**Частное образовательное учреждение высшего образования**

**«Международный Институт Дизайна и Сервиса»**

**(ЧОУВО МИДиС)**

Кафедра математики и информатики

**КУРСОВАЯ РАБОТА**

по дисциплине: «Объектно-ориентированный анализ и программирование

на языке С#»

на тему: «**Проектирование и разработка АСУ «Кинотеатр»**

Направление подготовки 09.03.03 Прикладная информатика,

направленность (профиль) Прикладная информатика в экономике

Научный руководитель

доцент КМиМ

С.С.Чеботарев

Автор работы

Студент группы ПИ-224

Кочкина Анастасия Алексеевна

Челябинск 2022

**Содержание**

Оглавление

[Введение 3](#_Toc116396735)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc116396736)

[1.1 Описание предметной области 5](#_Toc116396737)

[1.2 Выделение словаря предметной области 7](#_Toc116396738)

[1.3 Виды связей между объектами 7](#_Toc116396739)

[1.3.1 Связь «один к одному» 7](#_Toc116396740)

[1.3.2 Связь «один ко многим» 8](#_Toc116396741)

[1.3.3 Связь «многие ко многим» 8](#_Toc116396742)

[1.4 Анализ существующих программных решений 9](#_Toc116396743)

[1.5 Техническое задание 16](#_Toc116396744)

[1.5.1 Общие сведения 17](#_Toc116396745)

[1.5.2 Назначение АСУ 17](#_Toc116396746)

[1.5.3 Характеристика объекта автоматизации 18](#_Toc116396747)

[1.5.4 Требования к системе 19](#_Toc116396748)

[1.5.5 Состав и содержание работ по созданию системы 20](#_Toc116396749)

[1.5.6 Условия эксплуатации 20](#_Toc116396750)

[2.1 Проектирование и разработка классов 20](#_Toc116396751)

[2.2. 1 Форма получения информации 23](#_Toc116396752)

[2.2. 2 Форма редактирования данных о кинотеатрах 24](#_Toc116396753)

[2.2. 3 Форма редактирования данных о фильмах 25](#_Toc116396754)

[2.2.4 Форма редактирования данных о сеансах 26](#_Toc116396755)

[2.2.5 Форма редактирования данных о билетах 26](#_Toc116396756)

[2.2.6 Форма получения данных 27](#_Toc116396757)

[2.2.7 Форма редактирования данных о репертуаре кинотеатров 28](#_Toc116396758)

[2.2.8 Форма для просмотра конкретных данных о фильме 28](#_Toc116396759)

[2.2.9 Форма для просмотра адреса кинотеатра 29](#_Toc116396760)

[2.2.10 Форма для просмотра данных по всем таблицам 30](#_Toc116396761)

[2.3 Описание и разработка базы данных 32](#_Toc116396762)

[2.4 Краткое руководство пользователя по эксплуатации 34](#_Toc116396763)

[Системы 34](#_Toc116396764)

[ЗАКЛЮЧЕНИЕ 35](#_Toc116396765)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36](#_Toc116396766)

# Введение

«Проектирование и разработка АСУ «Кинотеатр»» - так звучит названия данной курсовой работы. В современном мире происходят такая тенденция, как постоянно и непрерывно увеличивающийся поток информации. В связи с этим появляется острая необходимость в процессах, которые могли бы приводить информацию к общепринятому виду, а также в автоматизации рутинной, монотонной работы. Проектирование и разработка такой системы позволит увеличить производительность и эффективность труда, а также качество и скорость работы. Именно поэтому эта тема является актуальной. Разработка системы должна обеспечивать хранение информации о кинотеатрах, о фильмах, сеансах и билета, а также предоставлять сведения, запрошенные организатором.

Исходя из названия, цель работы – разработка автоматизированной системы управления, при помощи объектно-ориентированного подхода к программированию на C#.

Были поставлены следующие задачи, для достижения основной цели:

* провести системный анализ предметной области;
* составить техническое задание для решения задачи;
* спроектировать и разработать классы для хранения и представления данных и моделей;
* разработать пользовательский интерфейс согласно дизайн-макетам;
* выполнить реализацию приложения с использованием современных средств программирования, а также его тестирование и отладку.

Объектом исследования данной курсовой работы являются методы и технологии разработки программных продуктов с применения объектно-ориентированного программирования и языка программирования С#.

Предметом исследования данной курсовой работы являются методы, алгоритмы и приёмы разработки АСУ «Кинотеатр».

Информационной базой исследования является учебная литература по информатике и программированию, техническая документация по языку С# инструментальной среды MS Visual Studio 2017, а также SQL Server Management Studio(SSMS).Windows Forms.NET.

# ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

# Описание предметной области

Предметная область в разных сферах обозначает разное, начиная с «научной теории», и, заканчивая «логикой». В информационной среде предметная область – часть реального мира, которая подлежит изучению с целью автоматизации организации управления. В данной работе название предметной области – «Кинотеатр». Объектами являются лица, события или предметы, о которых ниже представлена информация.

В данной работе предметную область составляют:

Каждый кинотеатр характеризуется такими параметрами, как:

* Название кинотеатра
* Район города, в котором расположен кинотеатр
* Категория
* Вместимость

Каждый фильм характеризуется следующими параметрами:

* Название фильма
* Режиссёр
* Оператор
* Актёры, сыгравшие главные роли
* Жанр
* Производство
* Наличие призов кинофестивалей
* Продолжительность сеанса

Каждый сеанс характеризуется следующими параметрами:

* Дата
* Время

Сведения о билетах включают следующие параметры:

* Цена

Репертуар кинотеатров на месяц включает в себя:

* Фильмы
* Дата
* Название кинотеатра
* Цена билета
* Количество свободных мест на сеанс

Под субъектами предметной области данной АСУ будем понимать работника справочной службы, который может использовать следующие операции:

* Ввести новый фильм в репертуар или убрать фильм из репертуара
* Корректировать перечень фильмов, находящихся в прокате – добавлять новые фильмы, и снимать с проката
* Корректировать перечень кинотеатров – добавлять новые, и закрывать

Также справочная служба может вывести для себя следующую информацию:

* Репертуар кинотеатра
* Адрес и район кинотеатра
* Число свободных мест на данный сеанс в указанном кинотеатре
* Цена билетов на данный сеанс в указанном кинотеатре
* Жанр, производство и режиссёр данного фильма
* Какие фильмы имеют награды, когда и, в каких кинотеатрах они демонстрируются.
* В каких кинотеатрах в указанный день на указанных сеансах демонстрируется комедия?
* В каких кинотеатрах, и когда демонстрируются фильмы с участием указанного актера?

# 1.2 Выделение словаря предметной области

Кинотеатр – это общественное здание или его часть с оборудованием для публичной демонстрации кинофильмов.

Сеанс - показ или исполнение чего-нибудь, совершающиеся в определённый промежуток времени без перерыва.

Репертуар кинотеатра - совокупность произведений, идущих в кинотеатре.

Справочная служба - служба, предназначенная для предоставления пользователям услуг хранения информации и обработки запросов пользователей.

Прокат фильмов - это раздел на платформе, в котором доступны самые новые и актуальные фильмы, использование которых не входит ни в один из предусмотренных для абонирования пакетов контента.

# 1.3 Виды связей между объектами

Всего существует три вида взаимосвязей между объектами.

•связь «один к одному»;  
 • связь «один ко многим»;

• связь «многие ко многим».

# 1.3.1 Связь **«**один к одному**»**

**Связь «один к одному»** – это ситуация, при которой одному объекту таблицы А соответствует один объект таблицы Б, и наоборот.

В соответствии с рис.1, одному сеансу принадлежит один билет. Такой тип связи обычно характерен в одном классе или в одной таблице.

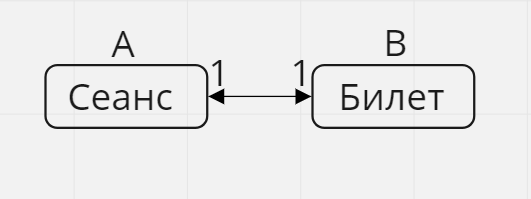


Рисунок 1 – Связь «один к одному»

# 1.3.2 Связь **«**один ко многим**»**

**Связь «один ко многим»** - это ситуация, при которой объекту А может принадлежать или же соответствовать несколько объектов Б, но объекту Б может соответствовать только один объект А.

В соответствии с рис.2, одному кинотеатру может соответствовать несколько фильмов, а одному фильму соответствует только один кинотеатр.

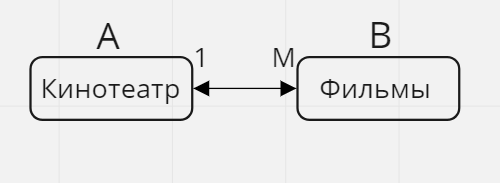


Рисунок 2 - Связь **«**один ко многим

# 1.3.3 Связь **«**многие ко многим**»**

**Связь «многие ко многим» -** это ситуация, при которой нескольким объектам из таблицы А может соответствовать несколько объектов из таблицы Б, и в тоже время нескольким объектам из таблицы Б соответствует несколько объектов из таблицы А.

В соответствии с рис.3, на один фильм может быть несколько сеансов, и на один сеанс может быть несколько фильмов.

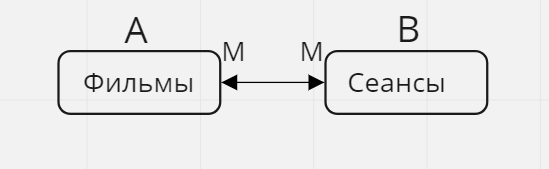


Рисунок 3 - Связь "многие ко многим"

# 1.4 Анализ существующих программных решений

Для примера похожий программных решений я взяла такие проекты, как «1С: Кинотеатр» и «1С:Театр».

Программное обеспечение «1С:Кинотеатр» (рис.4) создано для автоматизации ведения учета в кинотеатрах. Программа разделена на 4 основных раздела: предприятие, продажа билетов, расписание сеансов / анонсы, справочная информация. Рассмотрим немного подробнее каждый из них.

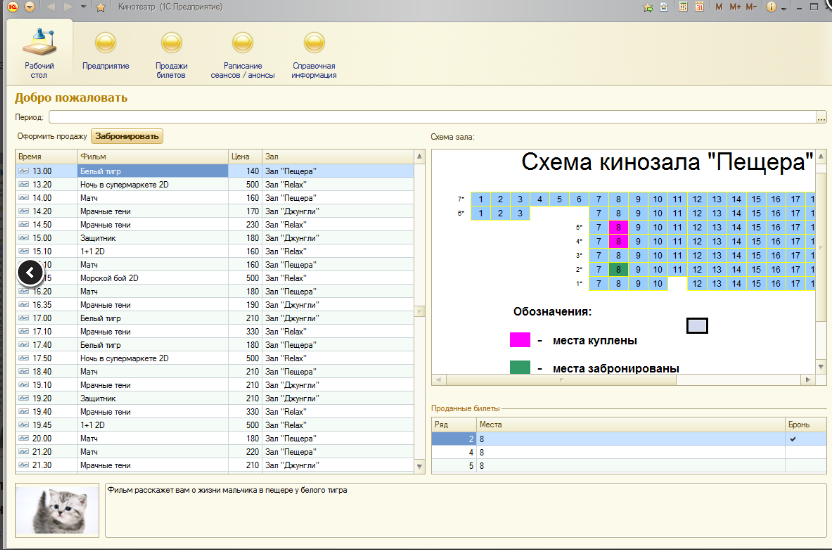


Рисунок 4 – «1С: Кинотеатр»

Предприятие. (рис. 5) В Данном разделе сосредоточена информация, касающаяся работы кинотеатра и залов. Состоит из: «Справочник-Кинозалы». Хранит в себе список кинозалов. Для каждого кинозала можно выбрать соответствующую ему схему зала.

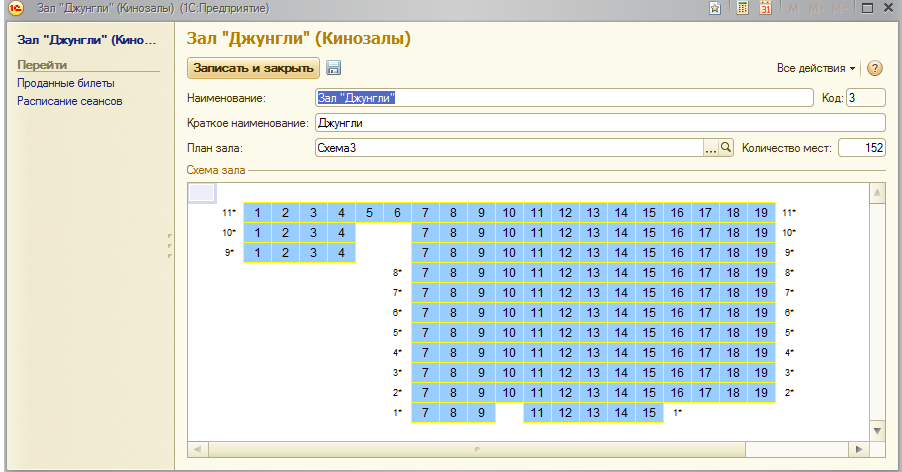


Рисунок 5 - Предприятие

2) Продажи билетов. (рис.6) Основной раздел, отвечающий за продажу, бронирование, распечатку билетов и чеков. Состоит из:

«Документ-Продажа билетов». Данный документ является основным для конфигурации. Через него осуществляется продажа / бронирование билетов, а также их распечатка.

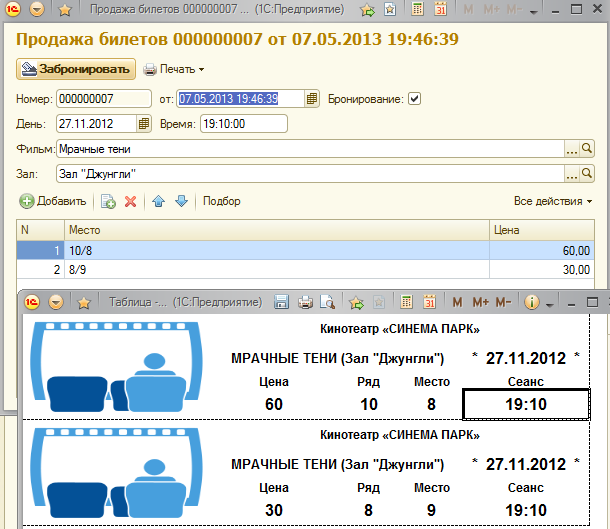


Рисунок 6 – Продажа билетов

3) Расписание сеансов. (рис.7) Раздел несет в себе справочную информацию о всех фильмах, анонсах и расписаниях. Содержит в себе объекты: «Справочник-Расписание сеансов». Хранит расписание сенасов для фильмов на выбранный период времени.

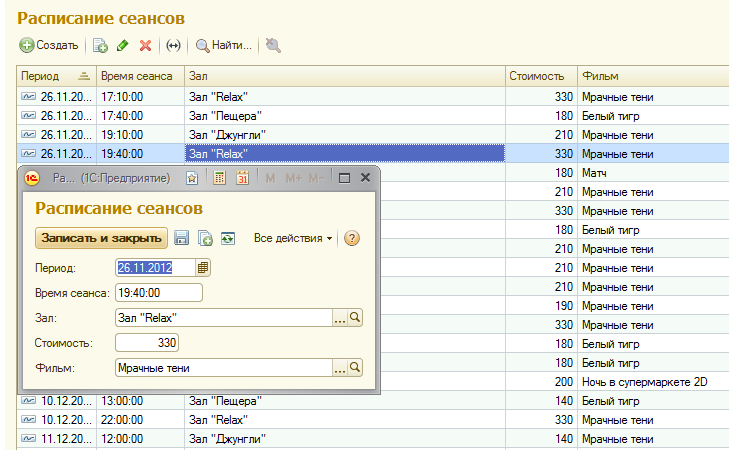


Рисунок 7 – Расписание сеансов

4) Справочная информация. (рис. 8) Содержится собранная из всех разделов справочная информация, облегчающая настройку конфигурации:

«Справочник-Кинозалы».

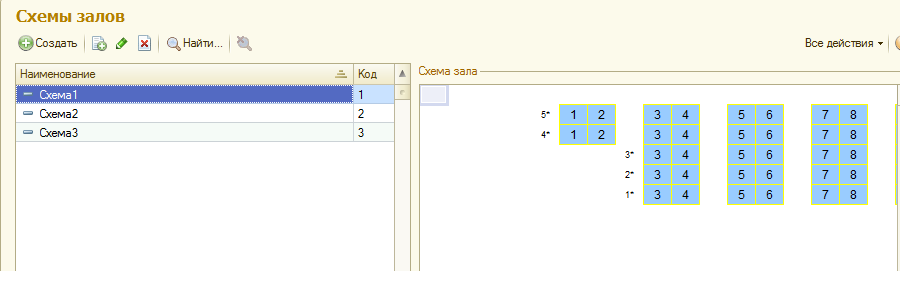


Рисунок 8 – Справочная информация

Перейдем к рассмотрению второго программного обеспечения – «1С:Театр». (рис. 9) Эта автоматизированная система управления может использоваться в управлении деятельностью театральных, концертных организаций и коллективов всех форм собственности.

Программный продукт «1C:Театр» автоматизирует группы процессов, которые охватывают всю деятельность театра, а именно:

* **Автоматизация процессов управления постановочной частью и художественными мастерскими театра;**
* **Автоматизация финансово-экономического управления деятельностью театра;**
* Автоматизация управления нормативно-справочной информацией;
* Автоматизация составления и корректировки репертуара, графика репетиций, графика гастролей;
* Автоматизация управления проведения спектаклей и репетиций;

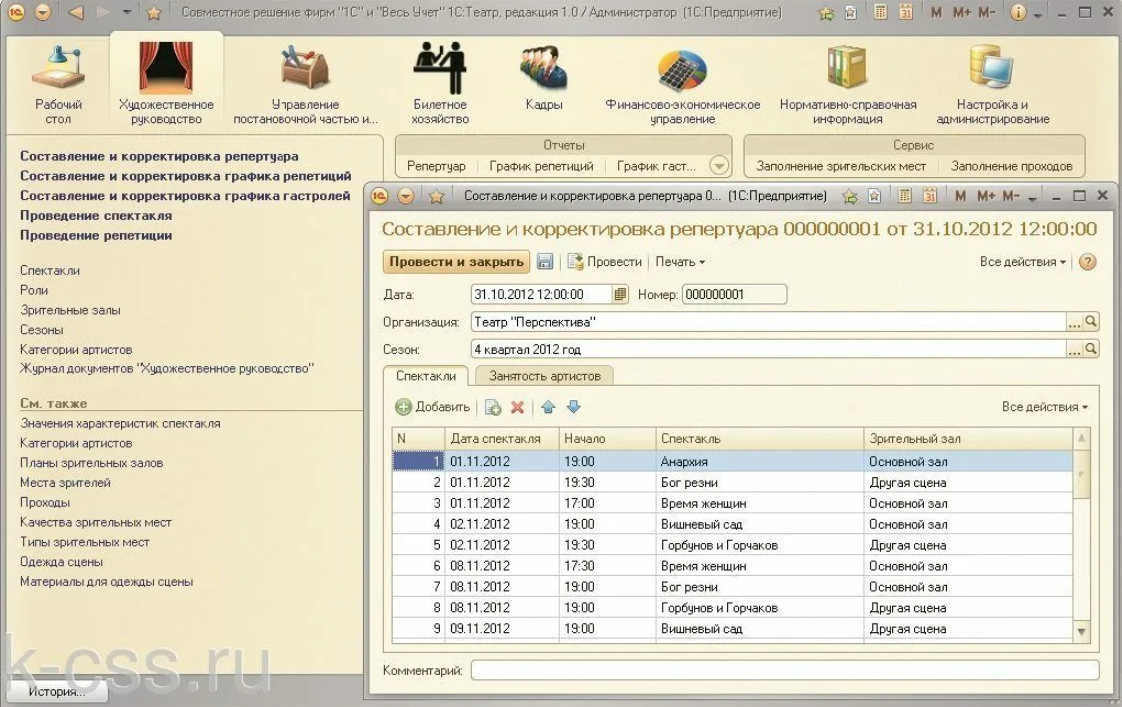


Рисунок 9 – «1С: Театр»

Программные средства, используемые для решения задач:

**Microsoft Visual Studio –** этоинтегрированная среда разработки программного обеспечения от компании Microsoft, предназначенная разрабатывать как консольные приложения, так и игры, и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP, а также веб-сайты, веб-приложения и веб-службы.

**Windows Forms.NET –**это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора вVisual Studio.

**SQL Server Management Studio (SSMS)** — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server.

Объектно-ориентированный язык С# создан специально для работы с фреймворком .NET (Windows Forms, WPF, ASP.NET, Xamarin).

C# относится к семье языков с С-подобным синтаксисом, из них его синтаксис наиболее близок к [C++](https://ru.wikipedia.org/wiki/C%2B%2B) и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, [события](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [переменные](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [свойства](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [обобщённые](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) типы и методы, [итераторы](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80), [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) с поддержкой [замыканий](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [LINQ](https://ru.wikipedia.org/wiki/Language_Integrated_Query), [исключения](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9), [комментарии](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B8_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) в формате [XML](https://ru.wikipedia.org/wiki/XML).

Переняв многое от своих предшественников языков [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [Delphi](https://ru.wikipedia.org/wiki/Delphi_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)" \o "Delphi (язык программирования)), [Модула](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B0-2), [Smalltalk](https://ru.wikipedia.org/wiki/Smalltalk" \o "Smalltalk) и, в особенности, [Java](https://ru.wikipedia.org/wiki/Java" \o "Java) — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается [множественная реализация интерфейсов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BD%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B8_%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%8)). C# изначально был создан языком со сложной архитектурой для решения сложных задач, и программирования больших программных обеспечений.

Объектно-ориентированные языки программирования отличаются от функциональных или процедурных языков. Функциональное или процедурное программирование идеально работает в простых программах. Функции аккуратно вложены друг в друга, взаимодействуют друг с другом, можно передать данные из одной функции в другую. Особенность таких [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) состоит в том, что задачи разбиваются на шаги и решаются шаг за шагом.

Используя процедурный язык, программист определяет языковые конструкции для выполнения последовательности алгоритмических шагов. К процедурным языкам программирования относятся языки 1С, Фортран, Си и т.д. В этих языках используется оператор GOTO. В его основе – процедуры и функции. Объектно-ориентированные языки основаны на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса. К объектно-ориентированным языкам кроме С# относятся такие языки, как PHP. PYTHON, JAVA, JAVA SCRIPT и т.д.

Основные принципы структурирования в случае ООП связаны с различными аспектами базового понимания предметной задачи, которое требуется для оптимального управления соответствующей моделью:

* **Инкапсуляция** позволяет открыть только ту часть функций и данных, которая нужна для внешних пользователей, а остальное спрятать внутри класса.
* Модификатор доступ**а** – ключевые слова в объектно-ориентированных языках, которые задают параметры доступа для классов, методов и прочих элементов. Они определяют степень доступности объектов класса извне. В языке С# реализуются следующие модификаторы доступа: public. private и т.д.
* **Наследование** позволяет делать производные структуры на основе базовых, тем самым давая возможность осуществлять повторное использование этих структур.
* «Абстрактный класс» – это родительский класс, который содержит общие абстрактные атрибуты. На его основе нельзя создать объект, но его может унаследовать родительский класс.
* Полиморфизм – принцип ООП, когда один фрагмент кода может работать с разными типами в C# реализуется с помощью перегрузок. П**олиморфизм**позволяет программировать на основе интерфейсов, у которых может быть несколько реализаций. Полиморфизм осуществляется в объектно-ориентированных-языках путём использования виртуальных методов, что является очень удобным и безопасным. Полиморфизм – это **ключевое свойство ООП** для построения архитектуры. Он позволяет сделать модуль независимым от конкретной реализации интерфейса. Этот принцип называется **инверсией зависимостей**, на котором основаны все плагинные системы.
* Инверсия зависимостей так называется, что она позволяет изменить направление зависимостей. Сначала начинаем писать в простом стиле, когда высокоуровневые функции зависят от низкоуровневых. Однако, когда программа начинает становиться слишком сложной, мы инвертируем эти зависимости в противоположную сторону: высокоуровневые функции теперь зависят не от конкретных реализаций, а от интерфейсов, а реализации теперь лежат в своих модулях. **Любая зависимость всегда может быть инвертирована.** В этом и есть достоинство ООП.
* Возможна также реализация инъекции или внедрения зависимости — процесс предоставления внешней зависимости [программному компоненту](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). Это ситуация, при которой объект отдаёт заботу о построении требуемых ему зависимостей внешнему, специально предназначенному для этого общему механизму.

# 1.5 Техническое задание

Согласно ГОСТ 34.602-89 «Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» на составление технического задания на автоматизированные системы ТЗ на АС является основным документом, определяющим требования и порядок создания (развития или модернизации - далее создания) автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка при вводе в действие.

В нормативном документе ГОСТ 34.602-89 ТЗ на АС содержит следующие разделы, которые могут быть разделены на подразделы:

1) общие сведения;  
 2) назначение и цели создания системы;  
 3) характеристика объектов автоматизации;

4) требования к системе;  
 5) состав и содержание работ по созданию системы;  
 6) порядок контроля и приемки системы;  
 7) требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;  
 8) требования к документированию;  
 9) источники разработки.

Техническое задание было разработано на основании требований настоящего ГОСТа.

# 1.5.1 Общие сведения

Задание на выполнение настоящего курсового проекта было выдано по дисциплине «Объектно-ориентированное программирование на языке С#».

Наименование: «Проектирование и разработка АСУ «Кинотеатр».

Целью разработки является демонстрация умений и навыков составления автоматизированной системы управления с использованием языка программирования С# в среде программирования Visual Studio 2017, с использованием SQL SERVER Management STUDIO 18 и Windows Forms.

# 1.5.2 Назначение АСУ

Данный программный продукт должен обеспечить организатору выставки удобную работу с данными. Она предназначена, чтобы добавлять, удалять, редактировать, сохранять данные и отвечать на запросы организатора.

# 1.5.3 Характеристика объекта автоматизации

Система предназначена для решения перечисленных ниже задач:

* Обеспечивать хранение сведений о кинотеатрах города, фильмах, которые в них демонстрируются, сеансах и билетах на эти сеансы;
* Корректировать перечень фильмов, находящихся в прокате – добавлять новые или снимать с проката;
* Корректировать перечень кинотеатров – добавлять или убирать;
* Отображать информацию о кинотеатрах, фильмах, сеансах и билетах;

Также по соответствующим запросам администратора справочной службы выдавать информацию:

* Репертуар кинотеатра;
* Адрес и район кинотеатра;
* Число свободных мест на данный сеанс в указанном кинотеатре;
* Цена билетов на данный сеанс в указанном кинотеатре;
* Жанр, производство и режиссер данного фильма;

Администратор справочной службы должен иметь возможность:

* Добавлять фильм в репертуар или убирать соответственно;
* Добавлять кинотеатр или убирать;
* Изменять сведения о работающих кинотеатрах;
* Изменять сведения о фильмах, которые идут в прокате;
* Получать информацию о сеансах на фильмы;
* Получать информацию о ценах на билеты на сеансы;
* Получать сгруппированный по районам города отчет за прошедший месяц о прокате фильмов;

Отчет должен содержать в себе:

* Название кинотеатра;
* Название фильма;
* Средняя цена билета;
* Доход по каждому кинотеатру;
* Доход по каждому району;
* Суммарный доход кинотеатров;

# 1.5.4 Требования к системе

На компьютере должны быть установлены:

* операционная система Windows с установленным Microsoft .NET Framework 2.0;
* дополнительно должны быть установлены Visual Studio 2017, SQL SERVER Menagment STUDIO 18. Windows forms.

Visual Studio 2017 может устанавливаться и запускаться в следующих операционных системах:

* Windows 10 версии 1507 и выше: Домашняя, Pro, для образовательных учреждений и Корпоративная (выпуски LTSC и S не поддерживаются).
* Windows Server 2016: Standard и Datacenter Windows 8.1 (с [обновлением 2919355](https://support.microsoft.com/kb/2919355)): Core, Профессиональная и Корпоративная.
* Windows Server 2012 R2 (с [обновлением 2919355](https://support.microsoft.com/kb/2919355)): Essentials, Standard, Datacenter.
* Windows 7 с пакетом обновления 1 (SP1) (с последними обновлениями Windows): Домашняя расширенная, Профессиональная, Корпоративная, Максимальная.
* Процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор.
* 2 ГБ ОЗУ; рекомендуется 4 ГБ ОЗУ (минимум 2,5 ГБ при выполнении на виртуальной машине).
* Место на жестком диске: до 130 ГБ свободного места в зависимости от установленных компонентов, обычно для установки требуется от 20 до 50 ГБ свободного места.
* Скорость жесткого диска: для повышения производительности установите Windows и Visual Studio на твердотельный накопитель (SSD).
* Видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей); для оптимальной работы Visual Studio рекомендуется разрешение WXGA (1366 на 768 пикселей) или более высокое.

# 1.5.5 Состав и содержание работ по созданию системы

Стадия разработки программного средства включает в себя следующие этапы:

* Получение исходной информации;
* Проектирование алгоритмов;
* Кодирование программы;
* Тестирование и отладка программы;
* Оформление программной документации с описанием пользовательского интерфейса

# 1.5.6 Условия эксплуатации

Данный программный продукт предполагается эксплуатировать в условиях, благоприятных для оператора ЭВМ.

**ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА**

**ПРОГРАММНОГО РЕШЕНИЯ**

# 2.1 Проектирование и разработка классов

 В ООП главным элементом является класс, включающий множество объектов с одинаковыми свойствами, операциями и отношениями. Класс имеет внутреннее (реализацию) и внешнее представление – интерфейс. В данной программе реализованы классы DatabaseManager, EnumTable, Program, Form1, Form2, FormHelpDesk, FormDataFilm, FormEditCinemaFilms, FormEditMovie, FormEditRepertoire, FormEditSession, FormEditTickets, FormFilm, FormRepertoire, FormViewAddress, FormViewFreeSeats, а также интерфейс dataGridView.

В классе DatabaseManager содержится вся логика работы программы и описаны все методы. Класс EnumTable содержит в себе список таблиц, которые используются в нашей базе данных. Интерфейс dataGridView позволяет преобразовать все наши таблицы для вывода на экран. Класс Program - это стандартный класс в VisualStudio, который отвечает за запуск первоначальной главной формы (Form1). Класс Form1 изначальная страница после запуска, на которой происходит момент входа в нашу программу.

Перейдем к рассмотрению остальных классов:

* Form2 – класс с тремя кнопками для выбора следующей страницы: фильмы, репертуар кинотеатров и справочная служба;
* FormFilm – важный класс, который с помощью интерфейса dataGridView и класса DatabaseManager позволяет загрузить нам таблицы данных о кинотеатрах, фильмах, которые демонстрируются в этих кинотеатрах, сеансах на фильмы и о билетах. Таблица «кинотеатры» содержит такие атрибуты, как: название кинотеатра, район города, категория и вместимость. Атрибуты таблицы «фильмы»: название фильма, режиссёр, оператор, актёры, жанр, производство, наличие призов кинофестивалей, продолжительность сеанса. У таблицы «сеансы» всего два атрибута: дата/время и количество свободных мест. И к таблице «билеты» прилагается атрибут – цена. Можно отфильтровать таблицы с кинотеатрами и фильмами. Также данный класс позволяет перейти на другие страницы для редактирования таблиц: «FormEditCinemaFilms», «FormEditMovie», «FormEditSession», «FormEditTickets». Эти классы позволяют нам добавлять, удалять или изменять данные таблиц. И предоставляет дальнейшее сохранение всех изменений в нашу базу данных. Также присутствует возможность вывода отчета за последний месяц.
* FormRepertoire – класс, который с помощью интерфейса dataGridView и класса DatabaseManager позволяет загрузить нам таблицу с данными о репертуаре кинотеатров. У этой таблицы присутствуют атрибуты: название кинотеатра, название фильма, цена за билет, количество свободных мест и сеанс. И в этом классе есть возможность перехода на страницу редактирования репертуара кинотеатров – «FormEditRepertoire». Этот класс аналогично прошлым предоставляет возможность добавления, удаления или изменения данных. И предоставляет дальнейшее сохранение всех изменений в нашу базу данных.
* FormDataFilm – класс, который с помощью интерфейса dataGridView и класса DatabaseManager, предоставляет администратору возможность увидеть по конкретному фильму его жанр, производство и режиссёра.
* FormViewAddress – класс, который с помощью интерфейса dataGridView и класса DatabaseManager, предоставляет администратору возможность посмотреть адрес у выбранного кинотеатра, а также вывести этот адрес документом.
* FormViewFreeSeats – класс, который с помощью интерфейса dataGridView и класса DatabaseManager, предоставляет администратору реализовать вывод данных, при необходимости, кинотеатра, фильмов, сеансов и билетов. Также возможна фильтрация по названию кинотеатра, и по названию или жанру фильма.

**2.2 Проектирование и разработка экранных форм пользовательского**

**интерфейса**

Для продуктивной работы реализован следующий пользовательский интерфейс. Данное программное средство можно разбить на модули, взаимодействующие друг с другом. На рис. 10 представлена структура пользовательского интерфейса.

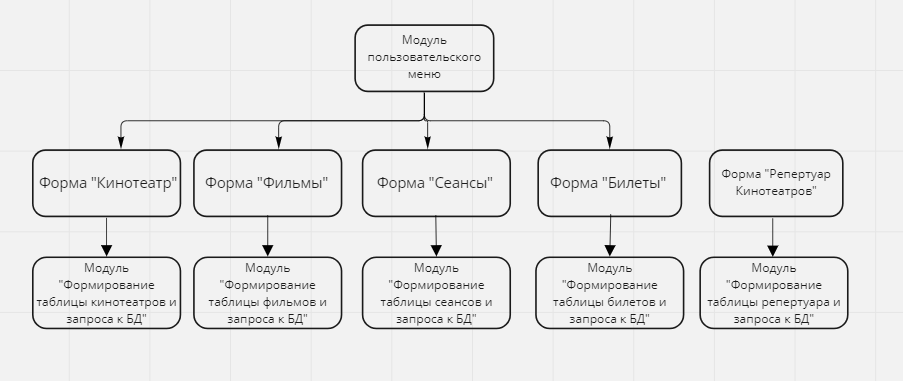


Рисунок 10 - Структура пользовательского интерфейса

# 2.2. 1 Форма получения информации

Форма получения информации представляет собой форму согласно рис. 11.

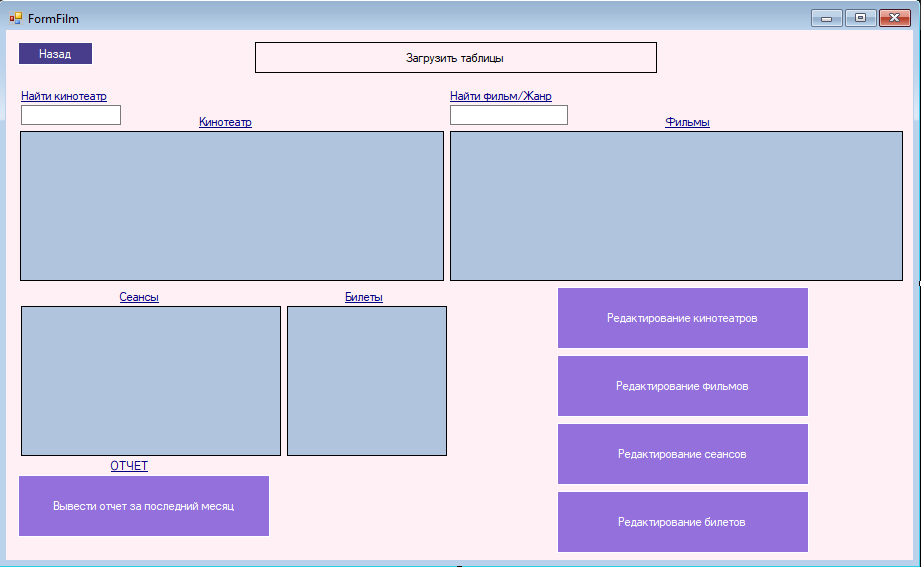


Рисунок 11 - Форма получения информации

На данной форме представлены следующие элементы:

* Кнопка «Загрузить таблицы» загрузит данные с каждой таблицы в окна;
* Окно «Кинотеатр» покажет данные из таблицы базы данных «Кинотеатр». В окошке выше можно найти конкретный кинотеатр по названию;
* Окно «Фильмы» покажет данные из таблицы базы данных «Фильмы». Можно так же найти фильм по названию или жанру.
* Окно «Сеансы» покажет данные из таблицы базы данных «Сеансы».
* Окно «Билеты» покажет данные из таблицы базы данных «Билеты».
* Кнопка «Вывести отчет за последний месяц» дает возможность вывести в виде word – документа отчет о доходах за последний месяц.
* Кнопки «Редактирование кинотеатров», «Редактирование фильмов», «Редактирование сеансов», «Редактирование билетов» переходят на страницы редактирования соответственно.

# 2.2. 2 Форма редактирования данных о кинотеатрах

Форма позволяет редактировать данные о кинотеатрах в таблице базы данных соответственно рис. 12

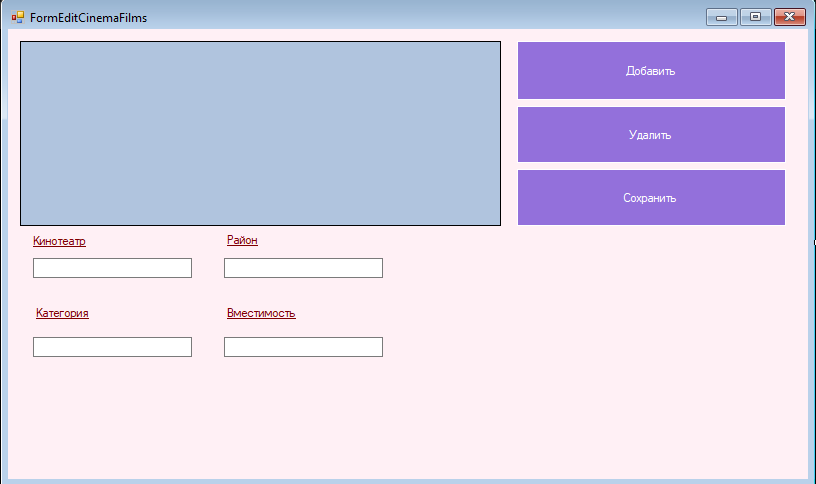


Рисунок 12 - Форма редактирования данных о кинотеатрах

На этой форме представлены следующие элементы:

* Окно для показа таблицы с данными.
* Поля для заполнения данных о кинотеатре: название, район, категория, вместимость.
* Кнопки для добавления, удаления и сохранения результата.

# 2.2. 3 Форма редактирования данных о фильмах

Форма позволяет редактировать данные о фильмах в таблице базы данных соответственно рис. 13.

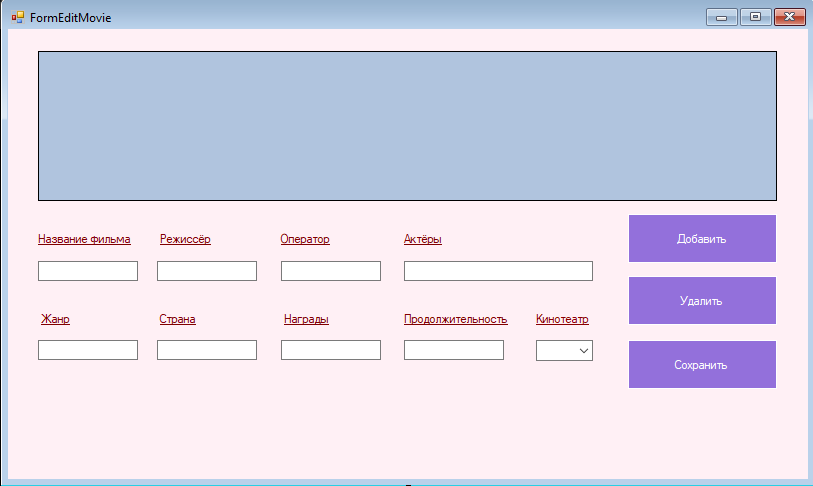


Рисунок 13 - Форма редактирования данных о фильмах

На этой форме представлены следующие элементы:

* Окно для показа таблицы с данными.
* Поля для заполнения данных о фильме: название, режиссёр, оператор, актёры, жанр, страна, награды, продолжительность и идентификатор кинотеатра, к которому будет относится фильм.
* Кнопки для добавления, удаления и сохранения результата.

# 2.2.4 Форма редактирования данных о сеансах

Форма позволяет редактировать данные о сеансах в таблице базы данных соответственно рис. 14.

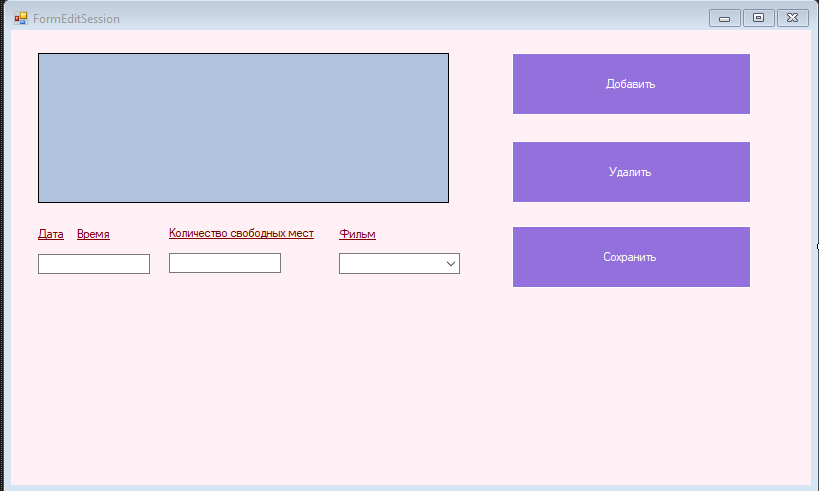


Рисунок 14 - Форма редактирования данных о сеансах

На этой форме представлены следующие элементы:

* Окно для показа таблицы с данными.
* Поля для заполнения данных о сеансе: дата/время, количество свободных мест и идентификатор фильма, на который заполняется сеанс
* Кнопки для добавления, удаления и сохранения результата.

# 2.2.5 Форма редактирования данных о билетах

Форма позволяет редактировать данные о билетах в таблице базы данных соответственно рис. 15.

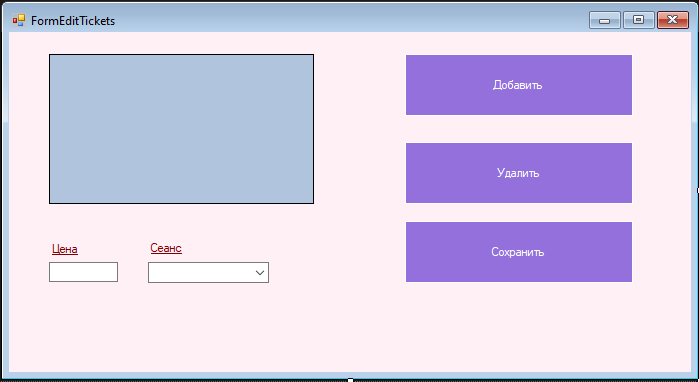


Рисунок 15 - Форма редактирования данных о билетах

На этой форме представлены следующие элементы:

* Окно для показа таблицы с данными.
* Поля для заполнения данных о билетах: цена и идентификатор сеанса.
* Кнопки для добавления, удаления и сохранения результата.

# 2.2.6 Форма получения данных

Форма позволяет загрузить данные из таблицы «Репертуар кинотеатров» соответственно рис. 16.

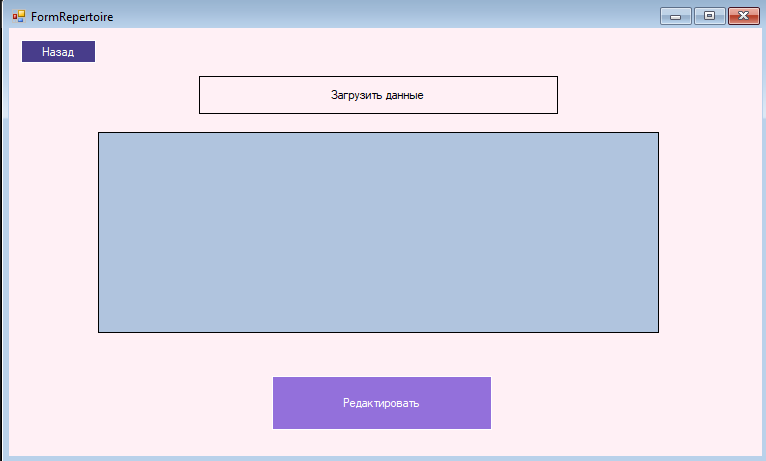


Рисунок 16 - Форма получения данных

На данной форме представлены следующие элементы:

* Кнопка «Загрузить таблицы» загрузит данные из таблицы в окно;
* Кнопка «Редактировать» переведёт пользователя на форму редактирования репертуара.

# 2.2.7 Форма редактирования данных о репертуаре кинотеатров

Форма позволяет редактировать данные о репертуаре в таблице базы данных соответственно рис. 17.

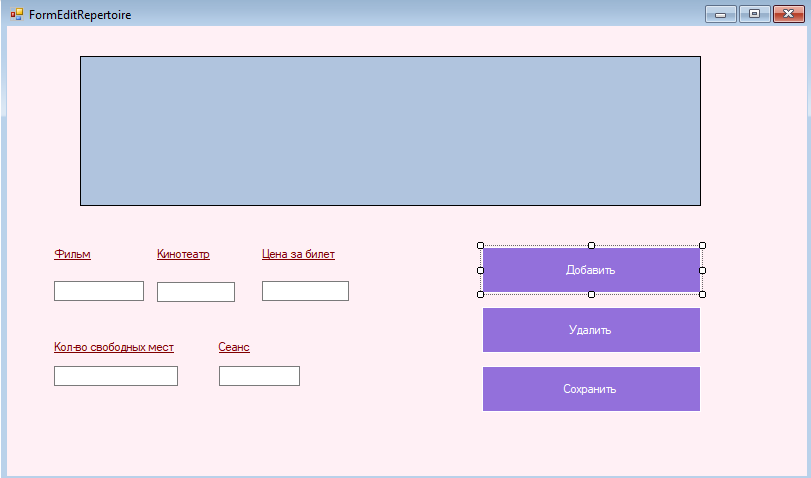


Рисунок 17 - Форма редактирования данных о репертуаре кинотеатров

На этой форме представлены следующие элементы:

* Окно для показа таблицы с данными.
* Поля для заполнения данных о репертуаре: фильм, кинотеатр, цена за билет, количество свободных мест, сеанс.
* Кнопки для добавления, удаления и сохранения результата.

# 2.2.8 Форма для просмотра конкретных данных о фильме

Форма позволяет получить данные о фильме соответственно рис. 18.

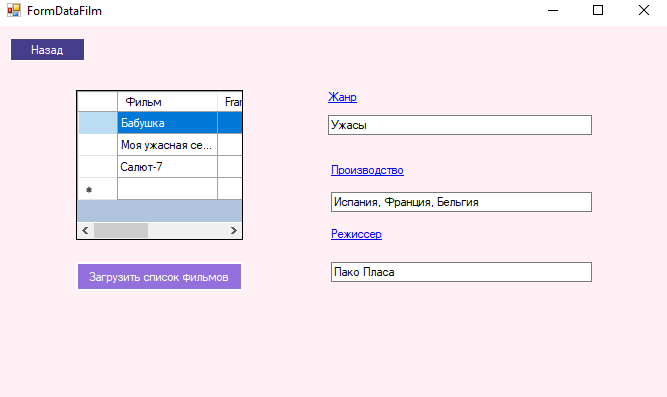


Рисунок 18 - Форма для просмотра конкретных данных о фильме

На этой форме представлены следующие элементы:

* Окно для отображения таблицы с фильмами.
* Кнопка «загрузить список фильмов» для показа таблицы.
* Поля, в которых представлена некоторая информация о выбранном фильме, а именно – жанр, производство и режиссёр.

# 2.2.9 Форма для просмотра адреса кинотеатра

Форма позволяет получить данные об адресе выбранного кинотеатра соответственно рис. 19.

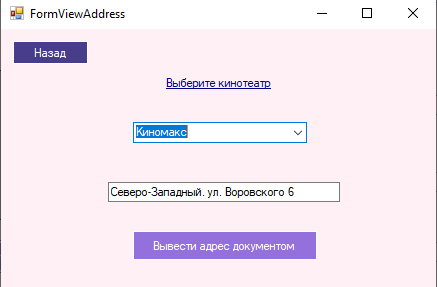


Рисунок 19 - Форма для просмотра адреса кинотеатра

На этой форме представлены следующие элементы:

* Выпадающий список для выбора кинотеатра.
* Поле для вывода адреса выбранного кинотеатра.
* Кнопка «Вывести адрес документом» позволяет напечатать адрес в документе Word.

# 2.2.10 Форма для просмотра данных по всем таблицам

Форма позволяет получить данные о всех таблицах базы данных при необходимости соответственно рис. 20.

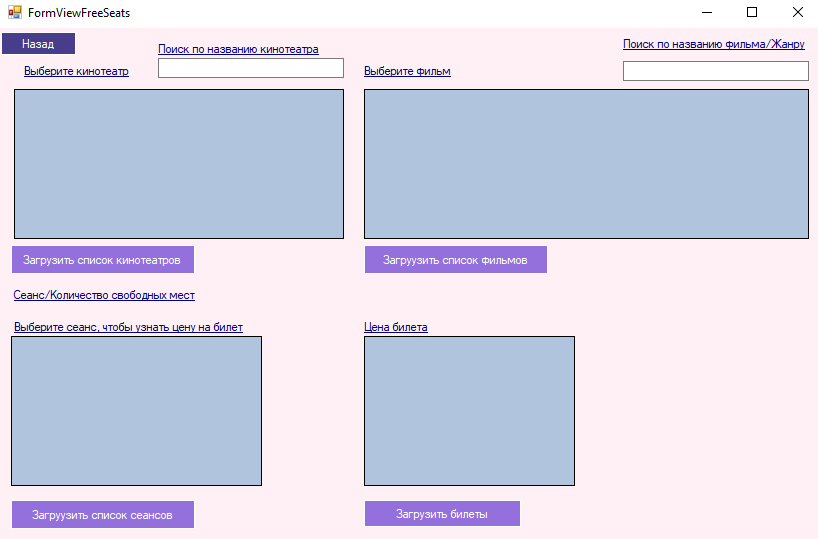


Рисунок 20 - Форма для просмотра данных по всем таблицам

На данной форме представлены следующие элементы:

* Кнопка «Загрузить список кинотеатров» загрузит данные с таблицы «Кинотеатр»;
* Окно «Кинотеатр» покажет данные из таблицы базы данных «Кинотеатр». В окошке выше можно произвести поиск по названию кинотеатра;
* Кнопка «Загрузить список фильмов» загрузит данные с таблицы «Фильмы»;
* Окно «Фильмы» покажет данные из таблицы базы данных «Фильмы». В окошке выше можно произвести поиск по названию/жанру фильма;
* При необходимости, есть кнопка «Загрузить список сеансов», чтобы увидеть данные по сеансам в таблице.
* Также, при необходимости, есть кнопка «Загрузить билеты», чтобы увидеть данные по билетам.

# 2.3 Описание и разработка базы данных

Для эффективной работы была реализована следующая база данных (рис. 21).

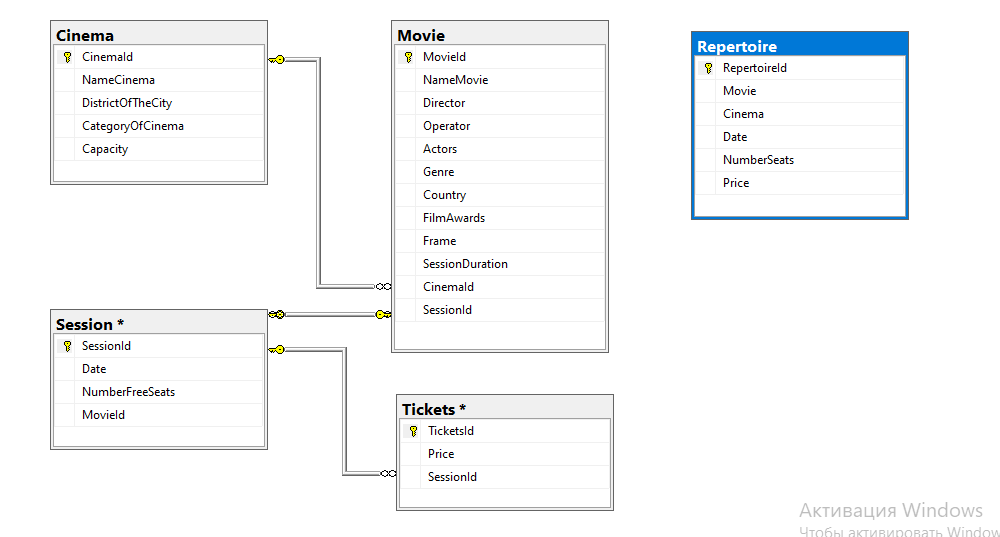


Рисунок 21 – Схема базы данных

Таблица Cinema содержит следующие атрибуты:

* ИД кинотеатра
* Название кинотеатра
* Район города
* Категория кинотеатра
* Вместимость

Таблица Movie содержит следующие атрибуты:

* ИД фильма
* Название фильма
* Режиссёр
* Оператор
* Актёры
* Жанр
* Страна
* Награды кинофестивалей
* Кадр
* Продолжительность
* ИД кинотеатра
* ИД Сеанса

Таблица Session содержит следующие атрибуты:

* ИД сеанса
* Дата/Время
* Количество свободных мест
* ИД фильма

Таблица Tickets содержит следующие атрибуты:

* ИД билета
* Цена
* ИД сеанса

Таблица Repertoire содержит следующие атрибуты:

* ИД репертуара
* Фильм
* Кинотеатр
* Дата
* Количество свободных мест
* Цена

Данная база данных позволяет хранить данные в отдельных таблицах. Таблицы взаимодействуют друг с другом путём установления в них отношений.

Отношения позволяют хранить данные в разных таблицах, связывая их отношениями данные из одной таблицы передаются в другую.

Например: выбрав фильм из таблицы Movie по атрибуту MovieId, который содержится в двух таблицах (рис. 15) одновременно мы можем выводить данные из одной таблицы в другую, всего лишь по одному атрибуту.

# 2.4 Краткое руководство пользователя по эксплуатации

# Системы

* 1. Открыть файл с именем Kursovaya\_V.1.exe
  2. В открывшемся графическом интерфейсе будет предложено меню возможных действий.
  3. Для показа всех таблиц с данными нажмите кнопку «Загрузить». Для добавления, удаления, изменения и сохранения данных о кинотеатре, нажмите на кнопку «Редактирование кинотеатров». После чего, откроется новая форма с понятным интерфейсом. Аналогично этому перейдите на другие формы для редактирования фильмов, сеансов и билетов. На этой же форме нажмите кнопку «Отчет» для вывода отчета о расходах и доходах в документе Word.
  4. В главном меню перейдите на вкладку «Репертуар кинотеатров». Загрузите таблицу, нажав на соответствующую кнопку. Затем перейдите по кнопке «Редактировать» на форму редактирования репертуара кинотеатров.
  5. В главном меню перейдя по кнопке «Справочная служба», Вы попадете на возможное получение более детальной информации.
  6. Нажмите на кнопку «Просмотр адреса выбранного кинотеатра», если требуется эта информация. Нажмите на кнопку «Просмотр данных о фильме» для просмотра деталей у каждого фильма. Нажмите на кнопку «Просмотр свободных мест» для просмотра деталей у каждой по отдельности таблицы.

# ЗАКЛЮЧЕНИЕ

Работая над данной курсовой работой, мною были изучены и закреплены основы объектно-ориентированного подхода к программированию на языке С# с использованием инструментальной среды MS Visual Studio 2017, реляционной базы данных SQL Server Management Studio 18, а также платформы пользовательского интерфейса Windows Forms.NET.Получилось достигнуть главной цели, а именно разработка и успешное функционирование системы управления. В ходе работы, также были достигнуты основные задачи, которые были представлены поэтапно. Укрепились навыки постановки задач, разработки алгоритмов и их решений, разработки технического задания, разработки и описания пользовательского интерфейса.

Исходя из всего вышеперечисленного, можно сказать, что проектирование и разработка подобных систем позволит увеличить качество и производительность работы, время и эффективность управления.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 34.602-89. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [Электронный ресурс] //Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/classifications>

2. Полное руководство по языку программирования С# 10 и платформе .NET 6 [Электронный ресурс] //Metanit.com. Сайт о программировании – Режим доступа <https://metanit.com/sharp/tutorial/>

3. Объектно-ориентированное программирование (C#) [Электронный ресурс] //Microsoft. Документация – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/fundamentals/tutorials/oop>

4. Объектно-ориентированное программирование [Электронный ресурс] //Код. Журнал Яндекс Практикума. – Режим доступа: <https://thecode.media/objective/>

5. Введение в ООП с примерами на С# [Электронный ресурс] // Tproger.– Режим доступа: <https://tproger.ru> 6. Объектно-ориентированное программирование [Электронный ресурс] //Skilbox Media – Режим доступа: <https://skillbox.ru/media/>

7. Объектно-ориентированное программирование [Электронный ресурс] //Библиофонд – Режим доступа: <https://www.bibliofond.ru/view.aspx?id=700569>

8. Предметная область и ee моделирование [Электронный ресурс] //РРепетитор. – Режим доступа: <file:///C:/Users/Home/Desktop/курсовая/Предметная%20область>

9. Предметная область [Электронный ресурс] //Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Предметная_область>

10. Класс (программирование) [Электронный ресурс] //Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Класс_(программирование)>

11. Объект (программирование) [Электронный ресурс] //Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Объект_(программирование)>

12. Модификаторы доступа (программирование) [Электронный ресурс] //Википедия. – Режим доступа: <https://translated.turbopages.org/proxy_u/en-ru.ru.f086b740-628229d2-ecb19e75-74722d776562/https/en.wikipedia.org/wiki/Access_modifiers>

13. Абстрактный класс (программирование) [Электронный ресурс] //Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Абстрактный_класс>

14. Интерфейс (программирование) [Электронный ресурс] //Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Интерфейс>

15. Модульное программирование [Электронный ресурс] //Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Модульное_программированиe>

16. Связи между таблицами базы данных [Электронный ресурс] //Хабр. – Режим доступа: <https://habr.com/ru/post/488054/>

17. NET для начинающих. Что такое среда .NET и как она работает? [Электронный ресурс] //Хабр – Режим доступа: <https://habr.com/ru/sandbox/13849/>

18. Новые возможности Visual Studio 2017. [Электронный ресурс] //Microsoft. Документация – Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017>

19. Что такое SQL Server Management Studio (SSMS). [Электронный ресурс] //Microsoft. Документация – Режим доступа: <https://docs.microsoft.com/ru-RU/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15&viewFallbackFrom=sql-server>