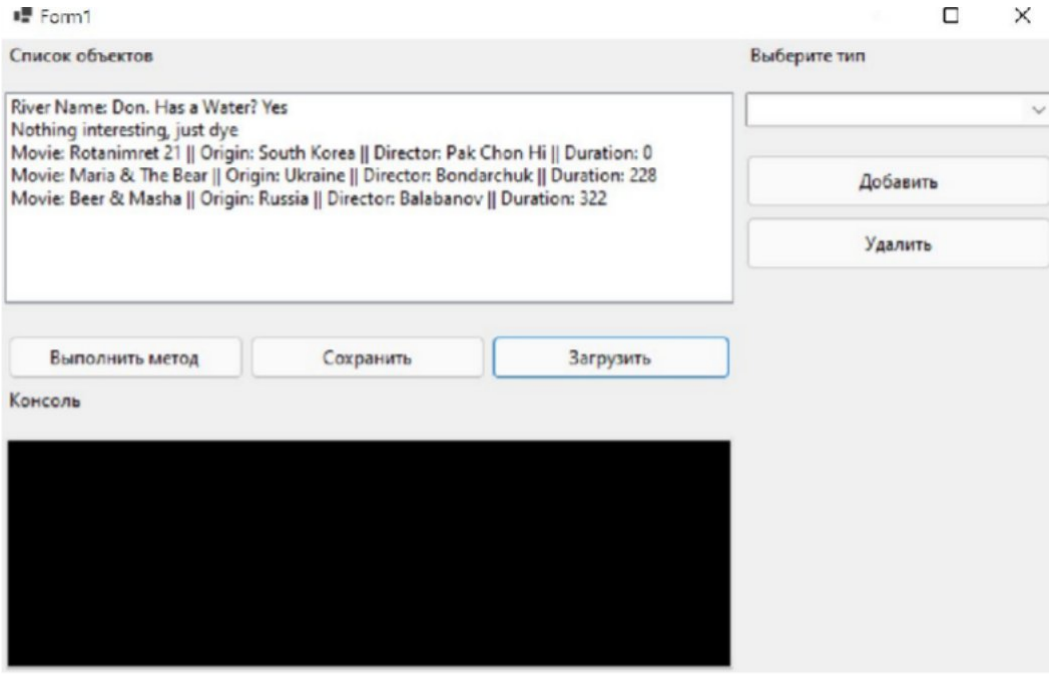

Рисунок 10 - Результат загрузки.

ЗАКЛЮЧЕНИЕ

Во время написания курсовой работы были приобретены навыки работы с языком программирования C# и технологией windows forms.

Было разработано многооконное приложение с возможностью сохранения информации в файл и чтения из него. С помощью Windows Forms реализована кнопочная форма, позволяющая пользователю использовать многооконное приложение. В процессе реализации данной задачи использовались основные свойства объектно-ориентированного программирования. Все поставленные цели и задачи выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Windows Forms [Электронный ресурс] Официальная документация по технологии windows forms URL: https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/?view=netdesktop-5.0 (дата обращения: 12.11.2021).

2. Metanit.com [Электронный ресурс] Руководство по программированию в Windows Forms URL: https://metanit.com/sharp/windowsforms/ (дата обращения: 12.11.2021).

3. Техническая документация Майкрософт [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru (дата обращения: 12.11.2021).

4. ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – Введ. 2018-07-01. М.: ФГУП СТАНДАРТИНФОРМ, 2018.

5. Мюллер Д. П., Семпф Б., Чак С. C# для чайников. – Диалектика-Вильямс, 2019. – 608 с.

ПРИЛОЖЕНИЕ А

Папка с проектом прилагается к отчету