Министерство образования Саратовской области

Государственное автономное профессиональное образовательное учреждение Саратовской области

«Новоузенский агротехнологический техникум»

**Специальность: 09.02.07 Информационные системы и программирование**

**Курсовая работа**

**по МДК 01.01. «Разработка программных модулей»**

На тему:

**«АРМ работы кассира в кинотеатре»**

Выполнил: студент(ка) группы: П-21

Насыров Артур Канатович

(фамилия, имя, отчество)

09.06.2021

(подпись, дата)

Руководитель: преподаватель

Бекбулатов Р.Ж.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество) (оценка)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

**Содержание**

Введение

1. Теоретические основы проектирования информационных систем
2. Постановка задач

2.1 Обоснование выбора языка и сред разработки

1. Логическая и физическая модель
2. Проектирование программного средства
3. Тестирование программного средства

5.1 Добавление

5.2 Редактирование

5.3 Удаление

5.4 Сортировка

5.5 Закрыть

5.6 Покупка

1. Руководство по установке и использованию

Заключение

Список использованной литературы

Приложение

**Введение**

В современном мире активно развиваются и внедряются информационные технологии. На сегодняшний день уже во многих областях деятельности человека можно некоторую часть работы доверить компьютеру, так как они могут сделать эту работу не хуже человека, а в некоторых случаях лучше. Персональные компьютеры намного быстрее человека могут произвести математические вычисление, поиск информации, решение каких–либо задач. Но без программного обеспечения компьютеры бесполезны. Для каждой области деятельности человека необходимо свое программное обеспечение. Сегодня без этих помощников не обходиться ни одна компания, ни одно предприятие.

На сегодняшний день на рынке программного обеспечения отмечается устойчивый интерес к программным продуктам, способным обеспечить эффективное управление различной области компаний и предприятий. Большинство руководителей понимают, что для успешной организации производственного процесса необходимо применять современные информационные технологии.

В связи с глобальной информатизацией всех областей жизнедеятельности человека, а также с растущим уровнем потребности к программным продуктам, способным обеспечить эффективное управление производственным процессом, предоставляется возможность создания программного средства, решающего поставленную задачу, применяя современные информационные технологии.

Заданием на курсовое проектирование является разработка программного средства для автоматизации работы кассира в кинотеатре, которое должно обеспечивать ведение базы данных о фильмах и сеансах, билетах и брони на них, залах, персонале и пользователях, осуществлять формирование различных отчётов, выполнять продажу билетов на сеанс.

Автоматизация перечисленных процессов исключит возникновение ошибок при работе с данными, избавит от больших объемов различной документации и ускорит работу.

В результате работы над курсовым проектом будет разработано программное средство и составлена пояснительная записка, которая включает в себя шесть разделов, содержащих необходимую и достаточную информацию по использованию данного программного средства, подробное описание программы и вычислительной системы.

В заключении проанализируется созданное программное приложение, определится степень соответствия поставленной задачи и выполненной работы.

Также пояснительная записка будет содержать приложение и графическую часть, состоящую из диаграммы UML и классов.

**1.Теоретические основы проектирования информационных систем.**

Под системой понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов.

В информатике понятие «система» широко распространено и имеет множество смысловых значений. Чаще всего оно используется применительно к набору технических средств и программ. Системой может называться аппаратная часть компьютера. Системой может также считаться множество программ для решения конкретных прикладных задач, дополненных процедурами ведения документации и управления расчетами.

Добавление к понятию «система» слова «информационная» отражает цель ее создания и функционирования. Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач из любой области. Они помогают анализировать проблемы и создавать новые продукты.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации персонального компьютера. В крупных организациях наряду с персональным компьютером в состав базы информационной системы может входить мэйнфрейм или суперЭВМ. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

Необходимо понимать разницу между компьютерами и информационными системами. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом для информационных систем. Информационная система немыслима без персонала, взаимодействующего с компьютерами и телекоммуникациями.

Одной из основных проблем, которые приходится решать при создании больших и сложных систем любой природы, в том числе и программного обеспечения, является проблема сложности. Ни один разработчик не в состоянии выйти за пределы человеческих возможностей и понять всю систему в целом. Единственный эффективный подход к решению этой проблемы, который выработало человечество за всю свою историю, заключается в построении сложной системы из небольшого количества крупных частей, каждая из которых, в свою очередь, строится из частей меньшего размера, и так далее, до тех пор, пока самые небольшие части можно будет строить из имеющегося материала. Этот подход известен под самыми разными названиями, среди них такие, как «разделяй и властвуй» (divide et impera), иерархическая декомпозиция и др.

По отношению к проектированию сложной программной системы это означает, что ее необходимо разделить (декомпозировать) на небольшие подсистемы, каждую из которых можно разрабатывать независимо от других. Это позволяет при разработке подсистемы любого уровня иметь дело только с ней, а не со всеми остальными частями системы. Правильная декомпозиция является главным способом преодоления сложности разработки больших систем ПО. Понятие «правильная» по отношению к декомпозиции означает следующее:

количество связей между отдельными подсистемами должно быть минимальным (принцип «слабой связанности» — Low Coupling);

связность отдельных частей внутри каждой подсистемы должна быть максимальной (принцип «сильного сцепления» — High Cohesion).

Структура системы должна быть такой, чтобы все взаимодействия между ее подсистемами укладывались в ограниченные, стандартные рамки, т.е.:

каждая подсистема должна инкапсулировать свое содержимое (скрывать его от других подсистем);

каждая подсистема должна иметь четко определенный интерфейс с другими подсистемами.

**2. Постановка задач**

**2.1 Обоснование выбора языка и сред разработки**

Для реализации курсового проекта выбрана интегрированная среда разработки Microsoft Visual Studio. В целом среда имеет много достоинств:

обеспечение интуитивно понятной, расширяемой, унифицированной среды для языка, конструкторов и инструментальных средств.

Предоставление разработчикам набора модернизированных взаимодействующих элементов, соответствующих имеющимся у разработчиков навыкам.

Предоставление высокопроизводительных инструментальных средств для всех этапов жизненного цикла разработки – от определения требований и планирования до последующего сопровождения продукта.

В качестве языка программирования был выбран C#.

Преимущества языка программирования C#:

Подлинная объектная ориентированность (всякая языковая сущность претендует на то, чтобы быть объектом)

Компонентно-ориентированное программирование

Безопасный (по сравнению с языками C и C++) код

Унифицированная система типизации

Поддержка событийно-ориентированного программирования

«Родной» язык для создания приложений в среде .NET

Объединение лучших идей современных языков программирования: Java, C++, Visual Basic и др.

C# — [язык программирования](http://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), сочетающий [объектно-ориентированные](http://ru.wikipedia.org/wiki/%D0%9E%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) и [аспектно-ориентированные](http://ru.wikipedia.org/wiki/%D0%90%D1%81%D0%BF%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) концепции. Разработан в [1998](http://ru.wikipedia.org/wiki/1998_%D0%B3%D0%BE%D0%B4)—[2001 годах](http://ru.wikipedia.org/wiki/2001_%D0%B3%D0%BE%D0%B4) группой инженеров под руководством [Андерса Хейлсберга](http://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%B9%D0%BB%D1%81%D0%B1%D0%B5%D1%80%D0%B3,_%D0%90%D0%BD%D0%B4%D0%B5%D1%80%D1%81) в компании [Microsoft](http://ru.wikipedia.org/wiki/Microsoft) как основной язык разработки приложений для платформы [Microsoft .NET](http://ru.wikipedia.org/wiki/.NET_Framework). [Компилятор](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств, вроде [Visual Studio](http://ru.wikipedia.org/wiki/Microsoft_Visual_Studio).

C# относится к семье языков с [C-подобным синтаксисом](http://ru.wikipedia.org/wiki/C-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81), из них его синтаксис наиболее близок к [С++](http://ru.wikipedia.org/wiki/C%2B%2B) и [Java](http://ru.wikipedia.org/wiki/Java). Язык имеет [строгую статическую типизацию](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), поддерживает [полиморфизм](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC_%D0%B2_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0%D1%85_%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), [перегрузку операторов](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B9), указатели на функции-члены классов, атрибуты, [события](http://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B5), [свойства](http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%28%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%29), [исключения](http://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](http://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_%28%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%29) в формате [XML](http://ru.wikipedia.org/wiki/XML). Переняв многое от своих предшественников — языков [С++](http://ru.wikipedia.org/wiki/C%2B%2B), [Java](http://ru.wikipedia.org/wiki/Java), [Delphi](http://ru.wikipedia.org/wiki/Delphi_%28%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%29), [Модула](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B0-2_%28%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%29) и [Smalltalk](http://ru.wikipedia.org/wiki/Smalltalk) — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает [множественное наследование](http://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%28%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%29#.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) классов (в отличие от [C++](http://ru.wikipedia.org/wiki/C%2B%2B)).

C# разрабатывался как язык программирования прикладного уровня для [CLR](http://ru.wikipedia.org/wiki/Common_Language_Runtime) и, как таковой, зависит, прежде всего, от возможностей самой [CLR](http://ru.wikipedia.org/wiki/Common_Language_Runtime). Это касается, прежде всего, системы типов C#, которая отражает [FCL](http://ru.wikipedia.org/wiki/Base_Class_Library). Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем. (Однако эта закономерность была нарушена с выходом C# 3.0, представляющим собой расширения языка, не опирающиеся на расширения платформы .NET.) CLR предоставляет C#, как и всем другим [.NET](http://ru.wikipedia.org/wiki/.NET_Framework)-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, [сборка мусора](http://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0) не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на [VB.NET](http://ru.wikipedia.org/wiki/Visual_Basic_.NET), [J#](http://ru.wikipedia.org/wiki/Visual_J%E2%99%AF) .

**3.Логическая и физическая модель.**

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы – атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация – это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

Структура базы данных разрабатываемого программного средства включает таблицу. Её структура, и краткое описание приводится в таблице

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля | Описание поля |
| ID | int | Длинное целое | Код |
| Films | nvarchar | 50 | Название фильма |
| Data | nvarchar | 50 | Дата выхода |
| Director | nvarchar | 50 | Режиссер |
| Price | nvarchar | 50 | Цена |

**4. Проектирование программного средства**

Исходя из исследования предметной области задачи принято решение об организации программного средства в виде базы данных MS SQL SERVER. В программе предполагается возможность просмотра записей базы данных, возможность добавления, редактирования и удаления записей. На форме располагаются различные элементы управления, предназначенные для определенных задач. За компонентом Button будут закреплены разные действия для выполнения определенных функций, таких как: удаление, добавления и редактирования данных.

Button «добавление» – вызывает форму в которой пользователь может добавить в базу данных новую запись.

Button «редактирование» – вызывает форму в которой пользователь может изменить любую запись имеющуюся в базе.

Button «удаление» – удаляет выбранную пользователем запись из базы данных.

Button «Сортировка» - сортирует данные в таблице по ID

Button «Закрыть» - закрывает откладку программы по нажатию

Button «Купить билет» - выводит окно с сообщением что покупка билета была успешна

Также имеется стандартный элемент управления Textbox используется для ввода различных данных.

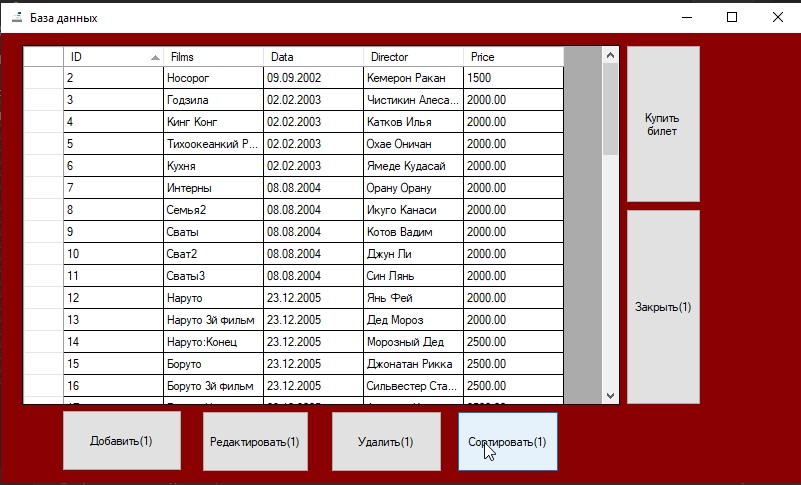
**5.Тестирование программного средства**

В процессе написания программного средства необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Тестирование программного средства – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами при условии, что функции отрабатывали на различных значениях.

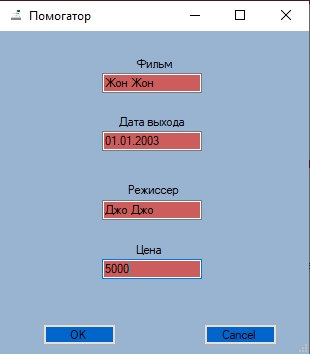
После входа в программу пользователю будет представлена вот такая форма

**5.1Добавление**

****Рисунок 5.1

Благодаря элементу DataGridView мы можем просмотреть и даже выбрать запись из базы данных.

При нажатии кнопки “Добавить запись” появляется вторая форма, форма для добавления

Рисунок 5.2

На данной форме расположено 4полей Textbox в которые нужно вписать данные для записи.

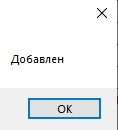
После заполнения формы мы можем нажать Ok, запись добавится, а форма добавления закроется и появится уведомление о том, что запись успешно добавлена.

Рисунок 5.3

И как мы видим добавленная нами запись появилась в таблице и в базе данных

**5.2 Редактирование**

Рисунок 5.4

Дальше мы рассмотрим Кнопку Изменения.

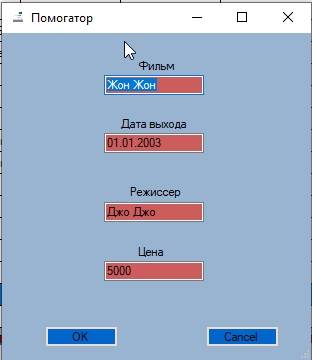
Пользователь может изменять любую строку в таблице. Она автоматически сохранится в базе данных

Рисунок 5.7

Тут мы можем поменять данные в записи. К примеру поставить цену фильма не 5000 а больше или меньше

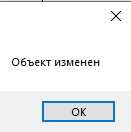
После нажимаем «ОК» и появляется такое окно

Рисунок 5.8

Это как Вы догадались значит что редактирование прошло успешно и теперь все данные которые ввел пользователь были сохранены

Как мы видим, все прошло успешно

 Рисунок 5.9

**5.3 Удаление**

Кнопка «Удалить» как уже ясно она удаляет выделенную вами строку из таблицы а так же из базы данных.

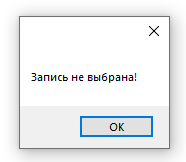
Если нажать на кнопку не выделив строку то появится такое окно

Рисунок 5.10

Сделав все правильно выделив строку и нажав на кнопку «Удалить»

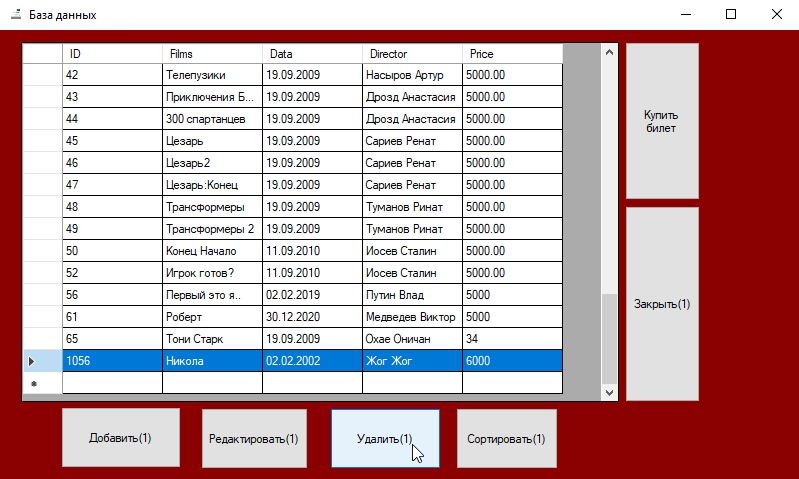


Рисунок 5.11

И нажав кнопку «Удалить» то мы увидим уже другое окно

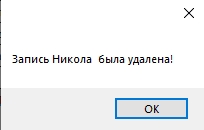


Рисунок 5.12

**5.4 Сортировка**

Кнопка «Сортировать» как уже ясно она сортирует таблицу по ID.

Покажу пару скриншотов не будем задерживаться на ней



Рисунок 5.13

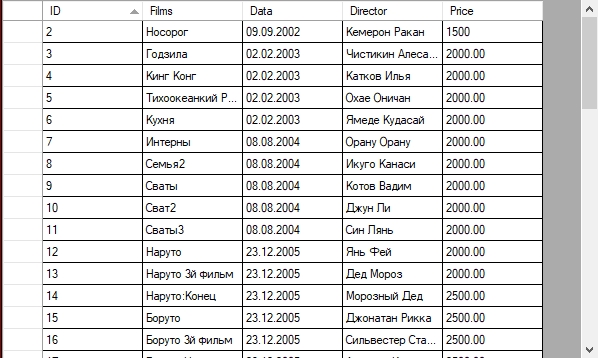


Рисунок 5.14

По факту на этом функционал этой кнопки закончен

**5.5 Закрыть**

Кнопка «Закрыть» просто закрывает откладку программы. Думаю Вы и так поняли как это

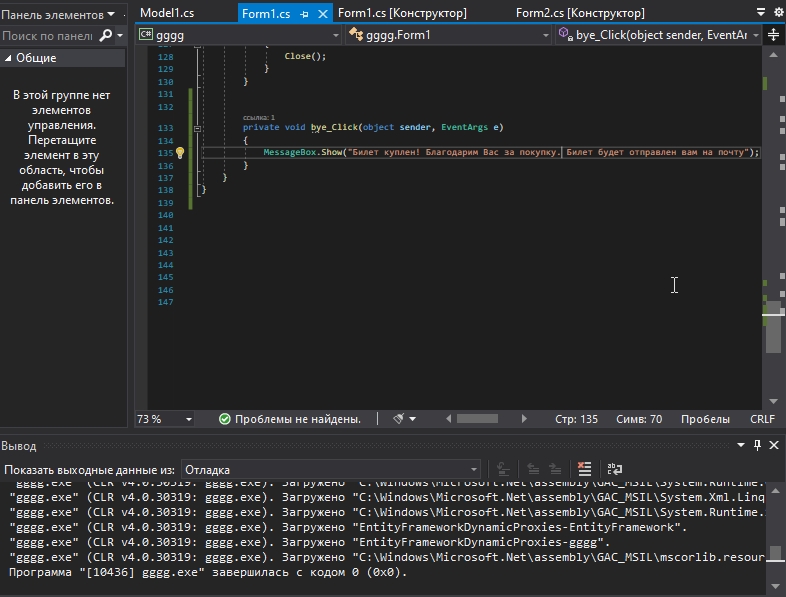
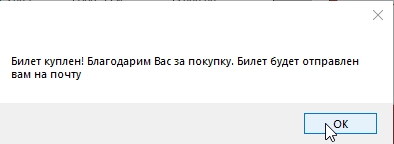


Рисунок 5.15

**5.6 Покупка билета**

Кнопка «Купить билет» - она просто выводит сообщение в окне что билет якобы куплен.

Рисунок 5.16

Простенько но со вкусом.

На этом Функционал данного приложения закончен.

**6. .Руководство по установке и использованию**

Для применения данного программного средства необходимы следующие технические требования:

- процессор Pentium 800 или выше;

- минимальный объем оперативной памяти 128 Мб;

- операционная система Windows XP и выше;

-пакет обновлений Framework v4.0

-рекомендуется монитор типа VGA или с лучшей

-разрешающей способностью;

-клавиатура

-мышь.

Компьютер должен работать под управлением операционной системы, начиная с Windows XP и выше. Наиболее удобной операционной системой для проведения испытаний является Windows 7, так как она ориентирована на максимальное использование всех возможностей ПК, сетевых ресурсов и обеспечение комфортных условий работы.

Программа имеет небольшой размер, оптимизирована для быстрого запуска и минимальной загрузки системы. Доступный дружественный интерфейс обеспечивает наглядность простоты в управлении программы и не содержит сложных настроек.

Для начала работы с программным средством необходимо запустить файл «Kursovaya.exe», размер которого составляет 150 КБ. Для функционирования программного средства необходимо установить платформу .NET Framework 4.0. Программное средство не нуждается в установке.

**Заключение**

В рамках курсового проектирования было разработано «Программное средство для автоматизации работы кассира в кинотеатре».

Программное средство имеет ряд достоинств: простой и понятный интерфейс, небольшой объем памяти, занимаемый приложением на различных носителях информации, удобно для быстрой обработки информации.

В программе реализованы такие задачи, как: добавление данных, изменение и удаление.

Программа реализована в полном объеме и в соответствии с заданными требованиями. Полностью отлажена и протестирована. Поставленные задачи выполнены.

В программном средстве, благодаря среде разработки Microsoft Visual Studio 2019, был разработан удобный, понятный и простой в использовании интерфейс.

Программа реализована полностью в соответствии с поставленной задачей курсового проекта. Приложение было протестировано и отлажено.

Проект был реализован с помощью среды разработки Microsoft Visual Studio 2019, с использованием языка C# и системы управления базами данных Microsoft SQL Server 2021.

Программное средство готово к практическому использованию. Данное приложение может быть дополнено и модернизировано.

Список литературы

1 Багласова Т.Г. Методические указания по выполнению дипломного проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий».– Мн.: ТБП, 2003

2 Багласова Т.Г. Методические указания по оформлению курсовых и дипломных проектов.- Мн.: ТБП, 2006

3 Кунец И.Е. Объектно – ориентированный анализ и проектирование. Работа в среде Rational Rose. Методическое пособие для учащихся ССУЗ по специальности 2-40 01 01 «Программное обеспечение информационных технологий».– Мн.: ТБП, 2006

4 МаккиА.И. VisualStudio 2010 для профессионалов. –СПб.: Питер, 2010

5 КарповР.Н. Microsoft Office 2007 – Мн.: Бином, 2007

6 Томров А.К. RationalRose для начинающих. – Мн.: Бином, 2005

7 Хернандес М. Вьесках. Дж. SQL. – СПб.: Питер, 2001

8 Петцольд Ч. Программирование для MicrosoftWindows на C# – М.: Издательско-торговый дом «Русская редакция», 2002

9 Холецкий В.И. Windows7операционная система. – СПб.: Питер, 2011

10 ГОСТ 19.301-2000 ЕСПД. Программа и методика испытаний

11 ГОСТ 19.401-2000 ЕСПД. Текст программы

12 ГОСТ 19.402-2000 ЕСПД. Описание программы

13 ГОСТ2.105-95 ЕСКД. Общие требования к текстовым документам

**Приложение**

**Листинг.**

На данном этапе были выделены и реализованы функции, позволяющие решить задачи, обеспечивающие достижение цели курсового проекта.

Связь с базой данных и её отображение на DataGridView осуществляется автоматически при запуске приложения. Код связи представлен ниже.

ModelBD.Model1 connect = new ModelBD.Model1();

public Form1()

{

InitializeComponent();

connect.Sport.Load();

dataGridView1.DataSource = connect.Sport.Local.ToBindingList();

}

Функция Добавления записи обрабатывается по нажатию кнопки “Добавить”

И вызывается методом “add1\_Click ”. При выполнении данной функции будут формироваться запросы, добавляться в базу информация о новой записи. Текст данной функции представлен ниже.

private void add1\_Click(object sender, EventArgs e)

{

Form2 form = new Form2();

DialogResult result = form.ShowDialog(this);

if (result == DialogResult.OK)

{

ModelBD.Client cliadd = new Client();

cliadd.FirstName = form.textBox1.Text;

cliadd.LastName = form.textBox2.Text;

cliadd.Phone = form.textBox3.Text;

cliadd.GenderCode = form.comboBox1.Text;

connect.Client.Add(cliadd);

connect.SaveChanges();

MessageBox.Show("Добавлен");

}

}

Функция Изменения записи обрабатывается по нажатию кнопки “Изменить” И вызывается методом “ redacted\_Click ”, после выбора записи, которую хотим изменить. Текст изменения приведен ниже.

private void redakted\_Click(object sender, EventArgs e)

{

Form2 formedit = new Form2();

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

int id = 0;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out id);

Client clientedit = connect.Client.Find(id);

formedit.textBox1.Text = clientedit.FirstName;

formedit.textBox2.Text = clientedit.LastName;

formedit.textBox3.Text = clientedit.Phone;

formedit.comboBox1.Text = clientedit.FirstName;

DialogResult resultedit = formedit.ShowDialog(this);

if (resultedit == DialogResult.OK)

{

clientedit.FirstName = formedit.textBox1.Text;

clientedit.LastName = formedit.textBox2.Text;

clientedit.Phone = formedit.textBox3.Text;

clientedit.GenderCode = formedit.comboBox1.Text;

connect.SaveChanges();

dataGridView1.Refresh();

MessageBox.Show("Объект изменен");

}

}

}

Функция удаления записи из базы данных происходит по нажатию кнопки “удалить” И вызывается методом “ delete\_Click”, после выбора записи, которую хотим удалить. Текст этой функции ниже.

private void delete\_Click(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int index = dataGridView1.SelectedRows[0].Index;

int id = 0;

bool converted = Int32.TryParse(dataGridView1[0, index].Value.ToString(), out id);

if (converted == true)

{

Client Clientdel = connect.Client.Find(id);

connect.Client.Remove(Clientdel);

connect.SaveChanges();

string buff = Clientdel.FirstName;

MessageBox.Show("Запись " + buff + " была удалена!");

}

}

else

{

MessageBox.Show("Запись не выбрана!");

}

}

Функция «Сортировка» происходит по нажатию кнопки «sort\_Click»

private void sort\_Click(object sender, EventArgs e)

{

{

dataGridView1.Sort(dataGridView1.Columns[0], x ? ListSortDirection.Ascending : ListSortDirection.Descending);

x = !x;

}

}

Функция «Закрыть» происходит по кнопке «close\_Click». Она просто закрывает откладку программу

private void close\_Click(object sender, EventArgs e)

{

{

Close();

}

}

Функция «Купить билет» это просто сообщение в окне. Происходит по кнопке «bye\_Click»

private void bye\_Click(object sender, EventArgs e)

{

MessageBox.Show("Билет куплен! Благодарим Вас за покупку. Билет будет отправлен вам на почту");

}