Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

филиал «Минский радиотехнический колледж»

ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ РАБОЧЕГО МЕСТА СОТРУДНИКА КИНОТЕАТРА

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Базы данных и системы управления базами данных»

КП32492.015102.081ПЗ

Руководители / Виничук О.Н./

Разработчик / Науменко А.А./

МИНСК 2016

Содержание

Введение3

1 Постановка задачи4

1.1 Описание предметной области4

1.2 Цели и задачи проектирования8

2 Проектирование задачи9

2.1 Проектирование схемы базы данных9

2.2 Алгоритм реализации задачи12

2.3 Описание сред реализации задачи13

3 Программная реализация17

3.1 Физическая структура базы данных17

3.2 Структура приложения 20

4 Тестирование24

5 Описание применения 26

Заключение 31

Список использованных источников 32

Приложение А Текст программы 33

Приложение Б Диаграмма вариантов использования 61

Приложение В Диаграмма деятельности62

Приложение Г Блок-схема63

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

*КП 32492.015102.081 ПЗ*

Разраб.

Науменко А.А.

Провер.

Виничук О.Н.

Реценз.

Н. Контр.

Утверд.

Программное средство автоматизации рабочего места сотрудника кинотеатра

Пояснительная записка

Лит.

Листов

63

МРК

У

Введение

Темой данного курсового проекта является разработка программного средства автоматизации рабочего места сотрудника кинотеатра.

Наличие автоматизированных касс-киосков помогает зрителю получить у кассира всю необходимую информацию прямо перед покупкой билета, самостоятельно выбрать удобное место приобретения билета, что сокращает очереди в кассах мест проведения мероприятий. Данный курсовой проект должен представить принципиально новую возможность бронирования билетов, а также позволить распечатывать билеты на любом принтере и значительно экономя время выкупа билетов.

Целью курсового проекта является разработка автоматизированной системы продажи билетов на сеансы в кино на языке SQL. При создании программного средства нужно учитывать специфику и особенности деятельности по продажам и распространению билетов в сфере культуры и искусства на территории Беларуси, современные тенденции работы информационных автоматизированных систем продаж.

Задачи курсового проекта включают разработку базы данных, предоставление клиенту необходимой информации, возможность ее редактирования, осуществлять поиск в программе.

Первая глава курсовой работы посвящена предпроектной стадии, то есть изучению объекта автоматизации, обоснованию необходимости создания системы комплексной автоматизации, формированию альтернативных концепций автоматизации, а также выбору наиболее подходящей.

Объектом курсового проекта является автоматизация процесса, что позволяет повысить производительность, сократить время выполнения процесса, снизить вероятность ошибок.

Предмет курсового проекта – продажа билетов на сеансы в кино.

Во второй и третьей главах описано техническое проектирование системы, представлены документы для приобретения необходимых технический, программных и аппаратных средств, расписана потребность в обслуживании автоматизированной системы, показано описание алгоритма комплексной автоматизации деятельности кинотеатра.

В третьей и четвертой главах данной курсовой работы представлена рабочая документация по программному средству.

1 Постановка задачи

Целью данной курсовой работы является создание информационной системы кинотеатра для хранения и обработки данных о совершенных заказах.

Процесс автоматизации кинотеатра заключается в разработке и внедрении программных продуктов для продажи и автоматизированного учета билетов. Система должна позволить централизовать управление всеми процессами, связанными с принятием и выполнением заказа, позволить руководителю своевременно получать достоверную информацию и, исходя из этого, строить правильную экономическую политику предприятия.

Организация применения автоматизированных систем управления в кинотеатрах очень важна, а в условиях экономического кризиса информационные технологии могут стать серьезным инструментом для оптимизации управления, сокращения расходов и дать неоспоримые конкурентные преимущества на рынке.

Разрабатываемое программное средство (ПС) предназначено для упрощения работы сотрудника кинотеатра, систематизации продажи билетов, предоставления клиенту информации о кинотеатре, сеансе, возможности бронирования мест, а также редактирования и осуществления поиска информации в программе. ПС должен содержать базу данных, в которую заносится и редактируется информация о сеансах, брони билетов и их стоимости.

* 1. Описание предметной области

Автоматизация бизнес процессов – залог эффективного управления. Автоматизация бизнес процессов приводит к уменьшению рутинных операций, позволяет намного быстрее обслуживать клиентов, дает больше возможностей для контроля, бизнес-процессы становятся более «прозрачными». Существенно улучшается работа по планированию закупок и поставок и другие плюсы. Все это в свою очередь существенно увеличивает рост прибыли, товарооборота и выручки, сокращает издержки. Уменьшение рутинных операций способствует значительно способствует сокращению затрат на персонал.

В настоящее время можно проследить тенденции к расширению рынка предоставления всевозможных развлекательных услуг. К данной среде относятся и кинотеатры. Кинотеатр – общественное здание или его часть, оборудованные для публичной демонстрации кинофильмов. Главное помещение кинотеатра – зрительный зал с экраном большого размера и системой воспроизведения звука, состоящей из нескольких громкоговорителей, обеспечивающих объемный звук.

Количество кинотеатров неумолимо увеличивается как в больших городах, население которых переваливает за миллион, так и в городах поменьше.

Сегодня используются автоматизированные системы, которые  представляют собой комплекс программ, обеспечивающих учет билетов на киносеансы. Они разработаны для увеличения объема продаваемых билетов, производительности кассиров, уменьшения и облегчения труда бухгалтеров и менеджеров.

Рабочее место кассира - это компьютер, соединенный со специальным принтером, на котором распечатываются билеты. Кассир в зависимости от имеющихся у него прав может продать билет на любой заведенный сеанс или забронировать часть билетов.

В век информационных технологий с каждым годом количество автоматизированных продуктов возрастает. В связи с этим можно привести несколько аналогов программных средств, предоставляющих услуги бронирования билетов.

Билетная система «Чудобилет», которая представлена на рисунке 1.1, обладает следующими функциями:

## возможность работы с клубными картами;

* On-line продажи и бронирования билетов;
* управление ценами мест и продажей билетов;
* создание афиши.



Рисунок 1.1 - Билетная система «Чудобилет»

Автоматизация касс на платформе "1С:Предприятие 8" представлена на рисунке 1.2. Она обладает следующими возможностями:

* предоставление подробной информации о сеансах;
* удобная работа со схемами залов;
* управление ценами мест и продажей билетов;
* печать билетов на бланках строгой отчетности;
* учёт бланков строгой отчетности;
* ведение учёта билетов и мест, переданных на реализацию другим распространителям;
* специализированная отчётность (накладные на реализацию билетов, накладные на возврат);
* широкие возможности обмена информацией с другими системами, с интернет-сайтами;
* масштабируемость и управляемость (широкие возможности адаптации программы под конкретные задачи).

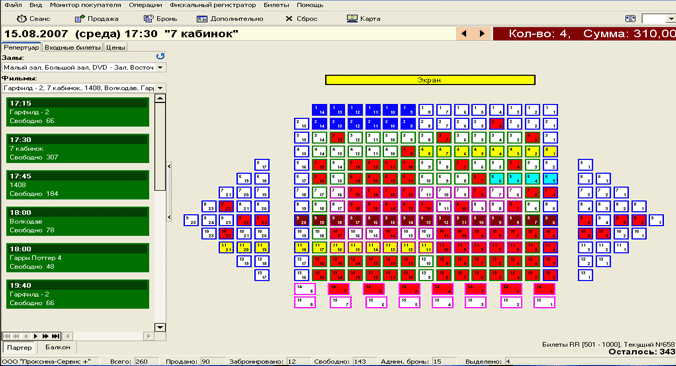


Рисунок 1.2 - Автоматизация касс на платформе «1С:Предприятие 8»

Система "Премьера" представлена на рисунке 3.

Возможности программы:

* Internet-справочная по расписанию сеансов и наличию мест;
* бронирование билетов через Internet;
* взаимодействие кассы и менеджерского компьютера через Internet;
* создание схемы зрительного зала;
* построение отчетов.

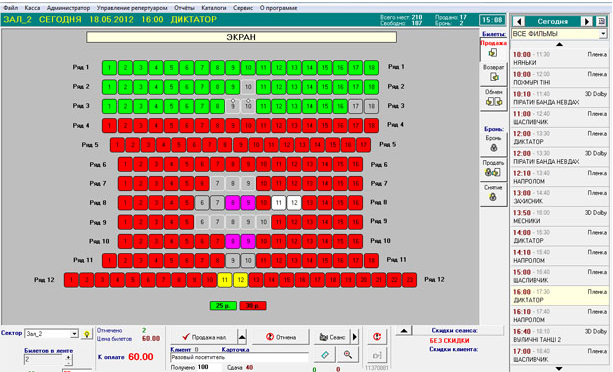


Рисунок 1.3 - Система «Премьера»

Рассмотрев аналоги разрабатываемого программного продукта, можно сказать, что разрабатываемый продукт может предоставить более лёгкий интерфейс и широкие возможности для пользователя, а именно: предоставление клиенту возможность бронировать места, осуществлять поиск информации в программе и возможность ее редактирования. Из недостатков системы можно выделить отсутствие возможности работать в On-line режиме.

Как и любая компьютерная программа, данное программное средство обрабатывает входные данные и выводит результат в виде выходных данных.

Входными данными разрабатываемой ПС являются:

* запросы пользователя: на поиск, бронирование, редактирование БД, просмотр справочной информации;
* критерии поиска (название, дата, время начала спектакля);
* информация о бронировании (название, время, дата начала спектакля, серийный номер билета, забронированное место, название театра, стоимость билета);

Выходными данными разрабатываемого ПС являются:

* информация отображаемая на экран по запросу пользователя;
* результаты поиска;
* изменённая БД.
* Постоянные данные:
* БД билетов со структурой: театр, спектакль, даты и время проведения, информация о бронировании;
* схемы зрительного зала.

В данной курсовой работе поставлена задача создания программного средства автоматизации рабочего места сотрудника кинотеатра. Поэтому при её выполнении было проведено исследование данной предметной области. В рамках данной курсовой работы реализуется проектирование и разработка учебной базы данных по предметной области. Следовательно, полученная БД не будет отражать всю бизнес-логику данной предметной области, а ограничиться лишь её частью.

1.2 Цели и задачи проектирования

Целью курсового проекта является разработка автоматизированной системы продажи билетов на сеансы в кино на языке SQL, которое позволяет:

* предоставить информацию клиенту о кинотеатре, сеансах, стоимости мест;
* бронировать места;
* редактировать информацию;
* осуществлять поиск информации в программе по критериям (название, время, дата начала сеанса, серийный номер билета, забронированное место, стоимость билета);

Разрабатываемая ПС должна быть обеспечена простой формой интерфейса:

* содержать краткие описания элементов меню;
* должен быть простым в использовании.

Должна имеется справочная система с кратким описанием, что будет способствовать быстрому освоению разрабатываемой ПС.

Разработать меню содержащие следующие пункты:

* сеансы;
* покупка билета;
* схема зрительного зала;
* информация о театре.

Необходимо разработать документацию программного средства, диаграмму вариантов использования и диаграмму деятельности.

2 Проектирование задачи

2.1 Проектирование схемы базы данных

На этапе проектирования схемы базы данных необходимо разработать логическую модель для физической реализации базы данных. Удачная разработка базы данных обеспечивает простоту ее поддержания. Данные следует сохранять в таблицах, причем каждая таблица должна содержать информацию одного типа.

Исходя из описания предметной области организацию данных можно представить набором следующих отношений:

* фильмы;
* сеансы;
* билет;
* место;
* состояние;
* возрастное ограничение;
* жанры;
* страна;
* продюсер;
* формат;
* актеры.

Далее следует идентифицировать связи.

Связь – это некоторая ассоциация между двумя отношениями. Связи позволяют по одному отношению находить другие отношения, связанные с ним. Они бывают следующих типов:

* один-к-одному;
* один-ко-многим;
* многие-к-одному;
* много-ко-многим.

Связь типа один-к-одному означает, что один экземпляр одного отношения связан с одним экземпляром второго отношения.

Связь типа один-ко-многим означает, что один экземпляр одного отношения связан с несколькими экземплярами второго отношения. Это наиболее часто используемый тип связи. Первое отношение (со стороны «один») называется родительским, второе (со стороны «много») – дочерним.

Связь типа один-ко-многим означает, что несколько экземпляров одного отношения связан с одним экземпляром второго отношения.

Связь типа много-ко-многим означает, что каждый экземпляр одного отношения может быть связан с несколькими экземплярами второго отношения, и каждый экземпляр второго отношения может быть связан с несколькими экземплярами первого отношения. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах проектирования схемы базы данных.

Каждая связь может иметь одну из двух модальностей связи, таких как:

* «может»;
* «должен».

Модальность «может» означает, что экземпляр одного отношения может быть связан с одним или несколькими экземплярами другого отношения, а может быть и не связан ни с одним экземпляром.

Модальность «должен» означает, что экземпляр одного отношения обязан быть связан не менее чем с одним экземпляром другого отношения.

Далее при проектировании схемы базы данных следует определить атрибуты отношений.

Отношение «Фильмы» имеет следующий набор атрибутов:

* название;
* длительность;
* дата начала проката;
* дата окончания проката;
* цена.

Отношение «Сеансы» имеет следующие атрибуты:

* дата начала сеанса;
* время начала сеанса.

Отношение «Билет» имеет следующие атрибуты:

* информация о билете;
* информация о фильме;
* информация о забронированном месте.

Отношение «Место» имеет следующие атрибуты:

* номер ряда;
* номер места.

Отношение «Состояние» имеет следующие атрибуты:

* состояние.

Отношение «Возрастное ограничение» имеет следующие атрибуты:

* возрастное ограничение.

Отношение «Жанры» имеет следующий атрибут: жанр.

Отношение «Страна» имеет следующий атрибут: страна.

Отношение «Продюсер» имеет следующие атрибуты:

* имя;
* фамилия.

Отношение «Формат» имеет следующий атрибут: формат.

Отношение «Актеры» имеет следующий атрибут: актер.

Схема БД представлена на рисунке 2.1.

Таким образом, схема сущность-связь будет иметь следующий вид, как показано на рисунке 2.1.

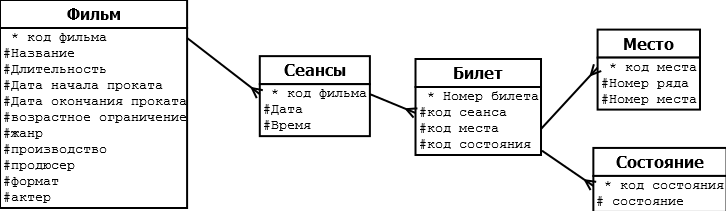


Рисунок 2.1 – Диаграмма «сущность-связь»

Инфологическая модель БД – это модель, которая описывает данные предметной области с использованием естественного языка. Информационно-логическая модель отображает данные предметной области в виде совокупности информационных объектов и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных.

Цель инфологического моделирования обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

Следующим шагом проектирования после выявления сущностей является определение связей между ними. Связь устанавливается между двумя сущностями, если логически взаимосвязаны их экземпляры.

Связь – ассоциирование двух или более сущностей. Если бы назначением базы данных было только хранение отдельных, не связанных между собой данных, то ее структура могла бы быть очень простой. Однако одно из основных требований к организации базы данных - это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи.

2.2 Алгоритм реализации задачи

Алгоритм – это система последовательных операций (в соответствии с определёнными правилами) для решения какой-нибудь задачи.

Виды алгоритмов:

* линейный алгоритм - набор команд (указаний), выполняемых последовательно во времени друг за другом;
* разветвляющийся алгоритм - алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько альтернативных ветвей алгоритма;
* циклический алгоритм – алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов.

Разрабатываемое программное средство предназначено для учета билетов на киносеансы.

В зависимости от прав доступа к программному средству будет реализовано два режима работы. В режиме пользователя возможны функции просмотра информации о кинотеатре, сеансе, количестве занятых мест, схеме зрительного зала, брони билета. В режиме администратора будет предоставлен доступ ко всей информации и возможность ее редактирования.

Алгоритм работы программного средства автоматизации рабочего места сотрудника кинотеатра представлен в приложении Г.

Проектирование информационных систем, включающих в себя базы данных, осуществляется на физическом и логическом уровнях. Решение проблем проектирования на физическом уровне во многом зависит от используемой системы управления базами данных (СУБД).

2.3 Описание сред реализации задачи

Базы данных – совокупность данных, организованная по определенным правилам, предусматривающая общие принципы описания, хранения, манипулирования данными, независимыми от прикладных программ.

СУБД – совокупность программ, предназначенных для управления БД и возможности получения пользователями необходимой информации из базы. В задачи СУБД входят:

* формирование и поддержание БД;
* обработка информации;
* прием запросов;
* предоставление информации пользователям;
* обеспечение целостности и реорганизации ценностей БД;
* организация совместной работы пользователей.

Для выполнения данного курсового проекта был выбран Язык SQL в СУБД.

 SQL (англ. Structured Query Language – язык структурированных запросов) – универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. SQL в его базовом виде является информационно-логическим языком, а не языком программирования. Вместе с тем стандарт языка спецификацией SQL/PSM предусматривает возможность его процедурных расширений, с учётом которых язык уже вполне может рассматриваться в качестве языка программирования.

SQL основывается на реляционной алгебре.

Язык SQL делится  на четыре части:

* операторы  определения данных (англ. Data Definition Language, DDL);
* операторы  манипуляции данными (англ. Data Manipulation Language, DML);
* операторы  определения доступа к данным (англ. Data Control Language, DCL);
* операторы управления транзакциями (англ. Transaction Control Language, TCL).

Data Definition Language (DDL) (язык описания данных) – это язык, используемый в компьютерных  программах для описания структуры  баз данных.

Функции языков DDL определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL эти глаголы – "create" ("создать"), "alter" ("изменить"), "drop" ("удалить"). Это превращает природу языка в ряд обязательных утверждений (команд) к базе данных.

Data Manipulation Language (DML) (язык управления данными) – это язык, используемый в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных. Функции языков DML определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL эти глаголы – "insert" ("вставить"), "update" ("обновить"), и "delete" ("удалить"). Это превращает природу языка в ряд обязательных утверждений (команд) к базе данных.

Язык SQL имеет слудующие преимущества:

* независимость от конкретной СУБД. Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально закладывались на применение по меньшей мере нескольких СУБД;
* наличие стандартов для выявления совместимости  и соответствия конкретной реализации SQL общепринятому стандарту только способствует «стабилизации» языка.
* декларативность. С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса.

Недостатки:

* повторяющиеся строки;
* неопределённые значения (nulls);
* явное  указание порядка колонок слева  направо;
* колонки  без имени и дублирующиеся  имена колонок;
* отсутствие поддержки свойства «=»;
* использование  указателей.

Для организации СУБД была выбрана программа Microsoft SQL server.

Одной из самых распространенных СУБД, используемых в крупных организациях, является Microsoft SQL server, СУБД от компании Microsoft. Она использует реляционную модель данных. Microsoft SQL server является клиент-серверной СУБД, что делает ее хорошей для использования в крупных организациях, имеющих мощный сервер. Платформа данных MS SQL Server 2012 включает следующие инструменты для работы с базой данных:

* поддержка  структурированных и неструктурированных (XML) данных;
* Replication Services: репликация данных для распределённых и мобильных приложений обработки  данных, высокая доступность систем, масштабируемый параллелизм со вторичными хранилищами данных для отчётных решений предприятия и интеграция с разнородными системами, включая  существующие базы данных Oracle;
* Notification Services: развитые возможности уведомлений  для разработки и внедрения масштабируемых приложений, способных доставлять персонализированные, своевременные обновления множеству соединённых и мобильных  устройств.
* Integration Services: возможности извлечения, преобразования и загрузки для хранилищ данных и  интеграции данных в масштабе предприятия.
* Analysis Services: аналитическая обработка в  реальном времени (OLAP) для быстрого, сложного анализа больших и смешанных  наборов данных, использующая многомерное  хранение.
* Reporting Services: исчерпывающее решение для  создания, управления и доставки как  традиционных бумажных отчётов, так  и интерактивных, основанных на технологии WWW отчётов.
* инструменты  управления: SQL Server включает средства управления для развитого управления и настройки баз данных, также  как и тесную интеграцию с такими инструментами, как Microsoft Operations Manager (MOM) и Microsoft Systems Management Server (SMS). Стандартные  протоколы доступа к данным существенно  уменьшают время, необходимое для  интеграции данных SQL Server с существующими  системами. В дополнение, поддержка Web служб встроена для обеспечения  взаимодействия с другими приложениями и платформами.
* инструменты  разработки: SQL Server предлагает интегрированные  инструменты разработки для ядра базы данных, извлечения, трансформации  и загрузки данных, извлечения информации, OLAP и отчётности, которые тесно  интегрированы с Microsoft Visual Studio® для  предоставления сквозных возможностей разработки приложений. Каждая главная  подсистема SQL Server поставляется со своей  собственной объектной моделью  и набором API для расширения системы  данных в любом направлении, которое  уникально для вашего бизнеса.

Для разработки интерфейса и программной составляющей была выбрана среда разработки Visual Studio 2013 и язык программирования C#.

Microsoft Visual Studio — это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию. Эти инструменты предназначены для максимально эффективной совместной работы; все они доступны в интегрированной среде разработки (IDE) Visual Studio.

Visual Studio можно использовать для создания различных типов приложений, от простых приложений для магазина и игр для мобильных клиентов до больших и сложных систем, обслуживающих предприятия и центры обработки данных. Вы можете создавать:

* приложения и игры, которые выполняются не только на платформе Windows, но и на Android и iOS;
* веб-сайты и веб-службы на основе ASP.NET, JQuery, AngularJS и других популярных платформ;
* приложения для самых разных платформ и устройств, включая, но не ограничиваясь: Office, Sharepoint, Hololens, Kinect и "Интернета вещей";
* игры и графические приложения для разных устройств Windows, включая Xbox, с поддержкой DirectX.
* по умолчанию Visual Studio обеспечивает поддержку C#, C и C++, JavaScript, F# и Visual Basic.

Приложение разрабатывается на языке программирования C#, который разработан для создания множества приложений, работающих в среде .NET Framework. Язык C# прост и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую С-подобным языкам.

В C# представлены функциональные возможности, позволяющие создавать надежные и устойчивые приложения. Среди них: функция сборки мусора для автоматического освобождения памяти, занимаемой неиспользуемыми объектами; функция обработки исключительных ситуаций.

3 Программная реализация

3.1 Физическая структура базы данных

Реляционная база данных – это совокупность отношений, содержащих всю информацию, которая должна храниться в БД. Пользователи могут воспринимать такую базу данных как совокупность таблиц. При проектировке таблиц, необходимо руководствоваться следующими основными принципами:

* не должно быть повторений и между таблицами;
* каждая таблица должна содержать информацию только на одну тему;
* каждое поле должно быть связано с темой таблицы;
* не рекомендуется включать в таблицу данные, которые являются результатом выражения;
* в таблице должна присутствовать вся необходимая информация;
* информацию следует разбивать на наименьшие логические единицы.
* каждая таблица состоит из однотипных строк и имеет уникальное имя;
* строки имеют фиксированное число полей (столбцов) и значений (множественные поля и повторяющиеся группы недопустимы). Иначе говоря, в каждой позиции таблицы на пересечении строки и столбца всегда имеется в точности одно значение или ничего;
* строки таблицы обязательно отличаются друг от друга хотя бы единственным значением, что позволяет однозначно идентифицировать любую строку такой таблицы;
* столбцам таблицы однозначно присваиваются имена, и в каждом из них размещаются однородные значения данных (даты, фамилии, целые числа или денежные суммы);
* полное информационное содержание базы данных представляется в виде явных значений данных и такой метод представления является единственным. В частности, не существует каких-либо специальных "связей" или указателей, соединяющих одну таблицу с другой;
* при выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке безотносительно к их информационному содержанию. Этому способствует наличие имен таблиц и их столбцов, а также возможность выделения любой их строки или любого набора строк с указанными признаками.

В таблицах данные распределяются по столбцам (которые называют полями) и строкам (которые называют записями). Все данные, содержащиеся в поле таблицы, должны иметь один и тот же тип. Каждое поле таблицы характеризуется наименованием, типом и шириной поля. При задании типа дан­ных поля можно также указать размер, формат и другие параметры, влияющие на отображение значения поля и точность числовых данных. Основные типы данных:

* текстовый. Текст или числа не требующие проведения расчётов;
* МЕМО. Поле этого типа предназначено для хранения небольших текстовых данных (до 64000 символов). Поле этого типа не может быть ключевым или проиндексированным;
* числовой. Этот тип данных содержит множество подтипов. От выбора подтипа (размера) зависит точность вычислений;
* счётчик. Уникальные, последовательно возрастающие числа, автоматически вводящиеся при добавлении новой записи в таблицу;
* логический. Логические значения, а так же поля, которые могут содержать одно из двух возможных значений;
* денежный. Денежные значения и числовые данные, используемые в математических вычислениях;
* Дата/Время. Дата и время хранятся в специальном фиксированном формате;
* поле объекта OLE. Включает звукозапись, Рис. и прочие типы данных. Поле этого типа не может быть ключевым или проиндексированным;
* гиперсвязь. Содержит адреса Web-страниц.

Используемые типы данных продемонстрированы в таблице 3.1:

Таблица 3.1 – типы данных в таблицах

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поле | Тип данных | | Поле | Тип данных | |
| Таблица «Фильмы» | | | Таблица «Билет» | | |
| идентификатор фильма | | int | идентификатор билета | | int |
| название | | varchar | идентификатор сеанса | | int |
| длительность | | time | идентификатор места | | int |
| дата начала проката | | date | идентификатор состояния | | int |
| дата окончания проката | | date | Таблица «Возрастное ограничение» | | |
| цена | | int | идентификатор возрастного ограничения | | int |
| идентификатор возрастного ограничения | | int | возрастное ограничение | | varchar |
| Продолжение таблицы 3.1 | | | | | |
| Поле | Тип данных | | Поле | Тип данных | |
| идентификатор жанра | | int | Таблица «Жанры» | | |
| идентификатор производителя | | int | идентификатор жанра | | int |
| идентификатор продюсера | | int | жанр | | varchar |
| идентификатор формата | | int | Таблица «Страна» | | |
| идентификатор актеров | | int | идентификатор производителя | | int |
| Таблица «Сеансы» | | | страна | | varchar |
| идентификатор фильма | | int | Таблица «Продюсер» | | |
| дата начала сеанса | | date | идентификатор продюсера | | int |
| время начала сеанса | | time | имя | | varchar |
| Таблица «Место» | | | фамилия | | varchar |
| идентификатор места | | int | Таблица «Формат» | | |
| номер ряда | | int | идентификатор формата | | int |
| номер места | | int | формат | | varchar |
| Таблица «Состояние» | | | Таблица «Актеры» | | |
| идентификатор состояния | | int | идентификатор актера | | int |
| состояние | | varchar | актер | | varchar |

Разрабатываемая в данном курсовом проекте база данных для сотрудника кинотеатра, позволяет автоматизировать весь процесс редактирования данных. При появлении новых задач разработчик может в кратчайшие сроки реализовать их в базе данных, путем добавления строк, столбцов и целых таблиц.

Рассмотрим реализованные таблицы базы данных кинотеатра.

Таблица «film» содержит в себе основную информацию о текущем сеансе и такие поля как «fname» (название фильма), «long» (длительность сеанса), «start» (дата начала проката), «finish» (дата окончания проката), «cost» (стоимость билета), «description» (описание фильма). Имеет первичный ключ «id\_film» и внешние ключи:

* «id\_vozrastogr»;
* «id\_janr»;
* «id\_strana»;
* «id\_produser»;
* «id\_format».

Таблица «vozrast» имеет поле перечня возрастных ограничений для фильма «vozrast» и первичный ключ «id\_vozrast».

Таблица «janr» включает поля «janr» и первичный ключ «id\_janr».

Таблица «strana» содержит поля «proizvodstvo», описывающее страну изготовления фильма, и первичный ключ «id\_proizvodstva».

Таблица «produser» характеризуется полями «produser» и «id\_ produser» – первичный ключ.

Таблица «akter» содержит имена актеров в поле «name» и имеет первичный ключ

«id\_akter».

Таблица «uchastniki» не имеет первичных ключей и предназначена для определения конкретному фильму множества атрибутов значений таблицы «akter».

Таблица «seansi» описывает полями «date» и «time» дату и время начала сеанса для фильма. Содержит внешний ключ «id\_film» и первичный ключ «id\_seansi».

Таблица «bilet» содержит описание 3 внешних ключа («id\_seans»,«id\_mesto»,«id\_mesto») для составления билета.

Таблица «mesto» определяет для купленного билета рад и место полями «rad», «mesto».

Таблица «sostojanie» характеризуется первичным ключом «id\_sostojanie» и полем «sostojanie».

3.2 Структура приложения

В разрабатываемом программном средстве существует 6 классов: «addkino», «Form1», «Plan», «ticket», «user», «registrationclass».

Класс «Plan» содержит схему зрительного зала и возможность покупки билетов.

В нем содержатся следующие поля:

* string name – название спектакля;
* datetime time – время начала спектакля;
* datetime date – дата начала спектакля.

Класс «addkino» – это класс, с помощью которого происходит добавление фильма.

В нем содержатся следующие поля:

* string name – название фильма;
* datetime date – дата начала спектакля;
* datetime time – время начала спектакля;
* string sost – состояние (забронирован\оплачен);
* string rad – забронированный ряд;
* float cost – стоимость билета;
* string mesto – забронированное место.

Класс «Form1»– данный класс является главным меню, который предоставляет выход ко всем основным функциям программы как для пользователя, так и для администратору. Позволяет сортировать афишу по трем критериям (дате начала сеанса, жанру и формату кино).

Класс «user» – содержит необходимые поля для авторизации пользователя

В нем содержатся следующие поля:

* public string Login – логин;
* public string Password – пароль.

Класс «registrationclass» – содержит методы считывания из файла информации для проверки на правильность ввода логина и пароля, дает возможность изменить данные и предоставляет функции администратора.

Класс «ticket» – формирует купленный билет на печать.

Также в программе разработаны следующие методы:

Класс «Form1»:

* public void akter() – вывод актеров, участвующих в фильме ;
* public void indexf() – определение индекса фильма;
* public void ShowTable() – запрос на вывод информации о фильме;
* public Form1() – инициализация компонентов формы;
* private void Form1\_FormClosing(object sender, FormClosingEventArgs e) – подтверждение о закрытии программы;
* private void pictureBox1\_Click(object sender, EventArgs e) – переход к следующему сеансу;
* private void pictureBox2\_Click(object sender, EventArgs e) – переход к предыдущему сеансу;
* private void pictureBox4\_Click(object sender, EventArgs e) – вызов окна авторизации;
* private void label2\_Click(object sender, EventArgs e) – переход к форме «Plan»;
* public void janr() – отображение жанров в компоненте «comboBox1»;
* private void button1\_Click(object sender, EventArgs e) – осуществление поиска.

Класс «Plan»:

* public Plan() – инициализация компонентов формы;
* public void bileti() – запрос на поиск купленных билетов на данный сеанс;
* public void kolbilet() – отображение купленных билетов на схеме зрительного зала;
* public void chistka () – очистка схемы зрительного зала;
* private void button1\_Click(object sender, EventArgs e) – бронь билета;
* private void button2\_Click(object sender, EventArgs e) – покупка билета;
* private void pictureBox2\_Click(object sender, EventArgs e) – перейти к следующему сеансу;
* private void pictureBox3\_Click(object sender, EventArgs e) - перейти к предыдущему сеансу;
* public void id() – запрос, определяющий забронированное место.

Класс user:

* private void button1\_Click(object sender, EventArgs e) – авторизация;
* private void button3\_Click(object sender, EventArgs e) – чтение из файла данных;
* private void изменитьЛогинToolStripMenuItem\_Click(object sender, EventArgs e) – изменение данных в файле.

Класс ticket:

* private void drawticket(string name, string time, string date, int cost, int Count, string rad, string place) – формирование билета;
* private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e) – печать билета.

Класс addkino:

* public addkino() - инициализация компонентов формы;
* public void filmakter() – список актеров, участвующих в фильме;
* public void film() – запрос на вывод названий всех фильмов;
* public void janr() – отображение всех жанров в компоненте;
* public void indexf() – определение индекса фильма по его названию;
* public void akter() – вывод актеров, участвующих в фильме;
* private void button1\_Click(object sender, EventArgs e) – добавление фильма, сеанса, актера, в зависимости от выделенного «radiobutton»;
* private void button2\_Click(object sender, EventArgs e) – удаление данных;
* private void radioButton1\_CheckedChanged(object sender, EventArgs e) – запрос на вывод информации о фильме;
* private void radioButton2\_CheckedChanged(object sender, EventArgs e) – запрос на вывод сеансов на фильм;
* private void radioButton3\_CheckedChanged(object sender, EventArgs e) - запрос на вывод купленных и забронированных билетов на сеанс;
* private void radioButton4\_CheckedChanged(object sender, EventArgs e) – запрос на вывод актеров, участвующих в фильме;
* private void checkBox1\_CheckedChanged(object sender, EventArgs e) – изменение отображаемой информации от выбранного «radiobutton».

Для разработки потребовались следующие компоненты:

* menuStrip – предоставляет систему меню для формы;
* button – элемент управления "кнопка", реагирующий на событие Click;
* textBox – элемент управления, который может использоваться для ввода информации;
* label – возвращает или задает текстовое содержимое;
* pictureBox – используется для отображения рисунка;
* dataTimePicker – отображает время и дату;
* bindingSourse – используется для подключения БД;
* comboBox – предназначен для выбора значения, с раскрывающимся списком, который можно отображать и скрывать нажатием принадлежащей этому элементу управления кнопки со стрелкой;
* checkbox – представляет флажок;
* printDocument – отправляет выходные данные на принтер при печати из приложения Windows Forms;
* printPreviewDialog – вызывает диалоговое окно для печати;
* radioButton – позволяет пользователю выбрать единственный вариант из группы доступных, когда используется вместе с другими элементами управления;
* colorDialod – представляет общее диалоговое окно, в котором отображаются доступные цвета и элементы управления, позволяющие пользователю определять собственные цвета;
* listBox – содержит список элементов для выбора.

Элементом для отображения табличных данных используется dataGridView.

4 Тестирование

Тестирование программного обеспечения – процесс исследования, испытания [программного продукта](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), имеющий две различные цели:

* продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
* выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

[Качество программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%87%D0%B5%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%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) можно определить как совокупную характеристику исследуемого ПО с учётом следующих составляющих:

* [надёжность](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B4%D1%91%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C);
* [сопровождаемость](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BF%D1%80%D0%BE%D0%B2%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F);
* практичность;
* эффективность;
* мобильность;
* функциональность.

Уровни тестирования

* [модульное тестирование](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция.
* [интеграционное тестирование](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – тестируются интерфейсы между компонентами, подсистемами или системами.
* [системное тестирование](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – тестируется интегрированная система на её соответствие [требованиям](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E).
* альфа-тестирование – имитация реальной работы с системой штатными [разработчиками](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82), либо реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в некоторых случаях может применяться для законченного продукта в качестве внутреннего приёмочного тестирования.
* [бета-тестирование](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%82%D0%B0-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) – в некоторых случаях выполняется распространение предварительной версии для некоторой большей группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок. Иногда бета-тестирование выполняется для того, чтобы получить обратную связь о продукте от его будущих пользователей.

Таблица 4.1 – Ход тестирования

|  |  |  |
| --- | --- | --- |
| Ход тестирования | Ожидание | Результат |
| Удаление фильма | Изменение БД | БД изменена |
| Добавление фильма | Изменение БД | БД изменена |
| Осуществление поиска | Вывод необходимой информации | Вывод необходимой информации |
| Изменение БД | Изменения информации в программе | Соответствующие изменения в программе |
| Добавление нового билета | Изменение БД, Изменение схемы зрительного зала | БД и схема зрительного зала изменена |
| Удаление билета | Изменение БД, Изменение схемы зрительного зала | БД и схема зрительного зала изменена |
| Изменение состояния билета «Забронирован» на «Оплачен» | Изменение БД, Изменение схемы зрительного зала | БД и схема зрительного зала изменена |
| Просмотр справочной информации | Отображение справочной информации | Отображение справочной информации |
| Авторизация | Доступ ко всей информации | Доступ ко всей информации |
| Изменение параметров авторизации | Изменение параметров авторизации | Параметры изменены |
| Поиск с пустыми полями | Запрет на поиск | Вывод предупреждения |
| Добавление с пустыми полями | Запрет на добавление | Вывод предупреждения |

В ходе тестирования программного средства автоматизации рабочего места сотрудника кинотеатра ошибок не найдено.

5. Описание применения

Системные требования – это описание примерных характеристик, которым должен соответствовать [компьютер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) для того, чтобы на нём могло использоваться какое-либо определённое [программное обеспечение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).

Данное программное средство было разработано на компьютере со следующими системными характеристиками:

* процессор 2.4 GHz;
* 8,00 ГБ ОЗУ;
* видеокарта на 64 МВ;
* операционная система Microsoft Windows 7.

Для оптимальной работы программы требуется:

* процессор 1,2 GHz;
* 256 МВ ОЗУ;
* видеокарта на 32 МВ;
* 10 МВ свободного дискового пространства;
* операционная система MicrosoftWindows 98/2000/NT/XP/Vista/7,8,10.

Для корректной работы программы требуется установить **SQL Server Management Studio Express с и поместить файлы dbcin и dbcin\_log в папку DATA в Microsoft SQL Server.**

Чтобы установить программное средство «Кинотеатр» необходим диск с установщиком «Setup.exe». При его открытии откроется окно, которое представлено на рисунке 5.1.

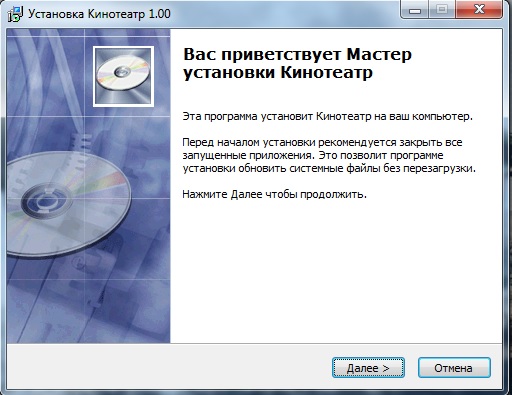


Рисунок 5.1 – Установка программного средства

Для продолжения установки нажмите «Далее». После чего откроется форма, представленная на рисунке рисунок 5.2, где требуется выбрать место установки.

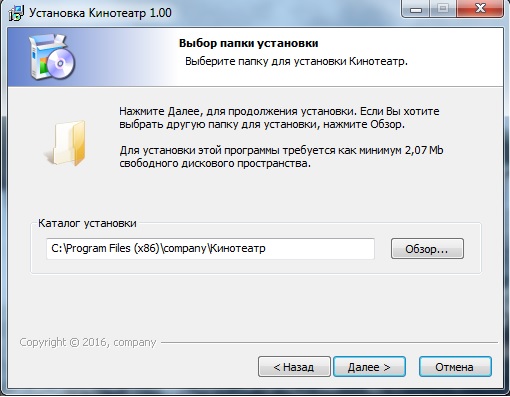


Рисунок 5.2 – Выбор места установки

Перед установкой будут представлены установочные параметры, как показано на рисунке 5.3. После окончания установки будет выдано соответствующее уведомление.

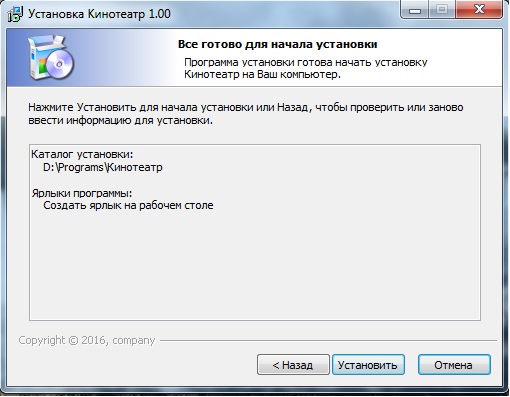


Рисунок 5.3 – Параметры установки

Соблюдение вышеперечисленных системных требований приведёт к оптимальной работе программы, не влияя на работу других приложений.

Начало работы осуществляется с открытия главной формы, представленной на рисунке 5.4, на котором пользователю предоставляется информация о текущих фильмах. Для просмотра данных о киносеансах используйте кнопки переключения. Чтобы выбрать интересующий фильм можно использовать критерии поиска. Поставьте галочку возле необходимого поля и нажмите на кнопку "Поиск". Для покупки билета нажмите на соответствующую кнопку.

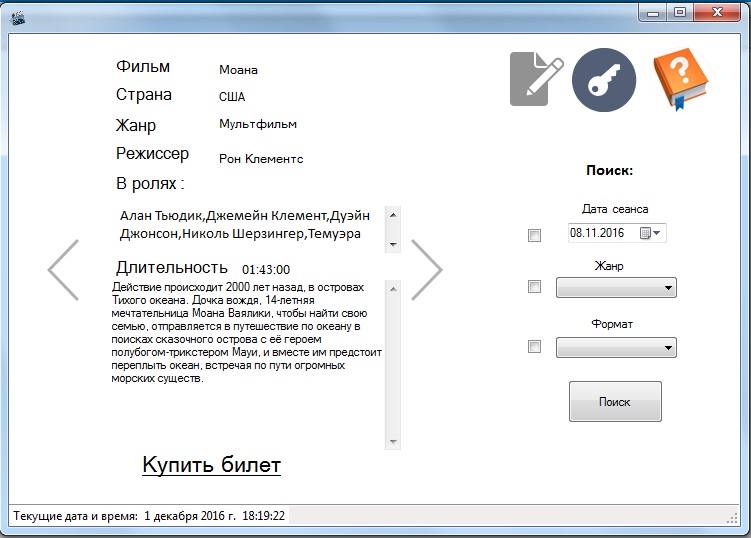


Рисунок 5.4 – Главное меню

На форме покупки билета, показанной на рисунке 5.5 можно просмотреть количество купленных и забронированных билетов на определенный сеанс, который отображаются справа от схемы зрительного зала. Если желаете пометить билет как купленный или забронированный выделите необходимое место и нажмите на кнопку "Купить" или "Забронировать". Для того чтобы изменить состояние билета с забронированного на оплаченный требуется выделить соответствующее место и нажать кнопку "Купить".

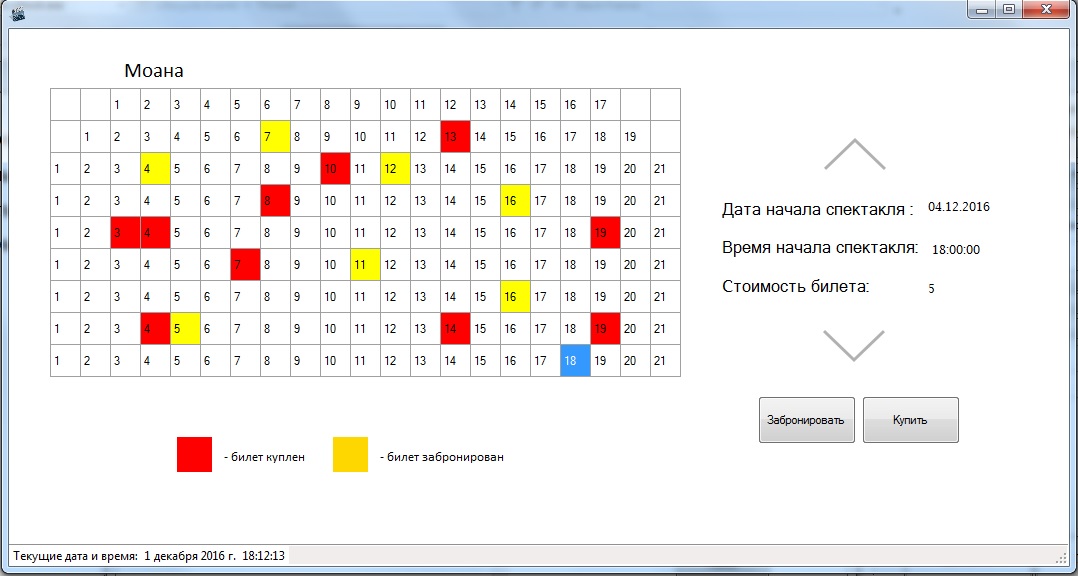


Рисунок 5.5– Форма покупки билета

Для авторизации необходимо ввести логин и пароль, как проиллюстрировано на рисунке 5.6, после чего открываются возможности добавлять и редактировать данные.

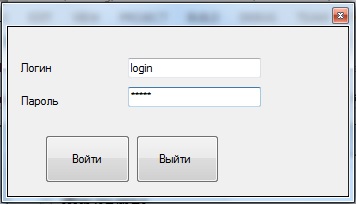


Рисунок 5.6 – Авторизация пользователя

Для манипуляции данными администратором выберите на форме, продемонстрированной на рисунке 5.7, необходимый пункт и заполните пустые поля данными для добавления.

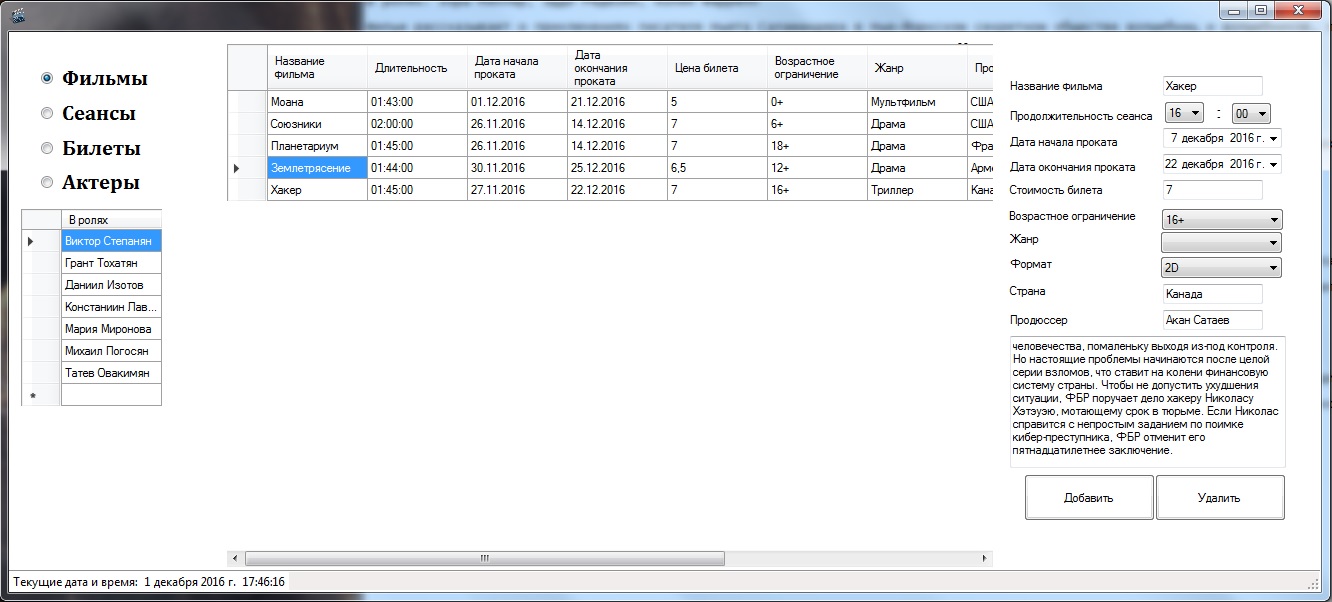


Рисунок 5.7 – Манипуляция данными

Чтобы удалить данные из базы данных, необходимо выделить любое поле или строку и нажать кнопку "Удалить".

Заключение

Целью курсового проекта является разработка автоматизированной системы продажи билетов на сеансы в кино.

Результатом выполнения курсового проекта стало разработанное приложение баз данных, позволяющее автоматизировать операции сотрудника кинотеатра и увеличить эффективность работы персонала.При создании программного средства учитывалась специфика и особенности деятельности кинотеатра.

Во время создания программного средства были изучены и укреплены навыки работы с языком SQL для создания, модификации и управления данными в СУБД Microsoft SQL Server . Также были разобраны основы системного программирования и основы объектно-ориентированного подхода к программированию. Разработанное приложение отвечает всем требованиям предметной области, таблицы созданной базы данных отвечают требованиям нормализации, что позволяет обеспечить целостность и непротиворечивость информации.

Основные преимущества курсового проекта:

* реализована возможность брони билетов;
* существует два режима работы: пользователь, администратор;
* автоматическое формирование билета для печати;
* возможность редактировать информацию;
* простой интерфейс;
* присутствует справочная система;
* высокая скорость обработки данных;
* высокая надежность автоматизированной системы.

В процессе решения поставленных задач курсовой работы использовались прикладные системы программирования и необходимые методы решения заданий.

По окончанию курсового проекта можно сказать, что были изучены и закреплены следующие задачи:

* ставить задачу и разрабатывать алгоритм её решения;
* использовать прикладные системы программирования;
* разрабатывать основные программные документы;
* работать с современными системами программирования, включая объектно-ориентированные.

При разработке программного средства, поставленная задача выполнена полностью.

# Список использованных источников

1. Голицина, О. Л., Попов, И. И. Microsoft SQL Server 2008. Основы T-SQL: Учебное пособие / Голицина О. Л. – М.: ФОРУМ: ИНФРА, 2005. – 340 c.
2. Нортроп Тони, Уилдермьюс Шон, Райан Билл. Основы разработки приложений на платформе Microsoft .Net Framework. Учебный курс Microsoft / Пер. с англ. – М.: «Русская редакция», СПб.: «Питер», 2007. – 514 c.
3. Павловская, Т. А. C#. Программирование на языке высокого уровня: Учебник для вузов / Павловская Т. А. – СПб.: БХВ-Петербург. 2007. – 245 c.
4. Павловская, Т. А. С/С++ Программирование на языке высокого уровня / Павловская Т. А. – СПб.: Питер, 2005. – 261 c.
5. Попов, И. И., Партыка Т. Л. Языки программирования: Учебное пособие/ Попов И. И., Партыка Т. Л. – М.: ФОРУМ: ИНФРА-М, 2008. – 452 c.
6. Ржеуцкая, С.Ю. Базы данных. Язык SQL / Ржеуцкая С.Ю – М.: Эксмо, 2008. – 251 c.

Приложение А

(обязательное)

Текст программы

public partial class Form1 : Form

{

public static int a;

public static string l1;

private SqlConnection con;

private SqlCommand cmd;

private SqlDataAdapter da;

private DataTable dt;

BindingSource bis;

ToolStripLabel dateLabel;

ToolStripLabel timeLabel;

ToolStripLabel infoLabel;

Timer timer;

int select = 0;

string akt;

int kol = 0;

int index = 0;

public void janr()

{

con.Open();

cmd = new SqlCommand("SELECT janr FROM janr ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView4.DataSource = bSource;

da.Update(dt);

comboBox1.Items.Clear();

con.Close();

for (int i = 0; i < dataGridView4.RowCount - 1; i++)

{ comboBox1.Items.Add(dataGridView4.Rows[i].Cells[0].Value); }

}

public void dt2()

{

akt = "";

textBox3.Text ="";

for (int i = 0; i < dataGridView2.RowCount - 1; i++)

{

akt += dataGridView2.Rows[i].Cells[0].Value.ToString();

if (i != dataGridView2.RowCount - 2) { akt += ","; }

else { akt += "."; }

}

textBox3.Text = akt;

}

public void indexf()

{

string fname = dataGridView1.Rows[select].Cells[0].Value.ToString();

cmd = new SqlCommand("SELECT id\_film FROM film where film.fname= '" + fname + "' ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView3.DataSource = bSource;

index = Convert.ToInt16(dataGridView3.Rows[0].Cells[0].Value);

label6.Text = index.ToString();

da.Update(dt);

}

public void akter()

{

con.Open();

indexf();

cmd = new SqlCommand("SELECT DISTINCT akter.name FROM film,akter,uchastniki where uchastniki.id\_film='" + index + "' and uchastniki.id\_akter=akter.id\_akter ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

da.Update(dt);

dt2();

con.Close();

}

public void ShowTable()

{

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

public Form1()

{

InitializeComponent();

pictureBox2.Visible = false;

con = new SqlConnection(@"Data Source=USERPC\SQLEXPRESS;Initial Catalog=dbcin;Integrated Security=True;Pooling=False");

ShowTable();

label1.Text = dataGridView1.Rows[0].Cells[0].Value.ToString();

label3.Text = dataGridView1.Rows[0].Cells[7].Value.ToString();

label4.Text = dataGridView1.Rows[0].Cells[8].Value.ToString();

label5.Text = dataGridView1.Rows[0].Cells[6].Value.ToString();

label19.Text = dataGridView1.Rows[0].Cells[1].Value.ToString();

textBox2.Text = dataGridView1.Rows[0].Cells[10].Value.ToString();

infoLabel = new ToolStripLabel();

infoLabel.Text = "Текущие дата и время:";

dateLabel = new ToolStripLabel();

timeLabel = new ToolStripLabel();

statusStrip1.Items.Add(infoLabel);

statusStrip1.Items.Add(dateLabel);

statusStrip1.Items.Add(timeLabel);

timer = new Timer() { Interval = 100 };

timer.Tick += timer\_Tick;

timer.Start();

for (int i = 0; i < dataGridView1.RowCount; i++)

{

for (int j = 0; j < dataGridView1.ColumnCount; j++)

{

dataGridView1.Columns[0].Width = 260;

dataGridView1.Columns[1].Width = 120;

dataGridView1.Columns[2].Width = 120;

dataGridView1.Columns[3].Width = 120;

dataGridView1.Columns[4].Width = 60;

dataGridView1.Columns[5].Width = 80;

dataGridView1.Columns[6].Width = 130;

dataGridView1.Columns[7].Width = 120;

dataGridView1.Columns[8].Width = 200;

dataGridView1.Columns[9].Width = 100;

dataGridView1.Columns[10].Width = 180;

}

}

}

void timer\_Tick(object sender, EventArgs e)

{

dateLabel.Text = DateTime.Now.ToLongDateString();

timeLabel.Text = DateTime.Now.ToLongTimeString();

if (kol == 0)

{

akter();

dt2();

kol++;

}

if (addkino.ob == 1)

{

ShowTable();

addkino.ob = 1;

}

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialog = MessageBox.Show("Вы действительно хотите выйти ?", "Выход", MessageBoxButtons.YesNo);

if (dialog==DialogResult.Yes)

{

Application.ExitThread();

}

else if (dialog==DialogResult.No)

{

e.Cancel = true;

}

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if ( select != dataGridView1.RowCount-1)

{

dataGridView1.Rows[select].Selected = false;

select++;

if (select != dataGridView1.RowCount - 1)

{

pictureBox1.Visible = true;

pictureBox2.Visible = true;

dataGridView1.Rows[select].Selected = true;

label1.Text = dataGridView1.Rows[select].Cells[0].Value.ToString();

label3.Text = dataGridView1.Rows[select].Cells[7].Value.ToString();

label4.Text = dataGridView1.Rows[select].Cells[8].Value.ToString();

label5.Text = dataGridView1.Rows[select].Cells[6].Value.ToString();

label19.Text = dataGridView1.Rows[select].Cells[1].Value.ToString();

textBox2.Text = dataGridView1.Rows[select].Cells[10].Value.ToString();

akter();

}

else

{

pictureBox1.Visible = false;

select--;

dataGridView1.Rows[select].Selected = true;

}

}

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

if (select != 0 )

{

pictureBox1.Visible = true;

pictureBox2.Visible = true;

dataGridView1.Rows[select].Selected = false;

select--;

dataGridView1.Rows[select].Selected = true;

label1.Text = dataGridView1.Rows[select].Cells[0].Value.ToString();

label3.Text = dataGridView1.Rows[select].Cells[7].Value.ToString();

label4.Text = dataGridView1.Rows[select].Cells[8].Value.ToString();

label5.Text = dataGridView1.Rows[select].Cells[6].Value.ToString();

label19.Text = dataGridView1.Rows[select].Cells[1].Value.ToString();

textBox2.Text = dataGridView1.Rows[select].Cells[10].Value.ToString();

akter();

}

else { pictureBox2.Visible = false; }

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

if (a == 0) { MessageBox.Show("Авторизируйтесь"); }

else

{

addkino ad = new addkino();

ad.Show();

}

}

private void label2\_Click(object sender, EventArgs e)

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.fname,seansi.date ,seansi.time FROM film,seansi where film.id\_film=seansi.id\_film and seansi.id\_film='" + index + "' ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView6.DataSource = bSource;

da.Update(dt);

con.Close();

if (dataGridView6.RowCount != 0)

{

l1 = label1.Text;

Plan plan = new Plan();

plan.Show();

}

else { MessageBox.Show("Нет сеансов !"); }

}

private void label2\_MouseMove(object sender, MouseEventArgs e)

{

label2.ForeColor = System.Drawing.Color.Blue;

}

private void label2\_MouseLeave(object sender, EventArgs e)

{

label2.ForeColor = System.Drawing.Color.Black;

}

private void comboBox1\_MouseDown(object sender, MouseEventArgs e)

{

janr();

}

private void button1\_Click(object sender, EventArgs e)

{

if (checkBox1.Checked || checkBox2.Checked || checkBox3.Checked)

{

label2.Visible = true;

if (checkBox1.Checked && !checkBox2.Checked && !checkBox3.Checked)

{

label16.Text = dateTimePicker1.Value.ToShortDateString();

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format,seansi where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format and seansi.date='" + label16.Text + "' and seansi.id\_film=film.id\_film");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (!checkBox1.Checked && checkBox2.Checked && !checkBox3.Checked)

{

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format and janr.janr='" + comboBox1.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (!checkBox1.Checked && !checkBox2.Checked && checkBox3.Checked)

{

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format and format.format='" + comboBox2.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (checkBox1.Checked && checkBox2.Checked && !checkBox3.Checked)

{

label16.Text = dateTimePicker1.Value.ToShortDateString();

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format,seansi where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format and seansi.date='" + label16.Text +"'and janr.janr='" + comboBox1.Text + "' and seansi.id\_film=film.id\_film");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (checkBox1.Checked && !checkBox2.Checked && checkBox3.Checked)

{

label16.Text = dateTimePicker1.Value.ToShortDateString();

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format,seansi where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format and seansi.date=='" + label16.Text +"'and format.format='" + comboBox2.Text + "'and seansi.id\_film=film.id\_film");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (!checkBox1.Checked && checkBox2.Checked && checkBox3.Checked)

{

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format and janr.janr='" + comboBox1.Text + "' and format.format='" + comboBox2.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (checkBox1.Checked && checkBox2.Checked && checkBox3.Checked)

{

label16.Text = dateTimePicker1.Value.ToShortDateString();

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format,seansi where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format and janr.janr='" + label16.Text + "' and janr.janr='" + comboBox1.Text + "' and format.format='" + comboBox2.Text + "'and seansi.id\_film=film.id\_film");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (dataGridView1.RowCount != 1)

{

label1.Text = dataGridView1.Rows[0].Cells[0].Value.ToString();

label3.Text = dataGridView1.Rows[0].Cells[7].Value.ToString();

label4.Text = dataGridView1.Rows[0].Cells[8].Value.ToString();

label5.Text = dataGridView1.Rows[0].Cells[6].Value.ToString();

label19.Text = dataGridView1.Rows[0].Cells[1].Value.ToString();

textBox2.Text = dataGridView1.Rows[0].Cells[10].Value.ToString();

}

else

{

label1.Text = "";

label3.Text = "";

label4.Text = "";

label5.Text = "";

label19.Text = "";

textBox2.Text ="";

textBox3.Text = "";

label2.Visible = false;

MessageBox.Show("Фильм не найден");

}

}

else { ShowTable(); }

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

user user = new user();

user.ShowDialog();

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

Help.ShowHelp(this, "HTML.chm");

}

}

}

public partial class Plan : Form

{

Plan p;

DataGridView[] tab = new DataGridView[9];

DataGridView[] tab21 = new DataGridView[7];

string per;

int select = 0;

ToolStripLabel dateLabel;

ToolStripLabel timeLabel;

ToolStripLabel infoLabel;

Timer timer;

private SqlConnection con;

private SqlCommand cmd;

private SqlDataAdapter da;

private DataTable dt;

BindingSource bis;

int proverka = 0;

void timer\_Tick(object sender, EventArgs e)

{

dateLabel.Text = DateTime.Now.ToLongDateString();

timeLabel.Text = DateTime.Now.ToLongTimeString();

}

public Plan()

{

InitializeComponent();

con = new SqlConnection(@"Data Source=USERPC\SQLEXPRESS;Initial Catalog=dbcin;Integrated Security=True;Pooling=False");

infoLabel = new ToolStripLabel();

infoLabel.Text = "Текущие дата и время:";

dateLabel = new ToolStripLabel();

timeLabel = new ToolStripLabel();

statusStrip1.Items.Add(infoLabel);

statusStrip1.Items.Add(dateLabel);

statusStrip1.Items.Add(timeLabel);

timer = new Timer() { Interval = 100 };

timer.Tick += timer\_Tick;

timer.Start();

dataGridView11.RowCount = 9;

dataGridView11.ColumnCount = 21;

for (int i = 0; i < 9; i++)

{

dataGridView11.Rows[i].Height = 32;

for (int j = 0; j < 21; j++)

{

dataGridView11.Columns[j].Width = 30;

if (i == 0)

{

if (j == 17) { break; }

else { dataGridView11.Rows[i].Cells[j + 2].Value = j + 1; }

}

if (i == 1)

{

if (j == 19) { break; }

else { dataGridView11.Rows[i].Cells[j + 1].Value = j + 1; }

}

if (i>1)

{

dataGridView11.Rows[i].Cells[j].Value = j + 1;

}

}

}

dataGridView11.Rows[0].Cells[0].Style.BackColor = Color.White;

dataGridView11.Rows[1].Cells[0].Style.BackColor = Color.White;

dataGridView11.Rows[0].Cells[1].Style.BackColor = Color.White;

dataGridView11.Rows[0].Cells[20].Style.BackColor = Color.White;

dataGridView11.Rows[1].Cells[20].Style.BackColor = Color.White;

dataGridView11.Rows[0].Cells[19].Style.BackColor = Color.White;

label1.Text = Form1.l1;

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.id\_film FROM film where film.fname='" + label1.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label2.Text = dt.Rows[0]["id\_film"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT film.cost FROM film where film.fname='" + label1.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label11.Text = dt.Rows[0]["cost"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT film.fname,seansi.date ,seansi.time FROM film,seansi where film.id\_film=seansi.id\_film and seansi.id\_film='" + label2.Text + "' ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView10.DataSource = bSource;

da.Update(dt);

con.Close();

if (dataGridView10.RowCount != 1)

{

label3.Text = dataGridView10.Rows[0].Cells[1].Value.ToString();

label4.Text = dataGridView10.Rows[0].Cells[2].Value.ToString();

}

else { MessageBox.Show("Нет сеансов !"); }

bileti();

kolbilet();

}

public void bileti()

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.fname,seansi.date ,seansi.time,sostojanie.sostojanie,mesto.rad,mesto.mesto FROM film,sostojanie,seansi,bilet,mesto where seansi.time='" + label4.Text + "' and seansi.date='" + label3.Text + "' and film.id\_film='" + label2.Text + "' and film.id\_film=seansi.id\_film and seansi.id\_seansi=bilet.id\_seans and bilet.id\_sostojanie=sostojanie.id\_sostojanie and bilet.id\_mesto=mesto.id\_mesto ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

public void kolbilet()

{

for (int i=0; i<dataGridView1.RowCount-1; i++)

{

if (dataGridView1.Rows[i].Cells[4].Value.ToString() == "1")

{

if (dataGridView1.Rows[i].Cells[3].Value.ToString() == "оплачен")

{

dataGridView11.Rows[Convert.ToInt16(dataGridView1.Rows[i].Cells[4].Value) - 1].Cells[Convert.ToInt16(dataGridView1.Rows[i].Cells[5].Value) +1].Style.BackColor = Color.Red;

}

if (dataGridView1.Rows[i].Cells[3].Value.ToString() == "забронирован")

{

dataGridView11.Rows[Convert.ToInt16(dataGridView1.Rows[i].Cells[4].Value) - 1].Cells[Convert.ToInt16(dataGridView1.Rows[i].Cells[5].Value) +1].Style.BackColor = Color.Yellow;

}

}

if (dataGridView1.Rows[i].Cells[4].Value.ToString() == "2")

{

if (dataGridView1.Rows[i].Cells[3].Value.ToString() == "оплачен")

{

dataGridView11.Rows[Convert.ToInt16(dataGridView1.Rows[i].Cells[4].Value) - 1].Cells[Convert.ToInt16(dataGridView1.Rows[i].Cells[5].Value) ].Style.BackColor = Color.Red;

}

if (dataGridView1.Rows[i].Cells[3].Value.ToString() == "забронирован")

{

dataGridView11.Rows[Convert.ToInt16(dataGridView1.Rows[i].Cells[4].Value) - 1].Cells[Convert.ToInt16(dataGridView1.Rows[i].Cells[5].Value) ].Style.BackColor = Color.Yellow;

}

}

if (dataGridView1.Rows[i].Cells[4].Value.ToString() != "2" && dataGridView1.Rows[i].Cells[4].Value.ToString() != "1")

{

if (dataGridView1.Rows[i].Cells[3].Value.ToString() == "оплачен")

{

dataGridView11.Rows[Convert.ToInt16(dataGridView1.Rows[i].Cells[4].Value) - 1].Cells[Convert.ToInt16(dataGridView1.Rows[i].Cells[5].Value) - 1].Style.BackColor = Color.Red;

}

if (dataGridView1.Rows[i].Cells[3].Value.ToString() == "забронирован")

{

dataGridView11.Rows[Convert.ToInt16(dataGridView1.Rows[i].Cells[4].Value) - 1].Cells[Convert.ToInt16(dataGridView1.Rows[i].Cells[5].Value) - 1].Style.BackColor = Color.Yellow;

}

}

}

}

public void chistka ()

{

for (int i=0; i<dataGridView11.RowCount;i++)

for (int j = 0; j < dataGridView11.ColumnCount; j++)

{

dataGridView11.Rows[i].Cells[j].Style.BackColor = Color.White;

}

}

private void button1\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 9; i++)

for (int j = 0; j < 21; j++)

{

if (dataGridView11.Rows[i].Cells[j].Selected == true && dataGridView11.Rows[i].Cells[j].Value != null)

{

if (dataGridView11.Rows[i].Cells[j].Style.BackColor == Color.Yellow)

{

if (MessageBox.Show("Билет забронирован, пометить его как купленный ?", "Подтверждение действия", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)

{

con.Open();

int mesto = Convert.ToInt16(dataGridView11.Rows[i].Cells[j].Value.ToString());

int rad = i + 1;

cmd = new SqlCommand("SELECT DISTINCT mesto.id\_mesto FROM mesto where mesto.rad='" + rad + "' and mesto.mesto='" + mesto + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label15.Text = dt.Rows[0]["id\_mesto"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT seansi.id\_seansi FROM seansi where seansi.date='" + label3.Text + "' and seansi.time='" + label4.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

label9.Text = dt.Rows[0]["id\_seansi"].ToString().Trim();

cmd = new SqlCommand("Update bilet set id\_sostojanie='2' where bilet.id\_mesto='" + Convert.ToInt16(label15.Text) + "' and bilet.id\_seans='" + Convert.ToInt16(label9.Text) + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

da.Update(dt);

con.Close();

string name = label1.Text;

string time = label4.Text;

string[] datetime = label3.Text.Split(' ');

string date = datetime[0];

int cost = Convert.ToInt16(label11.Text);

ticket t = new ticket(name, time, date, cost, rad, mesto);

t.Show();

bileti();

kolbilet();

proverka = 1;

}

}

if (proverka != 1)

{

if (dataGridView11.Rows[i].Cells[j].Style.BackColor != Color.Red)

{

if (dataGridView11.Rows[i].Cells[j].Style.BackColor != Color.Yellow)

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT seansi.id\_seansi FROM seansi where seansi.date='" + label3.Text + "' and seansi.time='" + label4.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

label9.Text = dt.Rows[0]["id\_seansi"].ToString().Trim();

da.Update(dt);

int mesto = Convert.ToInt16(dataGridView11.Rows[i].Cells[j].Value.ToString());

int rad = i + 1;

cmd = new SqlCommand("SELECT DISTINCT mesto.id\_mesto FROM mesto where mesto.rad='" + rad + "' and mesto.mesto='" + mesto + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

label10.Text = dt.Rows[0]["id\_mesto"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("insert into bilet(id\_seans,id\_mesto,id\_sostojanie) values ('" + Convert.ToInt16(label9.Text) + "','" + Convert.ToInt16(label10.Text) + "','2')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

da.Update(dt);

con.Close();

string name = label1.Text;

string time = label4.Text;

string[] datetime = label3.Text.Split(' ');

string date = datetime[0];

int cost = Convert.ToInt16(label11.Text);

ticket t = new ticket(name, time, date, cost, rad, mesto);

t.Show();

}

}

else { MessageBox.Show("Билет уже куплен !"); }

}

}

}

proverka = 0;

bileti();

kolbilet();

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

if (select != 0)

{

chistka();

dataGridView10.Rows[select].Selected = false;

select--;

dataGridView10.Rows[select].Selected = true;

label3.Text = dataGridView10.Rows[select].Cells[1].Value.ToString();

label4.Text = dataGridView10.Rows[select].Cells[2].Value.ToString();

bileti();

kolbilet();

}

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

if (select != dataGridView10.RowCount - 1)

{

dataGridView10.Rows[select].Selected = false;

select++;

if (select != dataGridView10.RowCount - 1)

{

dataGridView10.Rows[select].Selected = true;

label3.Text = dataGridView10.Rows[select].Cells[1].Value.ToString();

label4.Text = dataGridView10.Rows[select].Cells[2].Value.ToString();

chistka();

bileti();

kolbilet();

}

else

{

select--;

dataGridView10.Rows[select].Selected = true;

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

for (int i = 0; i < 9; i++)

for (int j = 0; j < 21; j++)

{

if (dataGridView11.Rows[i].Cells[j].Selected == true && dataGridView11.Rows[i].Cells[j].Value != null )

{

if (dataGridView11.Rows[i].Cells[j].Style.BackColor != Color.Yellow)

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT seansi.id\_seansi FROM seansi where seansi.date='" + label3.Text + "' and seansi.time='" + label4.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

label9.Text = dt.Rows[0]["id\_seansi"].ToString().Trim();

da.Update(dt);

int mesto = Convert.ToInt16(dataGridView11.Rows[i].Cells[j].Value.ToString());

int rad = i + 1;

cmd = new SqlCommand("SELECT DISTINCT mesto.id\_mesto FROM mesto where mesto.rad='" + rad + "' and mesto.mesto='" + mesto + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

label10.Text = dt.Rows[0]["id\_mesto"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("insert into bilet(id\_seans,id\_mesto,id\_sostojanie) values ('" + Convert.ToInt16(label9.Text) + "','" + Convert.ToInt16(label10.Text) + "','1')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

da.Update(dt);

con.Close();

}

else { MessageBox.Show("Билет уже забронирован!"); }

}

}

bileti();

kolbilet();

}

public void id()

{

for (int i = 0; i < 9; i++)

for (int j = 0; j < 21; j++)

{

if (dataGridView11.Rows[i].Cells[j].Selected == true && dataGridView11.Rows[i].Cells[j].Value != null)

{

con.Open();

int mesto = Convert.ToInt16(dataGridView11.Rows[i].Cells[j].Value.ToString());

int rad = i + 1;

cmd = new SqlCommand("SELECT DISTINCT mesto.id\_mesto FROM mesto where mesto.rad='" + rad + "' and mesto.mesto='" + mesto + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

label10.Text = dt.Rows[0]["id\_mesto"].ToString().Trim();

da.Update(dt);

con.Close();

}

}

}

public partial class addkino : Form

{

public static int ob=0;

private SqlConnection con;

private SqlCommand cmd;

private SqlDataAdapter da;

private DataTable dt;

BindingSource bis;

ToolStripLabel dateLabel;

ToolStripLabel timeLabel;

ToolStripLabel infoLabel;

Timer timer;

int prov = 0;

void timer\_Tick(object sender, EventArgs e)

{

dateLabel.Text = DateTime.Now.ToLongDateString();

timeLabel.Text = DateTime.Now.ToLongTimeString();

}

public addkino()

{

InitializeComponent();

con = new SqlConnection(@"Data Source=USERPC\SQLEXPRESS;Initial Catalog=dbcin;Integrated Security=True;Pooling=False");

comboBox4.Visible = false;

infoLabel = new ToolStripLabel();

infoLabel.Text = "Текущие дата и время:";

dateLabel = new ToolStripLabel();

timeLabel = new ToolStripLabel();

statusStrip1.Items.Add(infoLabel);

statusStrip1.Items.Add(dateLabel);

statusStrip1.Items.Add(timeLabel);

timer = new Timer() { Interval = 100 };

// timer = new Timer() { Interval = 1000 };

timer.Tick += timer\_Tick;

timer.Start();

label25.Visible = false;

label24.Visible = false;

comboBox7.Visible = false;

comboBox8.Visible = false;

label20.Visible = false;

label21.Visible = false;

comboBox5.Visible = false;

checkBox1.Visible = false;

textBox4.Visible = false;

dateTimePicker1.Visible = false;

dateTimePicker2.Visible = false;

textBox1.Visible = false;

}

public string jn;

public int index;

public void filmakter()

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.fname,akter.name FROM film,akter,uchastniki where uchastniki.id\_film=film.id\_film and uchastniki.id\_akter=akter.id\_akter ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

dataGridView1.Columns[0].HeaderText = "Фильм";

dataGridView1.Columns[1].HeaderText = "В ролях";

con.Close();

}

public void film()

{

con.Open();

cmd = new SqlCommand("SELECT fname FROM film ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView5.DataSource = bSource;

da.Update(dt);

con.Close();

comboBox4.Items.Clear();

for (int i = 0; i < dataGridView5.RowCount - 1; i++)

{ comboBox4.Items.Add(dataGridView5.Rows[i].Cells[0].Value); }

}

public void janr()

{

con.Open();

cmd = new SqlCommand("SELECT janr FROM janr ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView4.DataSource = bSource;

da.Update(dt);

con.Close();

comboBox2.Items.Clear();

for (int i = 0; i < dataGridView4.RowCount-1; i++)

{ comboBox2.Items.Add(dataGridView4.Rows[i].Cells[0].Value); }

}

public void indexf()

{

string fname = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[0].Value.ToString();

cmd = new SqlCommand("SELECT id\_film FROM film where film.fname= '" + fname + "' ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView3.DataSource = bSource;

index = Convert.ToInt16(dataGridView3.Rows[0].Cells[0].Value);

label6.Text = index.ToString();

da.Update(dt);

}

public void akter()

{

con.Open();

indexf();

cmd = new SqlCommand("SELECT DISTINCT akter.name FROM film,akter,uchastniki where uchastniki.id\_film='" + index + "' and uchastniki.id\_akter=akter.id\_akter ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView2.DataSource = bSource;

da.Update(dt);

// dt2();

dataGridView2.Columns[0].HeaderText = "В ролях";

con.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

if (radioButton1.Checked == true || radioButton2.Checked == true || radioButton3.Checked == true || radioButton4.Checked == true)

{

if (radioButton1.Checked)

{

if (textBox1.Text != "" && comboBox7.Text != "" && comboBox8.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox7.Text != "" && textBox8.Text != "" && comboBox1.Text != "" && comboBox2.Text != "" && comboBox3.Text != "")

{

ob = 1;

con.Open();

cmd = new SqlCommand("SELECT DISTINCT vozrast.id\_vozrast FROM vozrast where vozrast.vozrast='" + comboBox1.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label13.Text = dt.Rows[0]["id\_vozrast"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT janr.id\_janr FROM janr where janr.janr='" + comboBox2.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label14.Text = dt.Rows[0]["id\_janr"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT strana.proizvodstva FROM strana ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView6.DataSource = bSource;

da.Update(dt);

prov = 0;

for (int i = 0; i < dataGridView6.RowCount; i++)

{

if (dataGridView6.Rows[i].Cells[0].Value.ToString() == textBox7.Text)

{

cmd = new SqlCommand("SELECT DISTINCT strana.id\_proizvodstva FROM strana where strana.proizvodstva='" + textBox7.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label15.Text = dt.Rows[0]["id\_proizvodstva"].ToString().Trim();

da.Update(dt);

prov = 1;

}

else

{

if (i == dataGridView6.RowCount - 1 && prov != 1)

{

cmd = new SqlCommand("insert into strana (proizvodstva) values ('" + textBox7.Text + "')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT strana.id\_proizvodstva FROM strana where strana.proizvodstva='" + textBox7.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label15.Text = dt.Rows[0]["id\_proizvodstva"].ToString().Trim();

da.Update(dt);

}

}

}

prov = 0;

for (int i = 0; i < dataGridView6.RowCount; i++)

{

if (dataGridView6.Rows[i].Cells[0].Value.ToString() == textBox7.Text)

{

cmd = new SqlCommand("SELECT DISTINCT produser.id\_produser FROM produser where produser.name='" + textBox8.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label16.Text = dt.Rows[0]["id\_produser"].ToString().Trim();

da.Update(dt);

prov = 1;

}

else

{

if (i == dataGridView6.RowCount - 1 && prov != 1)

{

cmd = new SqlCommand("insert into produser (name) values ('" + textBox8.Text + "')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT produser.id\_produser FROM produser where produser.name='" + textBox8.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label16.Text = dt.Rows[0]["id\_produser"].ToString().Trim();

da.Update(dt);

con.Close();

}

}

}

cmd = new SqlCommand("SELECT DISTINCT format.id\_format FROM format where format.format='" + comboBox3.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label17.Text = dt.Rows[0]["id\_format"].ToString().Trim();

da.Update(dt);

label22.Text = dateTimePicker1.Value.ToShortDateString();

label23.Text = dateTimePicker2.Value.ToShortDateString();

cmd = new SqlCommand("insert into film (fname,long,start,finish,cost,id\_vozrastogr,id\_janr,id\_strana,id\_produser,id\_format,description) values ('" + textBox1.Text + "','" + comboBox7.Text + ":" + comboBox8.Text + ":00" + "','" + label22.Text + "','" + label23.Text + "','" + textBox3.Text + "','" + label13.Text + "','" + label14.Text + "','" + label15.Text + "','" + Convert.ToInt16(label16.Text) + "','" + Convert.ToInt16(label17.Text) + "','" + textBox4.Text + "')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

da.Update(dt);

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

else { MessageBox.Show("Введите данные !"); }

}

if (radioButton2.Checked)

{

if (comboBox7.Text != "" && comboBox8.Text != "" && comboBox4.Text != "")

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.id\_film FROM film where film.fname='" + comboBox4.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label11.Text = dt.Rows[0]["id\_film"].ToString().Trim();

da.Update(dt);

label12.Text = dateTimePicker1.Value.ToShortDateString();

cmd = new SqlCommand("insert into seansi(date,time,id\_film) values ('" + label12.Text + "','" + comboBox7.Text + ":" + comboBox8.Text + ":00" + "','" + Convert.ToInt16(label11.Text) + "')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

cmd = new SqlCommand("SELECT film.fname,seansi.date ,seansi.time FROM film,seansi where film.id\_film=seansi.id\_film ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

else { MessageBox.Show("Введите данные !"); }

}

if (radioButton4.Checked)

{

if (comboBox4.Text != "" && textBox2.Text != "")

{

con.Open();

con.Close();

con.Close();

con.Open();

ob = 1;

cmd = new SqlCommand("SELECT DISTINCT akter.name FROM film,akter,uchastniki ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView8.DataSource = bSource;

da.Update(dt);

for (int i = 0; i < dataGridView8.RowCount; i++)

{

if (dataGridView8.Rows[i].Cells[0].Value.ToString() == textBox2.Text)

{ break; }

else

{

if (i == dataGridView8.RowCount - 1)

{

cmd = new SqlCommand("insert into akter (name) values ('" + textBox2.Text + "')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

}

}

}

cmd = new SqlCommand("SELECT DISTINCT film.id\_film FROM film where film.fname='" + comboBox4.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

label11.Text = dt.Rows[0]["id\_film"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT akter.id\_akter FROM akter where akter.name = '" + textBox2.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

label12.Text = dt.Rows[0]["id\_akter"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT \* FROM uchastniki ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView8.DataSource = bSource;

da.Update(dt);

for (int i = 0; i < dataGridView8.RowCount; i++)

{

if (dataGridView8.Rows[i].Cells[0].Value.ToString() == label11.Text && dataGridView8.Rows[i].Cells[1].Value.ToString() == label12.Text)

{ break; }

else

{

if (i == dataGridView8.RowCount - 1)

{

cmd = new SqlCommand("insert into uchastniki values ('" + label11.Text + "','" + label12.Text + "')");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

con.Close();

filmakter();

}

}

}

}

else { MessageBox.Show("Введите данные !"); }

con.Close();

}

}

else { MessageBox.Show("Не выбрана операция"); }

}

private void button2\_Click(object sender, EventArgs e)

{

if (radioButton1.Checked)

{

con.Open();

ob = 1;

cmd = new SqlCommand("SELECT DISTINCT film.id\_film FROM film where film.fname='" + dataGridView1.CurrentRow.Cells[0].Value.ToString() + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label11.Text = dt.Rows[0]["id\_film"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand(" SELECT seansi.id\_seansi,film.fname,seansi.date ,seansi.time FROM film,seansi where film.id\_film=seansi.id\_film and seansi.id\_film='" + label11.Text + "' ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView7.DataSource = bSource;

da.Update(dt);

if (dataGridView7.RowCount != 0)

{

for (int i = 0; i < dataGridView7.RowCount; i++)

{

cmd = new SqlCommand("delete from bilet where bilet.id\_seans='" + dataGridView7.Rows[i].Cells[0].Value.ToString() + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

}

cmd = new SqlCommand("delete from seansi where id\_film='" + label11.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

cmd = new SqlCommand("delete from uchastniki where id\_film='" + label11.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

}

cmd = new SqlCommand("delete from film where id\_film='" + label11.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (radioButton2.Checked)

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.id\_film FROM film where film.fname='" + dataGridView1.CurrentRow.Cells[0].Value.ToString() + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label11.Text = dt.Rows[0]["id\_film"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT seansi.id\_seansi FROM seansi,film where seansi.id\_film = '" + label11.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

label12.Text = dt.Rows[0]["id\_seansi"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("delete from bilet where id\_seans='" + label12.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

cmd = new SqlCommand("delete from seansi where id\_seansi='" + label12.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

cmd = new SqlCommand("SELECT film.fname,seansi.date ,seansi.time FROM film,seansi where film.id\_film=seansi.id\_film ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (radioButton3.Checked)

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT mesto.id\_mesto FROM mesto where mesto.mesto='" + dataGridView1.CurrentRow.Cells[5].Value.ToString() + "' and mesto.rad='" + dataGridView1.CurrentRow.Cells[4].Value.ToString() + "' ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label11.Text = dt.Rows[0]["id\_mesto"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("delete from bilet where id\_mesto='" + label11.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT film.fname,seansi.date ,seansi.time,sostojanie.sostojanie,mesto.rad,mesto.mesto FROM film,sostojanie,seansi,bilet,mesto where film.id\_film=seansi.id\_film and seansi.id\_seansi=bilet.id\_seans and bilet.id\_sostojanie=sostojanie.id\_sostojanie and bilet.id\_mesto=mesto.id\_mesto ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

if (radioButton4.Checked)

{

con.Open();

ob = 1;

cmd = new SqlCommand("SELECT DISTINCT film.id\_film FROM film where film.fname='" + dataGridView1.CurrentRow.Cells[0].Value.ToString() + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

label11.Text = dt.Rows[0]["id\_film"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("SELECT DISTINCT akter.id\_akter FROM akter where akter.name = '" + dataGridView1.CurrentRow.Cells[1].Value.ToString() + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

label12.Text = dt.Rows[0]["id\_akter"].ToString().Trim();

da.Update(dt);

cmd = new SqlCommand("delete from uchastniki where id\_film='" + label11.Text + "' and id\_akter='" + label12.Text + "'");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

da.Update(dt);

con.Close();

filmakter();

}

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

label25.Visible = true;

label24.Visible = true;

comboBox7.Visible = true;

comboBox8.Visible = true;

label20.Visible = false;

label21.Visible = false;

comboBox5.Visible = false;

checkBox1.Visible = false;

button1.Visible = true;

textBox1.Visible = true;

textBox2.Visible = false;

textBox3.Visible = true;

textBox4.Visible = true;

textBox7.Visible = true;

textBox8.Visible = true;

comboBox1.Visible = true;

comboBox2.Visible = true;

comboBox3.Visible = true;

dateTimePicker1.Visible = true;

dateTimePicker2.Visible = true;

label2.Text = "Продолжительность сеанса";

label3.Text = "Дата начала проката";

label4.Text = "Дата окончания проката";

con.Open();

cmd = new SqlCommand("SELECT fname , long, start, finish, cost, vozrast, janr, proizvodstva , produser.name,format,description FROM film,vozrast,janr,strana,produser,format where film.id\_vozrastogr=vozrast.id\_vozrast and film.id\_janr=janr.id\_janr and film.id\_strana=strana.id\_proizvodstva and film.id\_produser=produser.id\_produser and film.id\_format=format.id\_format ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

dataGridView1.Columns[0].HeaderText = "Название фильма";

dataGridView1.Columns[1].HeaderText = "Длительность";

dataGridView1.Columns[2].HeaderText = "Дата начала проката";

dataGridView1.Columns[3].HeaderText = "Дата окончания проката";

dataGridView1.Columns[4].HeaderText = "Цена билета";

dataGridView1.Columns[5].HeaderText = "Возрастное ограничение";

dataGridView1.Columns[6].HeaderText = "Жанр";

dataGridView1.Columns[7].HeaderText = "Производство";

dataGridView1.Columns[8].HeaderText = "Продюсер";

dataGridView1.Columns[9].HeaderText = "Формат";

dataGridView1.Columns[10].HeaderText = "Описание";

con.Close();

janr();

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

textBox2.Visible = false;

label25.Visible = false;

label24.Visible = true;

comboBox7.Visible = true;

comboBox8.Visible = true;

label20.Visible = false;

label21.Visible = false;

comboBox5.Visible = false;

checkBox1.Visible = false;

label1.Visible = true;

label3.Text = "Дата начала сеанса";

label2.Text = "Время начала сеанса";

con.Open();

cmd = new SqlCommand("SELECT film.fname,seansi.date ,seansi.time FROM film,seansi where film.id\_film=seansi.id\_film ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

dataGridView1.Columns[0].HeaderText = "Название фильма";

dataGridView1.Columns[1].HeaderText = "Дата начала сеанса";

dataGridView1.Columns[2].HeaderText = "Время начала сеанса";

con.Close();

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

label25.Visible = false;

label20.Visible = true;

label21.Visible = true;

comboBox5.Visible = true;

checkBox1.Visible = true;

label24.Visible = false;

comboBox7.Visible = false;

comboBox8.Visible = false;

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.fname,seansi.date ,seansi.time,sostojanie.sostojanie,mesto.rad,mesto.mesto FROM film,sostojanie,seansi,bilet,mesto where film.id\_film=seansi.id\_film and seansi.id\_seansi=bilet.id\_seans and bilet.id\_sostojanie=sostojanie.id\_sostojanie and bilet.id\_mesto=mesto.id\_mesto ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

dataGridView1.Columns[0].HeaderText = "Название фильма";

dataGridView1.Columns[1].HeaderText = "Дата начала сеанса";

dataGridView1.Columns[2].HeaderText = "Время начала сеанса";

dataGridView1.Columns[3].HeaderText = "Состояние";

dataGridView1.Columns[4].HeaderText = "Ряд";

dataGridView1.Columns[5].HeaderText = "Место";

con.Close();

}

private void dataGridView1\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

akter();

}

private void comboBox4\_MouseDown(object sender, MouseEventArgs e)

{

film();

}

private void radioButton4\_CheckedChanged(object sender, EventArgs e)

{

filmakter();

label25.Visible = false;

label24.Visible = false;

comboBox7.Visible = false;

comboBox8.Visible = false;

label1.Visible = true;

comboBox4.Visible = true;

textBox2.Visible = true;

label2.Visible = true;

label2.Text = "Актер";

button1.Visible = true;

button2.Visible = true;

}

private void comboBox5\_MouseDown(object sender, MouseEventArgs e)

{

con.Open();

cmd = new SqlCommand("SELECT fname FROM film ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView5.DataSource = bSource;

da.Update(dt);

con.Close();

comboBox5.Items.Clear();

for (int i = 0; i < dataGridView5.RowCount - 1; i++)

{ comboBox5.Items.Add(dataGridView5.Rows[i].Cells[0].Value); }

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (!checkBox1.Checked)

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.fname,seansi.date ,seansi.time,sostojanie.sostojanie,mesto.rad,mesto.mesto FROM film,sostojanie,seansi,bilet,mesto where film.id\_film=seansi.id\_film and seansi.id\_seansi=bilet.id\_seans and bilet.id\_sostojanie=sostojanie.id\_sostojanie and bilet.id\_mesto=mesto.id\_mesto ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

}

private void comboBox5\_SelectedValueChanged(object sender, EventArgs e)

{

con.Open();

cmd = new SqlCommand("SELECT DISTINCT film.fname,seansi.date ,seansi.time,sostojanie.sostojanie,mesto.rad,mesto.mesto FROM film,sostojanie,seansi,bilet,mesto where film.id\_film=seansi.id\_film and seansi.id\_seansi=bilet.id\_seans and bilet.id\_sostojanie=sostojanie.id\_sostojanie and bilet.id\_mesto=mesto.id\_mesto and film.fname= '" + comboBox5.Text + "' ");

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

dt = new DataTable();

da.Fill(dt);

BindingSource bSource = new BindingSource();

bSource.DataSource = dt;

dataGridView1.DataSource = bSource;

da.Update(dt);

con.Close();

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar) && e.KeyChar != Convert.ToChar(8) && !((e.KeyChar == '.' )))

{

e.Handled = true;

}

}

}

}

public partial class user : Form

{

public static List<registrationclass> rg = new List<registrationclass>();

XmlSerializer formatter = new XmlSerializer(typeof(List<registrationclass>));

public user()

{

InitializeComponent();

if (Form1.a == 0) { menuStrip1.Visible = false; }

if (Form1.a == 1) { menuStrip1.Visible = true; }

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

label1.Text = textBox1.Text;

}

private void button1\_Click(object sender, EventArgs e)

{

using (FileStream fs = new FileStream("pas", FileMode.OpenOrCreate))

{

rg = (List<registrationclass>)formatter.Deserialize(fs);

foreach (registrationclass rgn in rg)

{

if (textBox1.Text == rgn.Login && textBox2.Text == rgn.Password)

{

MessageBox.Show("Авторизация прошла успешно");

Form1.a = 1;

menuStrip1.Visible = true;

}

else { MessageBox.Show("Авторизация не прошла"); }

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

Form1.a = 0;

button3.Visible = false;

this.Close();

}

private void button3\_Click(object sender, EventArgs e)

{

string Login = textBox1.Text;

string Password = textBox2.Text;

rg.Clear();

rg.Add(new registrationclass(Login, Password));

using (FileStream fs = new FileStream("pas", FileMode.Create))

{

formatter.Serialize(fs, rg);

fs.Close();

}

}

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

{

MessageBox.Show("Введите новые параметры");

foreach (registrationclass rgn in rg)

{

textBox1.Text = rgn.Login;

textBox2.Text = rgn.Password;

button3.Visible = true;

}

}

}

public partial class ticket : Form

{

Bitmap ticketimage;

public ticket(string name, string time, string date, int cost, int rad, int mesto)

{

InitializeComponent();

ticketimage = new Bitmap("Билет2.jpg");

pictureBox1.Size = ticketimage.Size;

CreateGraphics().DrawImage(ticketimage, new Point(0, 0));

drawticket(name, time, date, cost, rad, mesto);

}

private void drawticket(string name, string time, string date, int cost, int rad, int mesto)

{

Graphics g = Graphics.FromImage(ticketimage);

g.DrawString(name, new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(340f, 40f));

// g.DrawString(Convert.ToString(Count), new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(220f, 62f));

g.DrawString(time, new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(324f, 85f));

g.DrawString(date, new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(150f, 160f));

g.DrawString(Convert.ToString(cost), new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(200f, 114f));

g.DrawString(Convert.ToString(rad), new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(178f, 88f));

g.DrawString(Convert.ToString(mesto), new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(242f, 88f));

g.DrawString("Кинотеатр Беларусь", new Font("Microsoft Sans Serif", 8f), new SolidBrush(Color.Black), new PointF(280f, 160f));

pictureBox1.Image = ticketimage;

}

private void печатьToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

printPreviewDialog1.ShowDialog();

}

private void printDocument1\_PrintPage\_1(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawImage(ticketimage, new Point(0, 0));

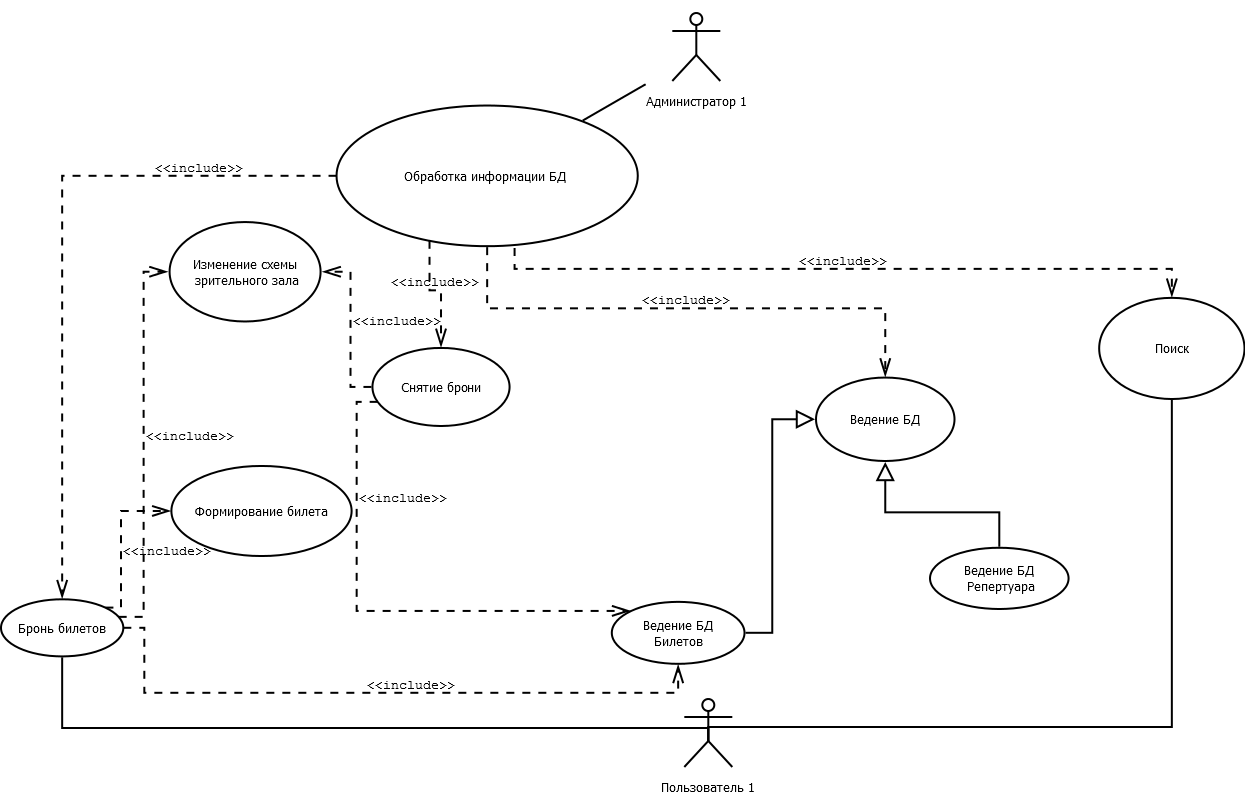
}

}

Приложение Б

(обязательное)

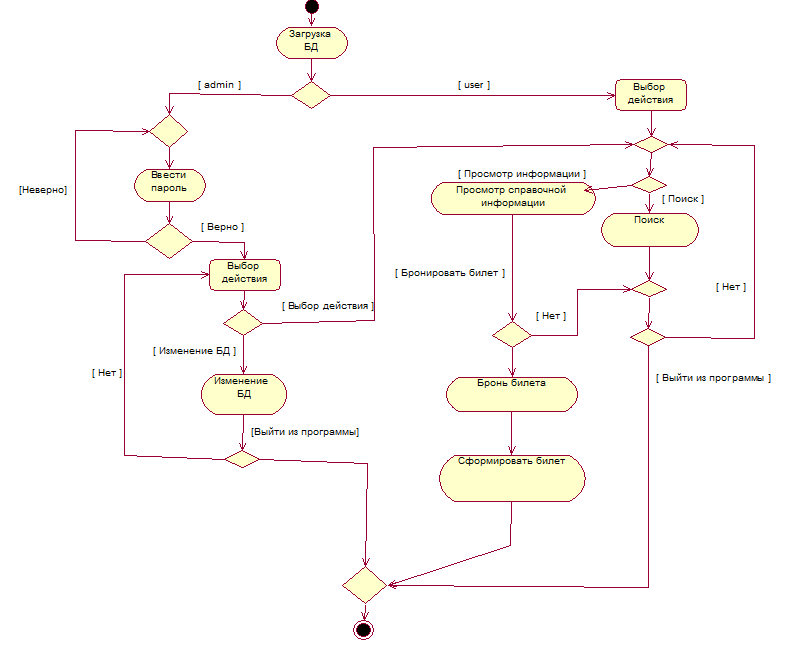
Диаграмма вариантов использования



Приложение В

(обязательное)

Диаграмма деятельности



Приложение Г

Блок-схема

Начало

Создание окна приложения и его элементов

Авторизоваться?

Да

Нет

Забронировать билет?

Нет

Да

Авторизация

Добавить сеанс?

Да

Нет

Просмотр информации

Бронь билета

Редактирование информации

Добавление сеанса

Нет

Да

Выйти?

Продолжить работу?

Нет

Да

Конец