# МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное бюджетное учреждение

"Слободской колледж педагогики и социальных отношений"

## КУРСОВОЙ ПРОЕКТ

по ПМ 01 «Разработка программных модулей» на тему:

# РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ АВТОМАТИЗАЦИИ ПРОДАЖИ БИЛЕТОВ В КИНОТЕАТРЕ

Выполнил: Платунов Павел

Андреевич

Специальность 09.02.07

Информационные системы и

программирование

Группа 21П-1

Форма обучения: очная

Руководитель: Калинин

Арсений Олегович

Дата защиты курсового проекта:

Председатель ПЦК:

Оценка за защиту курсового проекта:

Слободской

2024

# ОГЛАВЛЕНИЕ

Введение	3
1. Анализ предметной области	5
2. Разработка технического задания	12
3. Описание алгоритмов и схема функционирования программного модуля	16
4. Тестирование программного модуля	22
Заключение	25
Список литературы	26
Приложения	28

## **ВВЕДЕНИЕ**

В настоящее время достаточно популярным местом отдыха являются кинотеатры. Каждый день множество людей посещают различные кинотеатры для просмотра как недавно, так и давно вышедших фильмов, мультфильмов и других кинокартин. Из-за большого количества клиентов могут возникать очереди при продаже билетов на фильмы. Все это зависит от способности кассира быстро и эффективно работать, а также от наличия современных систем автоматизации.

Раньше продажа билетов производилась только в физическом виде, что занимало хоть и мало времени на заполнение одного билета, но при наплыве клиентов могло уходить очень много времени на обслуживание всей толпы. Тоже самое можно сказать и про сбор и анализ данных при работе кинотеатра, когда нужно было записывать количество проданных билетов и их цену на разные фильмы для того, чтобы в конце рабочего дня подвести итоги продаж и решить является ли показываемый фильм коммерчески выгодным. Сейчас же мы имеем возможность автоматизировать данные бизнес-процессы для повышения эффективности работы кассира и администрации кинотеатра.

Среди основных преимуществ такой автоматизации можно отметить:

- Сокращение времени обслуживания клиентов за счет автоматизации регистрации на сеанс по выбранному фильму и автоматизации печати билетов на выбранный клиентом фильм;
- Увеличение количества обрабатываемых клиентов, которое происходит за счет повышения эффективности работы кассира и сокращения времени обслуживания клиента;
- Повышение удобства клиента за счет уменьшения очередей из людей, которые хотят просмотреть фильм;
- Снижение затрат на содержание персонала за счет уменьшения количества кассиров, ведь при введении систем автоматизации эффективность

работы кассиров увеличиться, а время, затрачиваемое на обслуживание одного клиента уменьшиться;

• Сбор и аналитика данных также будут гораздо быстрее, эффективней, точней, а что самое важное удобней, ведь с введением автоматизации теперь можно отслеживать проданные билеты как на фильмы, так и на отдельный сеанс данного фильма для составления вывода о успешности показываемой кинокартины.

Цель курсового проекта – разработка программного обеспечения для автоматизации продажи билетов на фильмы в кинотеатре.

Задачи исследования:

- Описать предметную область.
- Разработать технического задание на создание программного продукта.
  - Описать архитектуру программы.
  - Описать алгоритмы и функционирование программы.
  - Провести тестирование и опытную эксплуатацию.
  - Разработать руководство оператора

Объект исследования – процесс продажи и учета проданных билетов в кинотеатре.

Предмет исследования – разработка программного модуля для автоматизации продажи билетов в кинотеатре.

Методы исследования: системный анализ и функциональное моделирование.

Информационную систему исследования составили официальные нормативно-правовые источники, данные об использовании современных информационных систем. Структура работы состоит из введения, трех глав, заключения, списка используемой литературы и приложений.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Кинотеатр — общественное здание или его часть с оборудованием для публичной демонстрации кинофильмов. И для его успешного функционирования необходимо выполнять множество различных функций таких как добавление новых фильмов, составление сеансов по фильмам, продажа билетов, подготовка и показ кинолент, и анализ данных для определения успешности фильма. Успешное функционирование системы зависит от четкой организации работы персонала и эффективного взаимодействия между его различными ролями.

Разрабатываемый программный модуль должен обладать возможностями выбора фильма и конкретного сеанса, на который посетитель хочет сходить, также обеспечить возможность выбора места в зале и печать билета после бронирования места. Еще программный модуль должно обеспечить базовый набор функциональности для обеспечения работы модуля без работы пользователя с базой данных на прямую через сторонние системы управления базой данных такие как SQL Server Management Studio, а также иметь возможность собирать и анализировать данные, полученные с продажи билетов для определения успешности фильма. Дополнительно стоит добавить возможность настройки конфигурации зала для большей универсальности программного модуля.

Для выполнения всех этих немало важных функций необходим обученный персонал:

- Администратор управляет системой, работает с базой данных, настраивает конфигурацию зала, управляет персоналом кинотеатра.
- Директор управляет персоналом, отслеживает продажи билетов на фильмы.
- Букер (менеджер по кинопрокату) добавляет новые фильмы, составляет и корректирует расписание сеансов, отслеживает продажи билетов на фильмы.

• Кассир – работает с расчетно-кассовым оборудованием, продает билеты на фильмы.

Для более четкого понятия рабочих задач персонала была составлена диаграмма вариантов использования (Рисунок 1).

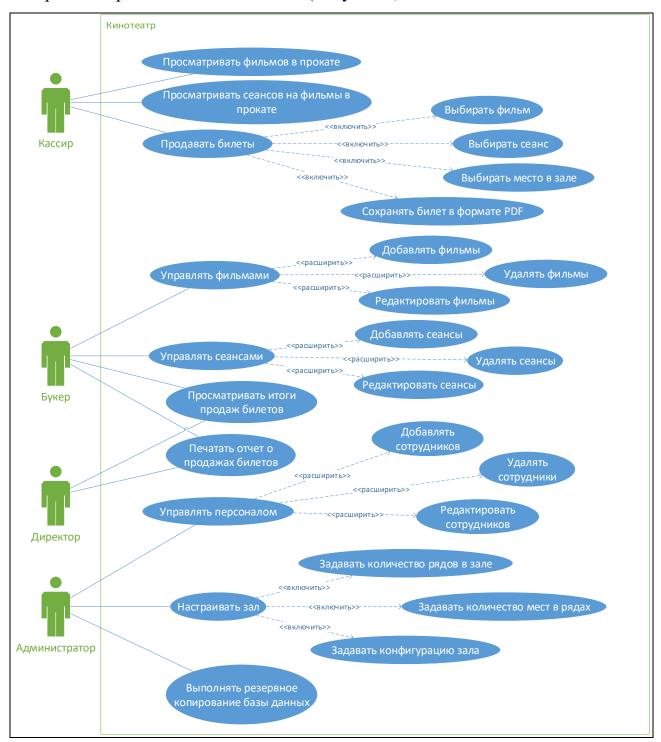


Рисунок 1 - диаграмма вариантов использования

По выделенным объектам и атрибутам была составлена ER – диаграмма показывающая структура создаваемой базы данных (Рисунок 2).

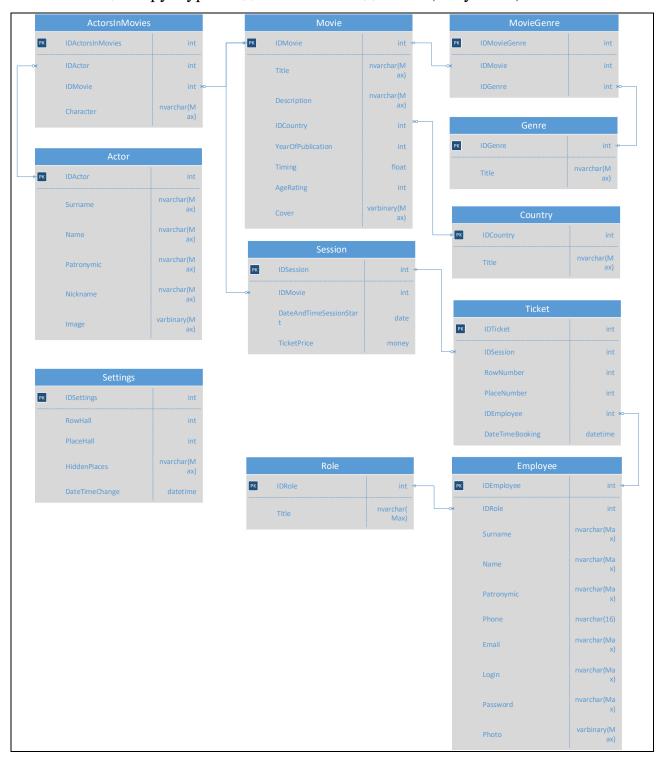


Рисунок 2 - ER диаграмма

С помощью построенной ER диаграммы, а также выделенным объектам и атрибутам была создана база данных (Рисунок 3).

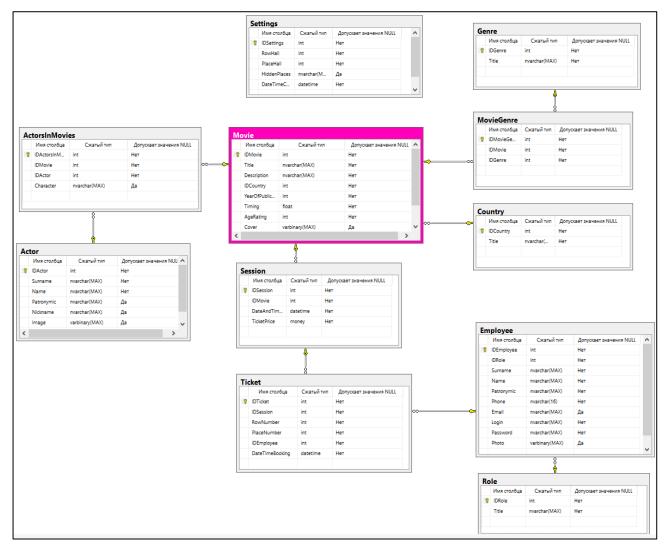


Рисунок 3 - Диаграмма базы данных

## Программы аналоги:

InTickets - это платформа для продажи билетов онлайн. Она предоставляет инструменты для создания мероприятий, управления билетами, продажи, обработки платежей и анализа данных. InTickets ориентирована на организаторов событий всех размеров, от небольших концертов до крупных фестивалей (Рисунок 4).

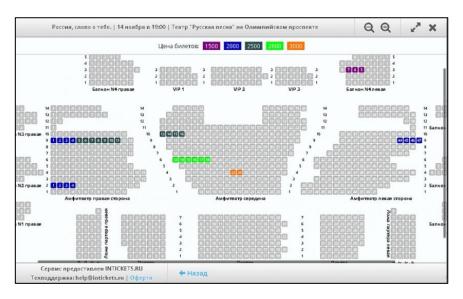


Рисунок 4 − InTickets

#### Преимущества:

- Есть мобильное приложение
- Гибкая настройка приложении позволяет адаптировать систему под специфику кинотеатра.
  - Интеграция с различными системами такими как 1С, Bitrix.
  - Стабильная работа
  - Быстрая и качественная техническая поддержка

#### Недостатки:

- Высокая цена покупки с предварительной договоренностью с поставщиком
  - Сложный в понимание интерфейс
  - Ограниченное количество функций в бесплатной версии

ТіскеtTool.net - это онлайн-сервис для создания и продажи билетов на события. Он предлагает простой и интуитивный интерфейс для создания мероприятий, установки цен на билеты, настройки дизайна и интеграции с различными платежными системами. TicketTool.net подходит для организаторов небольших мероприятий, таких как концерты, спектакли, семинары и т. д. (Рисунок 5).

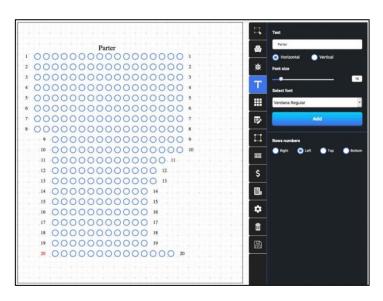


Рисунок 5 - TicketTool.net

#### Преимущества:

- Низкая стоимость программы 2,5% с продаж
- Простой интерфейс
- Удобный функционал позволяет легко продавать билеты, управлять расписанием сеансов и отслеживать продажи.
  - Мобильная версия
  - Бесплатная пробная демо версия

#### Недостатки:

- Ограниченные возможности настройки
- Меньше интеграций чем у InTicket
- Техническая поддержка своим качеством может варьироваться в зависимости от купленного плана

Тіскеt Тесh - это компания, которая разрабатывает и предлагает программное обеспечение для управления билетами. Их программное обеспечение позволяет организаторам событий продавать билеты онлайн, на месте и через мобильные приложения. Тіскеt Тесh фокусируется на обеспечении комплексного решения для управления билетами, которое включает в себя функции, такие как управление местами, управление запасами, обработка платежей и отчетность (Рисунок 6).

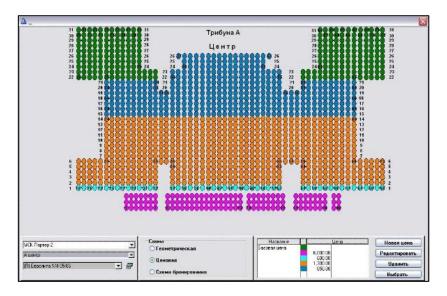


Рисунок 6 - Ticket Tech

## Преимущества:

- Предлагает передовые решения, например, автоматизированный маркетинг и анализ данных
  - Высокая скорость обработки данных
  - Позволяет легко продвигать кинотеатр в социальных сетях
  - Обеспечивает высокую степень защиты информации
  - Многоязычная поддержка

### Недостатки:

- Может быть очень дорогой до 12% с продаж
- Сложность настройки
- Ограниченное количество интеграций
- Отсутствие бесплатной пробной версии

# 2. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ

Наименование программы – «Сіпета». Программа предназначена для автоматизации работы кинотеатра ГОСТ 19.101-77 [1], ГОСТ 19.201-78 [2].

Разработка программы ведется на основании учебного плана и перечня тем утвержденных на заседании предметно цикловой комиссии информатики и программирования.

Функциональным назначением программы является автоматизация продажи билетов в кинотеатре.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- Управление сотрудниками;
- Управление фильмами;
- Управление сеансами;
- Настройка конфигурации зала;
- Расчет количества проданных билетов на фильмы;
- Расчет количества проданных билетов на сеансы выбранного фильма;
- Сохранение отчета по количеству проданных билетов на фильмы в файл формата pdf;
- Сохранение отчета по количеству проданных билетов на сеансы выбранного фильма в файл формата pdf;
  - Продажа билетов на выбранный сеанс фильма;
- Открытие pdf файла проданного билета для его печати или сохранения;

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнение заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

• организация бесперебойного питания технических средств;

- использование лицензионного программного обеспечения;
- отсутствие вредоносного программного обеспечения, наличие антивирусной программы;
- соблюдение правил и требований по эксплуатации технических средств.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 5 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий себя:

- процессор с тактовой частотой, 1 ГГц, не менее;
- оперативную память объемом 512 Мб, не менее;
- жесткий диск со свободным местом 500 Мб, не менее;
- монитор, с разрешением экрана 1366\*768, не менее;
- оптический привод;
- компьютерная мышь;
- клавиатура;

Исходные коды программы должны быть реализованы на языке С#. В качестве интегрированной среды разработки программы должна быть использована среда программирования Microsoft Visual Studio 2022 и Microsoft SQL Server 2014 Management Studio.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 7/8/10/11.

Программное обеспечение поставляется в виде изделия на CD диске либо USB накопителя.

Упаковка программного изделия должна осуществляться в упаковочную тару предприятия-изготовителя компакт диска или USB накопителя.

Требования к транспортировке и хранению должны соответствовать условиям эксплуатации носителей, на которых находится программный продукт.

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса.

Предварительный состав программной документации включает в себя следующие документы:

- техническое задание;
- руководство оператора.

Разработка должна быть проведена в следующие стадии и этапы:

1. Анализ требований:

На стадии анализ требований формулируются цели и задачи проекта. Создается основа для дальнейшего проектирования

### 2. Проектирование:

На стадии проектирование должны быть выполнены перечисленные ниже этапы работ:

• разработка программной документации;

На этапе разработка программной документации должна быть выполнена разработка технического задания.

При разработке технического задания должны быть выполнены перечисленные работы: постановка задачи, определение и уточнение требований к техническим средствам, определение требований к программе, определение стадий, этапов и сроков разработки программы и документации на нее, выбор языков программирования.

• разработка алгоритма программы;

На этапе разработки алгоритма программы должен быть разработан алгоритм работы программы.

#### • кодирование;

На стадии кодирования происходит реализация алгоритмов в среде программирования.

• тестирование и отладка.

На стадии тестирование и отладка происходит проверка алгоритмов, реализованных в программе на работоспособность в различных ситуациях. Исправление выявленных ошибок, повторное тестирование.

Приемо-сдаточные испытания должны проводиться при использовании технических средств. Приемка программы заключается в проверке работоспособности программы путем ввода реальных или демонстрационных данных.

Во время приемки работы разработчик предоставляет программу и документацию, которая к ней прилагается. Проводятся испытания программы, при успешных испытаниях программа вводится в эксплуатацию. При ошибках, недопустимых для успешной работы программного продукта — отправляется на доработку.

Было описано техническое задание, содержащее в себе информацию о программном продукте, его функциях, эксплуатации и требования, которые должны учитываться при создании программы и документации к ней.

# 3. ОПИСАНИЕ АЛГОРИТМОВ И СХЕМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО МОДУЛЯ

Наименование программы – «Сіпета». Программа предназначена для автоматизации продажи билетов на сеансы фильмов.

Функциональным назначением программы является автоматизация процесса продажи билетов.

Перед продажей билета на сеанс фильма кассир должен выбрать место в зале, после выбора места в зале информация о нем отобразиться на информационной панели "Бронирование билета" (Рисунок 7).

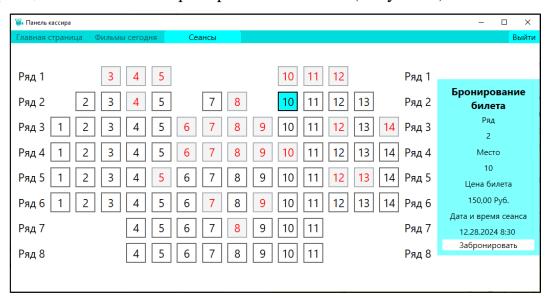


Рисунок 7 – Окно продажи билета на выбранный сеанс

Для отображения выбранного места в зале используется метод «HallButton\_Click» данный метод определяет нажатую кнопку меняет её стилистику и выводит данные о выбранном месте на информационную панель (Рисунок 8,9).

Рисунок 8 – Код метода HallButton Click

```
if (selectedRowPlaceButton != null)
       selectedRowPlaceButton.Background = Brushes.White;
      selectedRowPlaceButton.Foreground = Brushes.Black;
      selectedRowPlaceButton.BorderBrush = (Brush)(new BrushConverter().ConvertFrom("#FF707070"));
    Match match = Regex.Match(clickedButton.Name, @"Row(\d+)Place(\d+)");
    if (match.Success)
       selectedRowNumber = int.Parse(match.Groups[1].Value) + 1;
      selectedPlaceNumber = int.Parse(match.Groups[2].Value);
      clickedButton.Background = Brushes.Aqua;
      clickedButton.Foreground = Brushes.Black;
      clickedButton.BorderBrush = Brushes.Black;
       selectedRowPlaceButton = clickedButton;
    SelectedPlaceChange();
}
catch (Exception ex)
  MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
```

Рисунок 9 – Продолжение метода HallButton\_Click

Поле выбора места в зале кассир нажимает на кнопку "Забронировать" и подтверждает свой выбор (Рисунок 10)

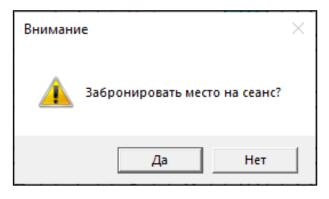


Рисунок 10 – Окно подтверждения бронирования

В результате перед кассиром откроется документ формата PDF с готовым к печати или сохранению билетом (Рисунок 11).

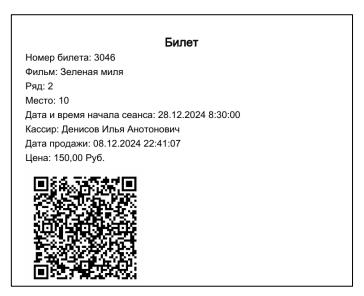


Рисунок 11 – Билет на выбранный сеанс фильма

Алгоритм работы модуля продажи билетов приведен ниже (Рисунок 12).

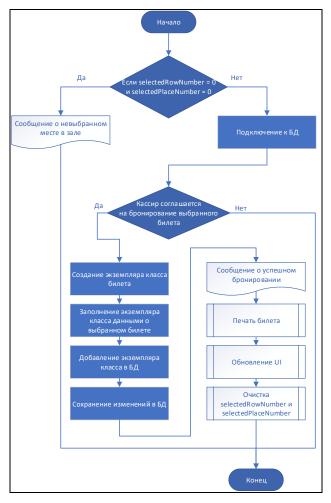


Рисунок 12 - Блок схема функции по продаже билетов на выбранный сеанс фильма

Для сохранения информации о забронированном месте на выбранный сеанс кино используется метод «BookingTicket\_Click», данный метод проверяет выбрал ли кассир место в зале, если нет то выводит соответствующее сообщение, в противном случает выводит сообщение пользователю для подтверждения бронирования билета в случает согласия кассира метод создает экземпляр класса Тicket в который записывает выбранное место и ряд в зале, сотрудника осуществляющего данную операцию, а также дату и время бронирования билета (Рисунок 13,14).

```
private void BookingTicket_Click(object sender, RoutedEventArgs e)
  try
    if (selectedRowNumber == 0 && selectedPlaceNumber == 0)
      MessageBox.Show("Место
                                                     выбранно",
                                                                   "Внимание",
                                                                                  MessageBoxButton.OK,
MessageBoxImage.Warning);
      return:
    using (var dataBase = new CinemaEntities())
      if (MessageBox.Show("Забронировать место на сеанс?", "Внимание", MessageBoxButton.YesNo,
MessageBoxImage.Warning) == MessageBoxResult.Yes)
         var newTicket = new Ticket();
        newTicket.IDSession = TransmittedData.idSelectedCashierSession;
        newTicket.RowNumber = selectedRowNumber;
        newTicket.PlaceNumber = selectedPlaceNumber;
        newTicket.IDEmployee = TransmittedData.idEmployee;
        newTicket.DateTimeBooking = DateTime.Now;
         dataBase.Ticket.Add(newTicket);
         dataBase.SaveChanges();
         MessageBox.Show("Mecto
                                                  зарезервировано",
                                                                      "Готово",
                                                                                  MessageBoxButton.OK,
                                          сеанс
MessageBoxImage.Information);
         Task.Run(() => PrintTicket(newTicket.IDTicket));
         selectedRowPlaceButton.Background = (Brush)(new BrushConverter().ConvertFrom("#FFDDDDDD"));
         selectedRowPlaceButton.BorderBrush = (Brush)(new BrushConverter().ConvertFrom("#FF707070"));
         selectedRowPlaceButton.Foreground = Brushes.Red;
         selectedRowPlaceButton.IsEnabled = false;
         ClearPlaceChange();
catch (Exception ex)
```

Рисунок 13 – Код метода BookingTicket\_Click

```
{
    MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
}
}
```

Рисунок 14 – Продолжение метода BookingTicket\_Click

После сохранения информации в базе данных запускается метод «PrintTicket» (Рисунок 15,16). Данный метод предназначен для вывода билета в PDF формате кассиру. Данный метод использует библиотеку «ITextShape» для формирования PDF файла и библиотеку «QR Coder» для формирования QR кода.

```
private async Task PrintTicket(int selectedTicket)
  try
    using (var dataBase = new CinemaEntities())
       var ticketData = (from ticket in dataBase.Ticket
                 session in dataBase.Session on ticket.IDSession equals session.IDSession into sessionGroup
                 from session in sessionGroup.DefaultIfEmpty()
                 movie in dataBase.Movie on session.IDMovie equals movie.IDMovie into movieGroup
                 from movie in movieGroup.DefaultIfEmpty()
                 employee in dataBase.Employee on ticket.IDEmployee equals employee.IDEmployee into
employeeGroup
                 from employee in employeeGroup.DefaultIfEmpty()
                 where (ticket.IDTicket == selectedTicket)
                 select new
                   ticket.IDTicket,
                   ticket.RowNumber,
                   ticket.PlaceNumber,
                   ticket.DateTimeBooking,
                   movie = movie. Title,
                   session.DateAndTimeSession,
                   employee.Surname,
                   employee.Name,
                   employee.Patronymic,
                   session.TicketPrice
                 }).FirstOrDefault();
      string tempFilePath = Path.GetTempFileName() + ".pdf";
       string
                                                      qrCodeText
$"\ticketData.IDTicket\|\ticketData.movie\|\ticketData.DateAndTimeSession\|\ticketData.Surname\}
{ticketData.Name} {ticketData.Patronymic}|{Math.Round(ticketData.TicketPrice,2)}";
       QRCodeGenerator qrGenerator = new QRCodeGenerator();
       QRCodeData qrCodeData = qrGenerator.CreateQrCode(qrCodeText, QRCodeGenerator.ECCLevel.Q);
       QRCode qrCode = new QRCode(qrCodeData);
       (System.Drawing.Bitmap
                                  qrCodeBitmap = qrCode.GetGraphic(20,
                                                                                System.Drawing.Color.Black,
System.Drawing.Color.White, null))
```

Рисунок 15 – Код метода PrintTicket

```
using (MemoryStream ms = new MemoryStream())
           qrCodeBitmap.Save(ms, System.Drawing.Imaging.ImageFormat.Png); // Сохраняем в PNG
           byte[] qrCodeBytes = ms.ToArray();
           Document doc = new Document();
           PdfWriter writer = PdfWriter.GetInstance(doc, new FileStream($@"{tempFilePath}", FileMode.Create));
           BaseFont baseFont = BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf", BaseFont.IDENTITY_H,
BaseFont.NOT_EMBEDDED);
           Font font = new Font(baseFont, 16);
           BaseFont
                            baseFontHead
                                                          BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf",
BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
           Font fontHead = new Font(baseFont, 20, Font.BOLD);
           doc.Open();
           Paragraph mainParagraph = new Paragraph("Билет", fontHead);
           mainParagraph.Alignment = Element.ALIGN CENTER;
           Paragraph paragraph = new Paragraph("Номер билета: " + ticketData.IDTicket, font);
           Paragraph paragraph1 = new Paragraph("Фильм: " + ticketData.movie, font);
           Paragraph paragraph2 = new Paragraph("Ряд: " + ticketData.RowNumber, font);
           Paragraph paragraph3 = new Paragraph("Mecтo: " + ticketData.PlaceNumber, font);
           Paragraph
                       paragraph4
                                          new
                                                 Paragraph("Дата
                                     =
                                                                   И
                                                                        время
                                                                                            сеанса:
ticketData.DateAndTimeSession, font);
           Paragraph paragraph5 = new Paragraph("Кассир: " + ticketData.Surname + " " + ticketData.Name +
+ ticketData.Patronymic, font);
           Paragraph paragraph6 = new Paragraph("Дата продажи: " + ticketData.DateTimeBooking, font);
           Paragraph
                             paragraph7
                                                          new
                                                                       Paragraph("Цена:
ticketData.TicketPrice.ToString().Remove(ticketData.TicketPrice.ToString().Length - 2, 2) + " Py6.", font);
           doc.Add(mainParagraph);
           doc.Add(paragraph);
           doc.Add(paragraph1);
           doc.Add(paragraph2);
           doc.Add(paragraph3);
           doc.Add(paragraph4);
           doc.Add(paragraph5);
           doc.Add(paragraph6);
           doc.Add(paragraph7);
           iTextSharp.text.Image image = iTextSharp.text.Image.GetInstance(qrCodeBytes);
           image.ScaleToFit(200, 200);
           doc.Add(image);
           doc.Close();
       Process.Start(tempFilePath);
  }
  catch (Exception ex)
    MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
```

Рисунок 16 – Продолжение метода PrintTicket

Полный код программы можно просмотреть в Приложении 1.

## 4. ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

Для проведения тестирования программы мною было произведено базовое тестирование во время разработки программы. При тестировании был выявлен ряд ошибок, которые возникли в ходе выполнения программы.

• Попытка бронирования билета без выбора места в зале Без выбора места в зале нажатие на кнопку "Забронировать"

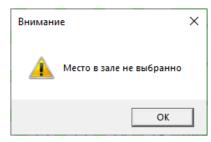


Рисунок 17 - Ошибка бронирования места

Ожидаемый результат: Сообщение о невозможности забронировать билет так как не выбрано место в зале.

Фактический результат: Сообщение о невозможности забронировать билет так как не выбрано место в зале. (Рисунок 17)

Решение проблемы: для вывода сообщения о невыбранном месте в зале реализована проверка значений выбранного ряда и места в зале.

• Попытка бронирования билета на занятое место
Попытка выбора места в зале, на который уже забронирован билет

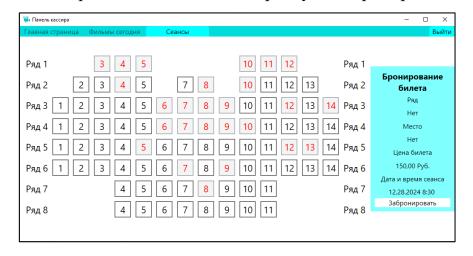


Рисунок 18 - Невозможность бронирования билета на занятое место

Ожидаемый результат: Невозможность выбора ряда и места в зале, на который уже куплен билет.

Фактический результат: Невозможность выбора ряда и места в зале, на который уже куплен билет. (Рисунок 18)

Решение проблемы: для решения данные проблемы места, на которые были забронированные билеты выделяются красным цветом, а также отключается взаимодействие с ними.

• Попытка забронировать билет повторно после бронирования

После бронирования билета на свободное место нажать на кнопку бронирования 2-й раз.

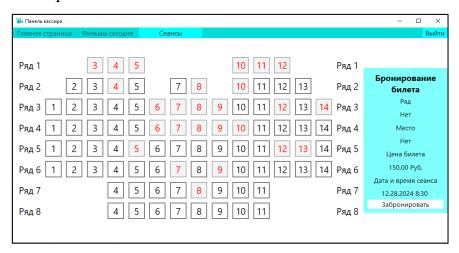


Рисунок 19 - Блокировка бронирования билета повторно

Ожидаемый результат: Значения ряда и места для бронирования на них билета сбрасываются, а в пользовательском интерфейсе место меняет цвет на красный, а также отключается взаимодействие с ним.

Фактический результат: Значения ряда и места для бронирования на них билета сбрасываются, а в пользовательском интерфейсе место меняет цвет на красный, а также отключается взаимодействие с ним. (Рисунок 19)

Решение проблемы: Сброс значений ряда и места и блокировка места в интерфейсе пользователя.

• Изменение размера интерфейса в зависимости от размера окна

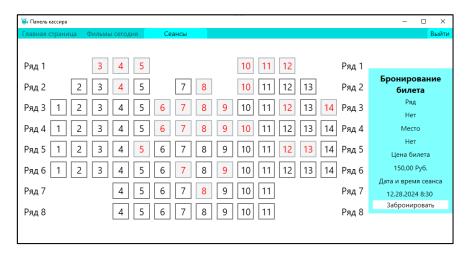


Рисунок 20 - Приложение при разрешении окна 800х1000

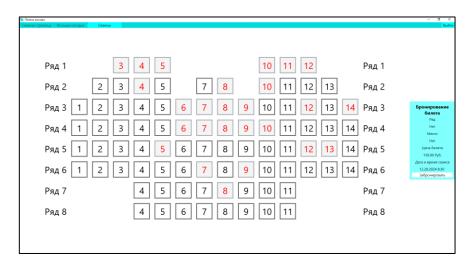


Рисунок 21 - Приложение при разрешении окна 1920х1080

Ожидаемый результат: Интерфейс автоматически изменяет размер по размер окна.

Фактический результат: Интерфейс автоматически изменяет размер по размер окна. (Рис. 20, 21)

Решение проблемы: Использование события SizeChange у окна для получения его размеров и исходя из них подгонки элементов при помощи RenderTransform.

#### **ЗАКЛЮЧЕНИЕ**

В результате выполнение курсового проекта был создан программный модуль для автоматизации продажи билетов в кинотеатре. Данный программный модуль позволяет кинотеатрам сократить времени обслуживания клиентов, увеличить пропускную способность кинотеатра, повысить удобство посетителей и снизить затраты на персонал. Также программный модуль позволяет собирать и анализировать данные коммерческой успешности фильма для планирования дальнейшей работы.

Преимущества программного модуля по сравнению с аналогами состоит в простом и интуитивно понятном интерфейсе для любого сотрудника кинотеатр, а также в возможности сбора и анализа данных для определения коммерчески успешного фильма.

В перспективах развития программного модуля можно отметить расширение функциональности таких как интеграция с системами внешних платежей (различные банковские карты, электронные кошельки, мобильные платежи), персонализация интерфейса для каждого пользователя программного модуля, использование данного модуля для реализации онлайн бронирования билета на сайте кинотеатра, встраивание программного модуля в состав десктопного приложения для большего охвата рабочих задач кинотеатра и улучшение производительности.

Таким образом, разработанное приложение представляет собой эффективное решение для оптимизации работы кинотеатра и повышения уровня сервиса для клиентов.

#### СПИСОК ЛИТЕРАТУРЫ

- 1. ГОСТ 19.101-77. Единая система программной документации. Виды программ и программных документов, введ. 01.01.1978.- г. Москва: Изд-во стандартов, 1980.-4 с.
- 2. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению, введ. 01.01.1980. М.: Изд-во стандартов, 1988. 3 с.
- 3. Алекс Дэвис Асинхронное программирование в С# 5.0. / Пер. с англ. Слинкин А. А. М.: ДМК Пресс, 2013. 120 с.: ил.
- 4. Введение в ADO.NET, 2015 [Электронный ресурс] URL: https://metanit.com/sharp/adonet/1.1.php?ysclid=m2xsepz8p6557122232 (дата обращения 15.10.2024)
- 5. Джон П. Смит Entity Framework Core в действии / пер. с англ. Д. А. Беликова. М.: ДМК Пресс, 2022. 690 с.: ил.
- 6. Обзор Entity Framework Core EF Core | Microsoft Learn, 2023 [Электронный ресурс] URL: https://learn.microsoft.com/ru-ru/ef/core/ (дата обращения: 25.09.2024)
- 7. Pабота с LINQ C# | Microsoft Learn, 2023 [Электронный ресурс] URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/tutorials/working-with-linq (дата обращения: 2.10.2024)
- 8. Трунин В. Путь программиста T-SQL. Теория и практика М:. Infocomp.ru, 2020-204 с.
- 9. ADO.NET | Microsoft Learn, 2023 [Электронный ресурс] URL: https://learn.microsoft.com/ru-ru/dotnet/framework/data/adonet/ (дата обращения 10.10.2024)
- 10. C# и .NET | LINQ, 2022 [Электронный ресурс] URL: https://metanit.com/sharp/tutorial/15.1.php?ysclid=m2xsgnx83x118354100 (дата обращения 3.10.2024)

- 11. С# и .NET | Асинхронные методы, async и await, 2022 [Электронный ресурс] URL: https://metanit.com/sharp/tutorial/13.3.php?ysclid=m2xsjn 72a4147828932 (дата обращения 16.11.2024)
- 12. Metanit C# | Запись таблиц в PDF, 2012 [Электронный ресурс] URL: https://metanit.com/sharp/articles/25.php?ysclid=m2xqyr4e12641032739 (дата обращения: 25.09.2024)
- 13. MS SQL Server в Entity Framework Core и С#, провайдер Microsoft.EntityFrameworkCore.SqlServer, метод UseSqlServer, 2021 [Электронный ресурс] URL: https://metanit.com/sharp/efcore/7.1.php?ysclid=m2xs59czu1292960590 (дата обращения 10.10.2024)
- 14. Windows Presentation Foundation WPF .NET Framework | Microsoft Learn, 2023 [Электронный ресурс] URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/?view=netframeworkdesktop-4.8 (дата обращения 27.09.2024)
- 15. WPF и C# | Полное руководство, 2023 [Электронный ресурс] URL: https://metanit.com/sharp/wpf (дата обращения: 26.09.2024)

# приложения

### Приложение 1

### КОД ПРОГРАММЫ

```
BackUpControls.xaml
                                                                               <Button x:Name="RestoreDataBase"
                                                                      Content="Восстановить" Width="180" Style="{DynamicResource
<Page x:Class="Cinema.Pages.BackUpControls"
                                                                      MainButtonStyle}" Click="RestoreDataBase_Click"/>
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                             </StackPanel>
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                          </Grid>
compatibility/2006"
                                                                        </Grid>
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                      </Page>
xmlns:local="clr-namespace:Cinema.Pages"
mc:Ignorable="d"
                                                                      BackUpControls.xaml.cs
d:DesignHeight="450" d:DesignWidth="800"
                                                                      using Microsoft.Win32;
Title="BackUpControls">
                                                                      using System;
                                                                      using System.Collections.Generic;
  <Grid Style="{DynamicResource MainColorStyle}">
                                                                      using System.Linq;
    <Grid Margin="10">
                                                                      using System.Text;
       <Grid.ColumnDefinitions>
                                                                      using System. Threading. Tasks;
         <ColumnDefinition/>
                                                                      using System. Windows;
         <ColumnDefinition/>
                                                                      using System.Windows.Controls;
       </Grid.ColumnDefinitions>
                                                                      using System.Windows.Data;
                                                                      using System.Windows.Documents;
       <StackPanel Grid.Column="0" HorizontalAlignment="Center">
                                                                      using System.Windows.Input;
         <TextBlock Margin="0,0,0,10" Text="Резервное
                                                                      using System.Windows.Media;
копирование" HorizontalAlignment="Center"
                                                                      using System.Windows.Media.Imaging;
Style="{DynamicResource TitleTextBlockStyle}"/>
                                                                      using System. Windows. Navigation;
         <WrapPanel Margin="0,0,0,10">
                                                                      using System.Windows.Shapes;
           <TextBox Margin="0,0,10,0" x:Name="BackUpPath"
Width="250" IsReadOnly="True" Style="{DynamicResource
                                                                      namespace Cinema.Pages
MainTextBoxStyle \}"/>
           <Button x:Name="OpenSavePath" Content="Обзор"
                                                                        /// <summary>
Width="120" Style="{DynamicResource MainButtonStyle}"
                                                                        /// Логика взаимодействия для BackUpControls.xaml
Click="OpenSavePath_Click"/>
                                                                        /// </summary>
         </WrapPanel>
                                                                        public partial class BackUpControls: Page
         <Button x:Name="SaveBackUp" Content="Сохранить"
Width="180" Style="{DynamicResource MainButtonStyle}"
                                                                          public BackUpControls()
Click="SaveBackUp_Click"/>
      </StackPanel>
                                                                             InitializeComponent();
       <StackPanel Grid.Column="1" HorizontalAlignment="Center">
         <TextBlock Margin="0,0,0,10" Text="Восстановление"
HorizontalAlignment="Center" Style="{DynamicResource
                                                                          public string saveBackUpPath;
TitleTextBlockStyle \} "/>
                                                                          public string restorePath;
         <WrapPanel Margin="0,0,0,10">
           <TextBox Margin="0,0,10,0" x:Name="RestorePath"
                                                                          private void OpenSavePath_Click(object sender, RoutedEventArgs
IsReadOnly="True" Width="250" Style="{DynamicResource
                                                                      e)
MainTextBoxStyle}"/>
           <Button x:Name="OpenRestorePath" Content="Обзор"
                                                                             SaveFileDialog saveFileDialog = new SaveFileDialog();
                                                                             saveFileDialog.Filter = "BackUp (*.BAK)|*.BAK";
Width="120" Style="{DynamicResource MainButtonStyle}"
Click="OpenRestorePath_Click"/>
         </WrapPanel>
                                                                            if \ (saveFileDialog.ShowDialog() == true) \\
                                                                               saveBackUpPath = saveFileDialog.FileName;
```

```
BackUpPath.Text = saveBackUpPath;
                                                                     }
      }
    }
                                                                     CurrentMovieCashierControls.xaml
                                                                     <Page x:Class="Cinema.Pages.CurrentMovieCashierControls"
                                                                     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    private void SaveBackUp_Click(object sender, RoutedEventArgs
e)
                                                                     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                     xmlns:mc="http://schemas.openxmlformats.org/markup-
      if(saveBackUpPath != null)
                                                                     compatibility/2006"
                                                                     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
         using (var dataBase = new CinemaEntities())
                                                                     xmlns:local="clr-namespace:Cinema.Pages"
                                                                     mc:Ignorable="d"
           var backUp =
                                                                     d:DesignHeight="450" d:DesignWidth="800"
dataBase.Database.SqlQuery<CinemaEntities>($"BACKUP
                                                                    Title="MovieCashierControls" SizeChanged="Page_SizeChanged"
DATABASE [Cinema] TO DISK = '{saveBackUpPath}'").ToList();
                                                                    Loaded="Page_Loaded">
           MessageBox.Show("Резервное копирование
выполненно", "Готово", MessageBoxButton.OK,
                                                                       <Grid Style="{DynamicResource MainColorStyle}">
MessageBoxImage.Information);
                                                                         <StackPanel>
                                                                            <Grid Style="{DynamicResource SecondColorStyle}">
                                                                              <WrapPanel Margin="10,0,10,0"
      }
    }
                                                                     VerticalAlignment="Center">
                                                                                <TextBlock Text="Поиск:" Margin="0,0,10,0"
    private void OpenRestorePath_Click(object sender,
                                                                     Style="{DynamicResource MainTextBlockStyle}"/>
RoutedEventArgs e)
                                                                                <TextBox x:Name="FindData" Width="200"
                                                                     Style="{DynamicResource MainTextBoxStyle}"
    {
      OpenFileDialog openFileDialog = new OpenFileDialog();
                                                                    TextChanged="FindData_TextChanged"/>
      openFileDialog.Filter = "BackUp (*.BAK)|*.BAK";
                                                                                <Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                     SecondColorSeparatorsRectangleStyle}"/>
      if (openFileDialog.ShowDialog() == true)
                                                                                <TextBlock Text="Найдено: " Style="{DynamicResource
                                                                    MainTextBlockStyle \}"/>
                                                                                <TextBlock x:Name="FindCounterData" Text="0/0"
         restorePath = openFileDialog.FileName;
         RestorePath.Text = restorePath;
                                                                    Style="{DynamicResource MainTextBlockStyle}"/>
                                                                              </WrapPanel>
      }
    }
                                                                            </Grid>
    private void RestoreDataBase_Click(object sender,
                                                                            <ListView x:Name="MovieCashierList" Height="400"</p>
RoutedEventArgs e)
                                                                     d:ItemsSource="{d:SampleData ItemCount=2}"
                                                                     ScrollViewer.CanContentScroll="False"
    {
      if (restorePath != null)
                                                                     MouseDoubleClick="MovieCashierList_MouseDoubleClick">
                                                                              <ListView.ItemContainerStyle>
         using (var dataBase = new CinemaEntities())
                                                                                <Style TargetType="ListViewItem">
                                                                                  <Setter Property="HorizontalContentAlignment"</pre>
                                                                     Value="Stretch" />
           var backUp =
dataBase.Database.SqlQuery<CinemaEntities>($"USE master; ALTER
                                                                                </Style>
DATABASE [Cinema] SET SINGLE_USER WITH ROLLBACK
                                                                             </ListView.ItemContainerStyle>
IMMEDIATE; RESTORE DATABASE [Cinema] FROM DISK =
                                                                              <ListView.View>
'{restorePath}' WITH REPLACE; ALTER DATABASE [Cinema] SET
                                                                                <GridView>
MULTI_USER;").ToList();
                                                                                  <GridViewColumn x:Name="MovieCodeGrid">
           MessageBox.Show("Востановление выполнено",
                                                                                    <Label Content="Код фильма"
"Готово", MessageBoxButton.OK, MessageBoxImage.Information);
                                                                     Style="{DynamicResource MainLableStyle}"/>
                                                                                    <GridViewColumn.CellTemplate>
      }
                                                                                       <DataTemplate>
    }
```

}

```
<Label Content="{Binding movieCashierID}"</pre>
                                                                                             < Label Content="{Binding
HorizontalAlignment="Center" Style="{DynamicResource
                                                                     movieCashierCountry}" Style="{DynamicResource
MainLableStyle \}"/>
                                                                     MainLableStyle \}"/>
                  </DataTemplate>
                                                                                           </WrapPanel>
                </GridViewColumn.CellTemplate>
                                                                                           <WrapPanel>
             </GridViewColumn>
                                                                                             <Label Content="Актеры:"
             <GridViewColumn x:Name="ImageGrid">
                                                                     Style="{DynamicResource MainLableStyle}"/>
                <Label Content="Постер" Style="{DynamicResource
                                                                                             <TextBlock Text="{Binding
MainLableStyle \}"/>
                                                                     movieCashierActors}" Margin="5,0,0,0" VerticalAlignment="Center"
                <GridViewColumn.CellTemplate>
                                                                     TextWrapping="Wrap" Style="{DynamicResource
                  <DataTemplate>
                                                                     MainTextBlockStyle \}"/>
                    <Image Source="{Binding movieCashierCover}"</pre>
                                                                                           </WrapPanel>
MaxHeight="400" HorizontalAlignment="Center"/>
                                                                                         </StackPanel>
                  </DataTemplate>
                                                                                       </DataTemplate>
                </GridViewColumn.CellTemplate>
                                                                                    </GridViewColumn.CellTemplate>
             </GridViewColumn>
                                                                                  </GridViewColumn>
             <GridViewColumn x:Name="MovieInfoGrid">
                                                                                  <GridViewColumn x:Name="DescriptionGrid">
                <Label Content="Информация о фильме"
                                                                                    <Label Content="Описание"
Style="{DynamicResource MainLableStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
                <GridViewColumn.CellTemplate>
                                                                                    <GridViewColumn.CellTemplate>
                  <DataTemplate>
                                                                                       <DataTemplate>
                    <StackPanel>
                                                                                         <TextBlock x:Name="DescriptionText"
                      <Label Content="{Binding
                                                                     Text="{Binding movieCashierDescription}" TextWrapping="Wrap"
movieCashierTitle}" Style="{DynamicResource MainLableStyle}"/>
                                                                     Style="{DynamicResource MainTextBlockStyle}"/>
                      <WrapPanel>
                                                                                       </DataTemplate>
                        <Label Content="Жанр(ы):"
                                                                                    </GridViewColumn.CellTemplate>
Style="{DynamicResource MainLableStyle}"/>
                                                                                  </GridViewColumn>
                        <Label Content="{Binding
                                                                                </GridView>
movieCashierGenre}" Style="{DynamicResource MainLableStyle}"/>
                                                                              </ListView.View>
                                                                            </ListView>
                      </WrapPanel>
                                                                         </StackPanel>
                      <WrapPanel>
                        <Label Content="Год публикации:"
                                                                       </Grid>
Style="{DynamicResource MainLableStyle}"/>
                                                                     </Page>
                        <Label Content="{Binding
movieCashierYearOfPublication}" Style="{DynamicResource
                                                                     CurrentMovieCashierControls.xaml.cs
MainLableStyle \} "/>
                                                                     using System;
                      </WrapPanel>
                                                                     using System.Collections.Generic;
                      <WrapPanel>
                                                                     using System.IO;
                        <Label Content="Хронометраж:"
                                                                     using System.Linq;
Style="{DynamicResource MainLableStyle}"/>
                                                                     using System.Text;
                        < Label Content="{Binding
                                                                     using System.Threading.Tasks;
movieCashierTiming}" Style="{DynamicResource MainLableStyle}"/>
                                                                     using System.Windows;
                      </WrapPanel>
                                                                     using System.Windows.Controls;
                      <WrapPanel>
                                                                     using System.Windows.Data;
                                                                     using System.Windows.Documents;
                        <Label Content="Возрастной рейтинг:"
Style="{DynamicResource MainLableStyle}"/>
                                                                     using System.Windows.Input;
                        <Label Content="{Binding
                                                                     using System.Windows.Media;
movieCashierAgeRating}" Style="{DynamicResource
                                                                     using System. Windows. Media. Imaging;
MainLableStyle \}"/>
                                                                     using System.Windows.Navigation;
                      </WrapPanel>
                                                                     using System.Windows.Shapes;
                      <WrapPanel>
                                                                     using static Cinema. Authorization;
                        <Label Content="Страна:"
Style="{DynamicResource MainLableStyle}"/>
                                                                     namespace Cinema.Pages
```

```
DateTime dateTimeTomorrow =
{
  /// <summary>
                                                                         DateTime.Now.AddDays(1).Date;
  /// Логика взаимодействия для CurrentMovieCashierControls.xaml
  /// </summary>
                                                                                     var movieData = (from movie in dataBase.Movie
  public partial class CurrentMovieCashierControls: Page
                                                                                               movieGenre in dataBase.MovieGenre on
                                                                         movie.IDMovie equals movieGenre.IDMovie into movieGenereGroup
    public CurrentMovieCashierControls()
                                                                                               from movieGenere in
       InitializeComponent();
                                                                         movieGenereGroup.DefaultIfEmpty()
                                                                                               join
                                                                                               genre in dataBase.Genre on
     public class MovieCashierData
                                                                         movieGenere.IDGenre equals genre.IDGenre into genereGroup
                                                                                               from genere in genereGroup.DefaultIfEmpty()
       public int movieCashierID { get; set; }
                                                                                               join
       public BitmapImage movieCashierCover { get; set; }
                                                                                              country in dataBase.Country on
       public string movieCashierTitle { get; set; }
                                                                         movie.IDCountry equals country.IDCountry into countryGroup
       public string movieCashierGenre { get; set; }
                                                                                               from country in countryGroup.DefaultIfEmpty()
       public int movieCashierYearOfPublication { get; set; }
       public double movieCashierTiming { get; set; }
                                                                                               actorsInMovies in dataBase.ActorsInMovies on
       public string movieCashierAgeRating { get; set; }
                                                                         movie.IDMovie equals actorsInMovies.IDMovie into
       public string movieCashierCountry { get; set; }
                                                                         actorsInMoviesGroup
       public string movieCashierDescription { get; set; }
                                                                                               from actorsInMovie in
       public string movieCashierActors { get; set; }
                                                                         actorsInMoviesGroup.DefaultIfEmpty()
     }
                                                                                               actors in dataBase.Actor on
     List<MovieCashierData> moviesCashierDataList = new
                                                                         actorsInMovie.IDActor equals actors.IDActor into actorsGroup
                                                                                               from actors in actorsGroup.DefaultIfEmpty()
List<MovieCashierData>();
                                                                                               session in dataBase.Session on movie.IDMovie
    private void Page_Loaded(object sender, RoutedEventArgs e)
                                                                         equals session.IDMovie into sessionGroup
     {
       try
                                                                                               from session in sessionGroup.DefaultIfEmpty()
                                                                                               where ((findLine == null ||
         MovieCashierList.ItemsSource = moviesCashierDataList;
                                                                         movie.Title.Contains(findLine) || genere.Title.Contains(findLine) ||
                                                                         movie.YearOfPublication.ToString().Contains(findLine) ||
         LoadData(null);
                                                                         movie.Description.Contains(findLine) ||
                                                                         country.Title.Contains(findLine) || actors.Surname.Contains(findLine) ||
       catch (Exception ex)
                                                                         actors.Name.Contains(findLine) || actors.Patronymic.Contains(findLine)
                                                                         || actors.Nickname.Contains(findLine)) &&
         MessageBox.Show(ex.Message, "Ошибка",
                                                                         (session.DateAndTimeSession >= dateTimeToday &&
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                         session.DateAndTimeSession <= dateTimeTomorrow))
       }
                                                                                               select new
     }
                                                                                                 movie.IDMovie,
    private void LoadData(string findLine)
                                                                                                 movie Cover
                                                                                                 movieTitle = movie.Title,
       try
                                                                                                 movieGenere = genere.Title,
                                                                                                 movie.YearOfPublication,
         using (var dataBase = new CinemaEntities())
                                                                                                 movie.Timing,
                                                                                                 movie.AgeRating,
           moviesCashierDataList.Clear();
                                                                                                 movie.Description,
                                                                                                 movieCountry = country.Title,
           DateTime dateTimeToday = DateTime.Now;
                                                                                                 movieActor = actors.Surname + " " +
                                                                         actors.Name + " " + actors.Patronymic + " " + actors.Nickname
```

```
movieDataClass.movieCashierCover = new
                     ).ToList();
                                                                        BitmapImage(new Uri("pack://application:,,,/Resource/NoImage.png",
                                                                       UriKind.Absolute));
           var\ grouped Movie Cashier Data = movie Data. Group By (m
=> m.IDMovie)
              .Select(g => new
                                                                                     movieDataClass.movieCashierTitle =
                                                                        movieLine.movieTitle;
                g.Key,
                                                                                     string movieGenereTemp = "";
                genres = g.Select(m => m.movieGenere).Distinct(),
                actors = g.Select(m => m.movieActor).Distinct()
                                                                                     int movieGenresCounterTemp = 1;
                                                                                     foreach (var genere in movieLine.movieGenres)
              }).ToList();
            var result = groupedMovieCashierData.Select(g =>
                                                                                        if (movieGenresCounterTemp !=
           {
                                                                        movieLine.movieGenres.Count())
              var movie = movieData.FirstOrDefault(m => m.IDMovie
                                                                                          movieGenereTemp += genere + ", ";
== g.Key);
              return new
                                                                                          movieGenereTemp += genere;
                movie.IDMovie,
                                                                                        movieGenresCounterTemp++;
                movie.Cover,
                movie.movieTitle,
                                                                                     movieDataClass.movieCashierGenre =
                movieGenres = g.genres,
                                                                       movieGenereTemp;
                movie.YearOfPublication,
                movie.Timing,
                                                                                     movieDataClass.movieCashierYearOfPublication =
                movie.AgeRating,
                                                                        movieLine.YearOfPublication;
                movie.Description,
                                                                                     movie Data Class. movie Cashier Timing = \\
                movie.movieCountry,
                                                                        movieLine.Timing;
                movieActors = g.actors
                                                                                     movie Data Class. movie Cashier Age Rating = \\
                                                                       movieLine.AgeRating + "+";
              };
            }).ToList();
                                                                                     movieDataClass.movieCashierDescription =
                                                                        movieLine.Description;
           foreach (var movieLine in result)
                                                                                     movieDataClass.movieCashierCountry =
                                                                        movieLine.movieCountry;
              MovieCashierData movieDataClass = new
MovieCashierData();
                                                                                     string movieActorTemp = "";
                                                                                     int movieActorCounterTemp = 1;
              movieDataClass.movieCashierID = movieLine.IDMovie;
                                                                                     foreach (var actor in movieLine.movieActors)
              if (movieLine.Cover != null)
                                                                                        if (movieActorCounterTemp !=
              {
                                                                       movieLine.movieActors.Count())
                BitmapImage cover = new BitmapImage();
                                                                                          movieActorTemp += actor + ", ";
                cover.BeginInit();
                cover.StreamSource = new
                                                                                        else
MemoryStream(movieLine.Cover);
                                                                                        {
                cover.EndInit();
                                                                                          movieActorTemp += actor;
                movieDataClass.movieCashierCover = cover;
                                                                                        movieActorCounterTemp++;
              else
                                                                                     movieDataClass.movieCashierActors =
                                                                       movieActorTemp;
```

```
moviesCashierDataList.Add(movieDataClass);
                                                                           private void FindData_TextChanged(object sender,
                                                                      TextChangedEventArgs e)
           var fullMovieCashierData = (from movie in
                                                                             LoadData(FindData.Text);
dataBase.Movie
                           join
                                                                           private void MovieCashierList_MouseDoubleClick(object sender,
                           session in dataBase.Session on
movie.IDMovie equals session.IDMovie into sessionGroup
                                                                      MouseButtonEventArgs e)
                           from session in
                                                                           {
sessionGroup.DefaultIfEmpty()
                                                                             try
                           where (session.DateAndTimeSession >=
dateTimeToday && session.DateAndTimeSession <=
                                                                                var selectedCashierMovie = MovieCashierList.SelectedItem
dateTimeTomorrow)
                                                                       as MovieCashierData;
                           group movie by movie. Title into
                                                                                if (selectedCashierMovie != null)
groupedMovies
                                                                                  TransmittedData.idSelectedCashierMovie =
                           select new
                                                                       selectedCashierMovie.movieCashierID;
                              MovieTitle = groupedMovies.Key,
                             TotalSessions =
                                                                                  CashierPanel cashierPanel = Window.GetWindow(this) as
groupedMovies.Count()
                                                                       CashierPanel;
                           }).ToList();
                                                                                  cashierPanel.SessionPageOpen();
           FindCounterData.Text = result.Count() + "/" +
fullMovieCashierData.Count();
                                                                             catch (Exception ex)
           MovieCashierList.Items.Refresh();
                                                                                MessageBox.Show(ex.Message, "Ошибка",
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
         }
       }
       catch (Exception ex)
                                                                           }
                                                                         }
         MessageBox.Show(ex.Message, "Ошибка",
                                                                       }
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                       EmployeeControls.xaml
       }
                                                                       <Page x:Class="Cinema.EmployeeControls"
     }
                                                                       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
     private void Page_SizeChanged(object sender,
                                                                       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
SizeChangedEventArgs e)
                                                                       xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                       compatibility/2006"
       try
                                                                       xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                          xmlns:local="clr-namespace:Cinema"
                                                                          mc:Ignorable="d"
         MovieCodeGrid.Width = 0;
         ImageGrid.Width = ActualWidth - ActualWidth * 0.8;
                                                                          d:DesignHeight="450" d:DesignWidth="800"
         MovieInfoGrid.Width = ActualWidth - ActualWidth * 0.7;
                                                                          Title="EmployeeControls" Loaded="Page_Loaded"
         DescriptionGrid.Width = ActualWidth - ActualWidth * 0.5;
                                                                      SizeChanged="Page_SizeChanged">
         MovieCashierList.Height = ActualHeight - 35;
                                                                         <Grid Style="{DynamicResource MainColorStyle}">
                                                                           <StackPanel>
       catch (Exception ex)
                                                                              <Grid Style="{DynamicResource SecondColorStyle}">
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                <WrapPanel Margin="10,0,0,0,0" VerticalAlignment="Center">
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                  <Button x:Name="AddEmployee" Margin="0,0,5,0"
                                                                       Content="Добавить" HorizontalAlignment="Left" Width="120"
                                                                       Style="{DynamicResource TabControlButtonStyle}"
     }
```

Click="AddEmployee\_Click"/>

```
<Button x:Name="RemoveEmployee" Content="Удалить"
                                                                                    <Label Content="Информация о сотруднике"
HorizontalAlignment="Left" Width="120" Style="{DynamicResource
                                                                    Style="{DynamicResource MainLableStyle}"/>
TabControlButtonStyle}" Click="RemoveEmployee_Click"/>
                                                                                   <GridViewColumn.CellTemplate>
           <Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                                      <DataTemplate>
                                                                                        <StackPanel>
SecondColorSeparatorsRectangleStyle}"/>
           <TextBlock Text="Поиск:" Margin="0,0,10,0"
                                                                                          <WrapPanel>
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                                            < Label Content="{Binding
           <TextBox x:Name="FindData" Margin="0,0,10,0"
                                                                    employeeSurname}" Style="{DynamicResource MainLableStyle}"/>
Width="200" Style="{DynamicResource MainTextBoxStyle}"
                                                                                            <Label Content="{Binding
                                                                    employeeName}" Style="{DynamicResource MainLableStyle}"/>
TextChanged="FindData_TextChanged"/>
           <Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                                            < Label Content="{Binding
SecondColorSeparatorsRectangleStyle}"/>
                                                                    employeePatronymic}" Style="{DynamicResource MainLableStyle}"/>
           <TextBlock Text="Найдено: " Style="{DynamicResource
                                                                                          </WrapPanel>
MainTextBlockStyle \}"/>
                                                                                          <WrapPanel>
           <TextBlock x:Name="FindCounterData" Text="0/0"
                                                                                            <Label Content="Телефон:"
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                    Style="{DynamicResource MainLableStyle}"/>
         </WrapPanel>
                                                                                            < Label Content="{Binding
                                                                    employeePhone}" Style="{DynamicResource MainLableStyle}"/>
       </Grid>
       <ListView x:Name="EmployeeList" Height="400"</p>
                                                                                          </WrapPanel>
d:ItemsSource="{d:SampleData ItemCount=2}"
                                                                                          <WrapPanel>
ScrollViewer.CanContentScroll="False"
                                                                                            <Label Content="Почта:"
MouseDoubleClick="EmployeeList_MouseDoubleClick">
                                                                    Style="{DynamicResource MainLableStyle}"/>
         <ListView.ItemContainerStyle>
                                                                                            < Label Content="{Binding
           <Style TargetType="ListViewItem">
                                                                    employeeEmail}" Style="{DynamicResource MainLableStyle}"/>
             <Setter Property="HorizontalContentAlignment"
                                                                                          </WrapPanel>
Value="Stretch" />
                                                                                          <WrapPanel>
                                                                                            <Label Content="Должность:"
           </Style>
         </ListView.ItemContainerStyle>
                                                                    Style="{DynamicResource MainLableStyle}"/>
         <ListView.View>
                                                                                            <Label Content="{Binding employeeRole}"</pre>
           <GridView>
                                                                    Style="{DynamicResource MainLableStyle}"/>
                                                                                          </WrapPanel>
             <GridViewColumn x:Name="EmployeeCodeGrid">
               <Label Content="Код сотрудника"
                                                                                        </StackPanel>
Style="{DynamicResource MainLableStyle}"/>
                                                                                      </DataTemplate>
               <GridViewColumn.CellTemplate>
                                                                                    </GridViewColumn.CellTemplate>
                                                                                 </GridViewColumn>
                  <DataTemplate>
                    <Label Content="{Binding employeeID}"</pre>
                                                                                  <GridViewColumn x:Name="AuthorizationGrid">
HorizontalAlignment="Center" Style="{DynamicResource
                                                                                    <Label Content="Данные авторизации"
MainLableStyle \}"/>
                                                                    Style="{DynamicResource MainLableStyle}"/>
                  </DataTemplate>
                                                                                   <GridViewColumn.CellTemplate>
               </GridViewColumn.CellTemplate>
                                                                                      <DataTemplate>
             </GridViewColumn>
                                                                                        <StackPanel>
             <GridViewColumn x:Name="ImageGrid">
                                                                                          <WrapPanel>
               <Label Content="Фотография"
                                                                                            <Label Content="Логин:"
Style="{DynamicResource MainLableStyle}"/>
                                                                    Style="{DynamicResource MainLableStyle}"/>
               <GridViewColumn.CellTemplate>
                                                                                            <Label Content="{Binding
                  <DataTemplate>
                                                                    employeeLogin}" Style="{DynamicResource MainLableStyle}"/>
                    <Image Source="{Binding employeePhoto}"</pre>
                                                                                          </WrapPanel>
MaxHeight="400" HorizontalAlignment="Center"/>
                                                                                          <WrapPanel>
                  </DataTemplate>
                                                                                            <Label Content="Пароль:"
               </GridViewColumn.CellTemplate>
                                                                    Style="{DynamicResource MainLableStyle}"/>
             </GridViewColumn>
                                                                                            < Label Content="{Binding
             <GridViewColumn x:Name="EmployeeInfoGrid">
                                                                    employeePassword}" Style="{DynamicResource MainLableStyle}"/>
```

</WrapPanel>

```
</StackPanel>
                                                                                try
                   </DataTemplate>
                 </GridViewColumn.CellTemplate>
                                                                                  EmployeeList.ItemsSource = employeeDataList;
              </GridViewColumn>
            </GridView>
                                                                                  LoadData(null);
         </ListView.View>
                                                                                }
       </ListView>
                                                                                catch (Exception ex)
     </StackPanel>
  </Grid>
                                                                                  MessageBox.Show(ex.Message, "Ошибка",
                                                                         MessageBoxButton.OK, MessageBoxImage.Error);
</Page>
EmployeeControls.xaml.cs
                                                                              }
using System;
using System.Collections.Generic;
                                                                              private void LoadData(string findLIne)
using System.IO;
                                                                              {
using System.Linq;
                                                                                try
using System. Windows;
using System.Windows.Controls;
                                                                                   using (var dataBase = new CinemaEntities())
using System.Windows.Input;
using System.Windows.Media.Imaging;
                                                                                     employeeDataList.Clear();
using static Cinema. Authorization;
                                                                                     var employeeData = (from employee in dataBase.Employee
namespace Cinema
                                                                                                 role in dataBase.Role on employee.IDRole
{
  /// <summary>
                                                                         equals role.IDRole into roleGroup
  /// Логика взаимодействия для EmployeeControls.xaml
                                                                                                 from role in roleGroup.DefaultIfEmpty()
                                                                                                 where (findLIne == null ||
  /// </summary>
  public partial class EmployeeControls : Page
                                                                         employee.Surname.Contains(findLIne) ||
                                                                         employee.Name.Contains(findLIne) ||
    public EmployeeControls()
                                                                         employee. Patronymic. Contains (find LIne) \parallel
                                                                         employee.Phone.Contains(findLIne) \parallel
       InitializeComponent();
                                                                         employee.Email.Contains(findLIne) ||
                                                                         employee.Login.Contains(findLIne) ||
     }
                                                                         employee.Password.Contains(findLIne))
    public class EmployeeData
                                                                                                 select new
       public int employeeID { get; set; }
                                                                                                   employee.IDEmployee,
       public BitmapImage employeePhoto { get; set; }
                                                                                                   employee.Photo,
       public string employeeSurname { get; set; }
                                                                                                   employee.Surname,
       public string employeeName { get; set; }
                                                                                                   employee.Name,
       public string employeePatronymic { get; set; }
                                                                                                   employee.Patronymic,
       public string employeePhone { get; set; }
                                                                                                   employee.Phone,
       public string employeeEmail { get; set; }
                                                                                                   employee.Email,
       public string employeeLogin { get; set; }
                                                                                                   employee.Login,
       public string employeePassword { get; set; }
                                                                                                   employee.Password,
       public string employeeRole { get; set; }
                                                                                                   employeeRole = role.Title,
                                                                                                 }).ToList();
     List<EmployeeData> employeeDataList = new
                                                                                     foreach (var employeeLine in employeeData)
List<EmployeeData>();
                                                                                       EmployeeData\ employeeData\ Class = new
     private void Page_Loaded(object sender, RoutedEventArgs e)
                                                                         EmployeeData();
```

```
employeeDataClass.employeeID =
employeeLine.IDEmployee;
                                                                          private void Page_SizeChanged(object sender,
                                                                     SizeChangedEventArgs e)
             if (employeeLine.Photo != null)
                                                                          {
                                                                            try
               BitmapImage bitmapImage = new BitmapImage();
               bitmapImage.BeginInit();
                                                                              EmployeeCodeGrid.Width = 0;
               bitmapImage.StreamSource = new
                                                                              ImageGrid.Width = ActualWidth - ActualWidth * 0.8;
                                                                              EmployeeInfoGrid.Width = ActualWidth - ActualWidth * 0.6;
MemoryStream(employeeLine.Photo);
               bitmapImage.EndInit();
                                                                              AuthorizationGrid.Width = ActualWidth - ActualWidth * 0.6;
                                                                              EmployeeList.Height = ActualHeight - 35;
               employeeDataClass.employeePhoto = bitmapImage;
                                                                            catch (Exception ex)
             }
             else
                                                                              MessageBox.Show(ex.Message, "Ошибка",
             {
                                                                     MessageBoxButton.OK, MessageBoxImage.Error);
               employeeDataClass.employeePhoto = new
BitmapImage(new Uri("pack://application:,,,/Resource/NoImage.png",
                                                                            }
UriKind.Absolute));
                                                                          }
             }
                                                                         private void FindData_TextChanged(object sender,
             employeeDataClass.employeeSurname =
                                                                     TextChangedEventArgs e)
employeeLine.Surname;
                                                                          {
             employee Data Class. employee Name = \\
                                                                            LoadData(FindData.Text);
employeeLine.Name;
                                                                          }
             employeeDataClass.employeePatronymic =
                                                                          private void EmployeeList_MouseDoubleClick(object sender,
employeeLine.Patronymic;
             employee Data Class. employee Phone = \\
                                                                     MouseButtonEventArgs e)
employeeLine.Phone;
                                                                          {
             employee Data Class. employee Email = \\
                                                                            try
employeeLine.Email;
             employeeDataClass.employeeLogin = \\
                                                                              var selectedEmployee = EmployeeList.SelectedItem as
                                                                     EmployeeData;
employeeLine.Login;
             employee Data Class. employee Password = \\
                                                                              if (selectedEmployee != null)
employeeLine.Password;
             employeeDataClass.employeeRole = \\
                                                                                Transmitted Data. id Selected Employee = \\
employeeLine.employeeRole;
                                                                     selectedEmployee.employeeID;
             employeeDataList.Add(employeeDataClass);
                                                                                AddOrEditEmployee addOrEditEmployee = new
                                                                     AddOrEditEmployee();
                                                                                addOrEditEmployee.ShowDialog();
           var fullEmployeeData = dataBase.Employee.ToList();
           LoadData(FindData.Text);
fullEmployeeData.Count();
           EmployeeList.Items.Refresh();
                                                                            catch (Exception ex)
         }
                                                                              MessageBox.Show(ex.Message, "Ошибка",
      catch (Exception ex)
                                                                     MessageBoxButton.OK, MessageBoxImage.Error);
         MessageBox.Show(ex.Message, "Ошибка",
                                                                          }
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                          private void AddEmployee_Click(object sender, RoutedEventArgs
      }
    }
                                                                     e)
```

```
var removeTickets = dataBase.Ticket.Where(w =>
     {
                                                                     w.IDEmployee == selectedEmployee.employeeID).ToList();
       try
        TransmittedData.idSelectedEmployee = -1;
                                                                                    if (removeTickets.Count > 0)
         AddOrEditEmployee addOrEditEmployee = new
                                                                                       if (MessageBox.Show("Удаляемый сотрудник
AddOrEditEmployee();
                                                                     имеет проданные билеты. Вы действительно хотите его удалить?",
         addOrEditEmployee.ShowDialog();
                                                                     "Внимание", MessageBoxButton. YesNo, MessageBoxImage. Warning)
                                                                     == MessageBoxResult.No)
         LoadData(FindData.Text);
                                                                                         return;
       catch (Exception ex)
                                                                                    dataBase.Ticket.RemoveRange(removeTickets);
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                    data Base. Employee. Remove (remove Employee);\\
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                    dataBase.SaveChanges();
                                                                                    MessageBox.Show("Данные были удаленны",
     }
                                                                     "Готово", MessageBoxButton.OK, MessageBoxImage.Information);
    private void RemoveEmployee_Click(object sender,
                                                                                  }
RoutedEventArgs e)
                                                                                  LoadData(FindData.Text);
      try
         var selectedEmployee = EmployeeList.SelectedItem as
EmployeeData;
                                                                            catch (Exception ex)
         if (selectedEmployee.employeeID ==
                                                                              MessageBox.Show(ex.Message, "Ошибка",
TransmittedData.idEmployee)
                                                                     MessageBoxButton.OK, MessageBoxImage.Error);
           MessageBox.Show("Вы не можете удалить свою же
                                                                         }
учетную запись", "Внимание", MessageBoxButton.OK,
MessageBoxImage.Warning);
                                                                     }
           return:
                                                                     MovieAnalysis.xaml
                                                                     <Page x:Class="Cinema.MovieAnalysis"
         if (selectedEmployee.employeeRole == "Администратор")
                                                                     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
           MessageBox.Show("Вы не можете удалить учетную
                                                                     xmlns:mc="http://schemas.openxmlformats.org/markup-
запись администратора", "Внимание", MessageBoxButton.OK,
                                                                     compatibility/2006"
MessageBoxImage.Warning);
                                                                     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                        xmlns:local="clr-namespace:Cinema"
           return;
                                                                        mc:Ignorable="d"
         }
                                                                        d:DesignHeight="450" d:DesignWidth="800"
         if (selectedEmployee != null)
                                                                        Title="TicketAnalysis" SizeChanged="Page_SizeChanged"
                                                                     Loaded="Page_Loaded">
         {
           if (MessageBox.Show("Вы действительно хотите удалить
данного сотрудника?", "Внимание", MessageBoxButton. YesNo,
                                                                       <Grid Style="{DynamicResource MainColorStyle}">
MessageBoxImage.Warning) == MessageBoxResult.Yes)
                                                                         <StackPanel>
                                                                            <Grid Style="{DynamicResource SecondColorStyle}">
             using (var dataBase = new CinemaEntities())
                                                                              <WrapPanel Margin="10,0,0,0,0" VerticalAlignment="Center">
                                                                                <Button x:Name="GoBackPage" Content="Назад"
                var removeEmployee = dataBase.Employee.Where(w
                                                                     Width="80" Style="{DynamicResource TabControlButtonStyle}"
=> w.IDEmployee == selectedEmployee.employeeID).FirstOrDefault();
                                                                     Click="GoBackPage_Click"/>
```

```
<Rectangle Margin="10,0,10,0" Fill="Black"
                                                                                     <Label Content="Информация"
Style="{DynamicResource SecondColorSeparatorsRectangleStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
           <CheckBox x:Name="FindDateChecker" Margin="0,4,5,0"</p>
                                                                                     <GridViewColumn.CellTemplate>
Style="{DynamicResource MainCheckBoxStyle}"
                                                                                       <DataTemplate>
Click="FindDateChecker_Click"/>
                                                                                         <StackPanel>
           <TextBlock Text="C:" Margin="0,0,10,0"
                                                                                            <WrapPanel>
                                                                                              <Label Content="Дата сеанса:"
Style="{DynamicResource MainTextBlockStyle}"/>
           <DatePicker x:Name="BeginDate" Margin="0,0,10,0"</pre>
                                                                     Style="{DynamicResource MainLableStyle}"/>
Style="{DynamicResource SecondDatePickerStyle}"
                                                                                              <Label Content="{Binding dateSession}"
SelectedDateChanged="FindDate_SelectedDateChanged"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
           <TextBlock Text="по" Margin="0,0,10,0"
                                                                                           </WrapPanel>
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                                            <WrapPanel>
           <DatePicker x:Name="EndDate"
                                                                                              <Label Content="Время сеанса:"
Style="{DynamicResource SecondDatePickerStyle}"
                                                                     Style="{DynamicResource MainLableStyle}"/>
SelectedDateChanged="FindDate_SelectedDateChanged"/>
                                                                                              <Label Content="{Binding timeSession}"</pre>
           <Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                     Style="{DynamicResource MainLableStyle}"/>
SecondColorSeparatorsRectangleStyle}"/>
                                                                                            </WrapPanel>
           <TextBlock Text="Найдено: " Style="{DynamicResource
                                                                                         </StackPanel>
MainTextBlockStyle}"/>
                                                                                       </DataTemplate>
           <TextBlock x:Name="FindCounterData" Text="0/0"
                                                                                     </GridViewColumn.CellTemplate>
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                                   </GridViewColumn>
         </WrapPanel>
                                                                                   <GridViewColumn x:Name="AnalyticsGrid">
         <WrapPanel Margin="0,0,10,0"
                                                                                     <Label Content="Аналитика продаж"
HorizontalAlignment="Right" VerticalAlignment="Center">
                                                                     Style="{DynamicResource MainLableStyle}"/>
           <Button x:Name="Print" Content="Печать отчета"
                                                                                     <GridViewColumn.CellTemplate>
Width="120" Style="{DynamicResource TabControlButtonStyle}"
                                                                                       <DataTemplate>
Click="Print_Click"/>
                                                                                         <StackPanel>
       </WrapPanel>
                                                                                            <WrapPanel>
       </Grid>
                                                                                              <Label Content="Количество проданных
                                                                     билетов: "Style="{DynamicResource MainLableStyle}"/>
       <ListView x:Name="SessionList" Height="400"</pre>
                                                                                              < Label Content="{Binding
d:ItemsSource="{d:SampleData ItemCount=2}"
                                                                     ticketCountSession}" Style="{DynamicResource MainLableStyle}"/>
ScrollViewer.CanContentScroll="False">
                                                                                           </WrapPanel>
         <ListView.ItemContainerStyle>
                                                                                            <WrapPanel>
           <Style TargetType="ListViewItem">
                                                                                              <Label Content="Цена билета:"
              <Setter Property="HorizontalContentAlignment"
                                                                     Style="{DynamicResource MainLableStyle}"/>
Value="Stretch" />
                                                                                              <Label Content="{Binding ticketPrice}"</pre>
           </Style>
                                                                     Style="{DynamicResource MainLableStyle}"/>
                                                                                           </WrapPanel>
         </ListView.ItemContainerStyle>
         <ListView.View>
                                                                                            <WrapPanel>
           <GridView>
                                                                                              <Label Content="Суммарные сборы:"
              <GridViewColumn x:Name="SessionCodeGrid">
                                                                     Style="{DynamicResource MainLableStyle}"/>
                <Label Content="Код сессии"
                                                                                              < Label Content="{Binding
                                                                     ticketSummarySession}" Style="{DynamicResource
Style="{DynamicResource MainLableStyle}"/>
                <GridViewColumn.CellTemplate>
                                                                     MainLableStyle \}"/>
                  <DataTemplate>
                                                                                            </WrapPanel>
                    <Label Content="{Binding idSession}"</pre>
                                                                                         </StackPanel>
HorizontalAlignment="Center" Style="{DynamicResource
                                                                                       </DataTemplate>
MainLableStyle \} "/>
                                                                                     </GridViewColumn.CellTemplate>
                  </DataTemplate>
                                                                                   </GridViewColumn>
                </GridViewColumn.CellTemplate>
                                                                                </GridView>
              </GridViewColumn>
                                                                              </ListView View>
              <GridViewColumn x:Name="SessionInformationGrid">
                                                                            </ListView>
```

```
</StackPanel>
                                                                                    MessageBox.Show(ex.Message, "Ошибка",
  </Grid>
                                                                          MessageBoxButton.OK, MessageBoxImage.Error);
</Page>
                                                                               }
MovieAnalysis.xaml.cs
using iTextSharp.text;
                                                                               private void LoadData(DatePicker beginDate, DatePicker endDate)
using iTextSharp.text.pdf;
using Microsoft.Win32;
                                                                                 try
using System;
                                                                                 {
using System.Collections.Generic;
                                                                                    using (var dataBase = new CinemaEntities())
using System.Diagnostics;
using System.IO;
                                                                                      sessionDataAnalysesList.Clear();
using System.Linq;
using System. Windows;
                                                                                      var sessionData = (from session in dataBase.Session
using System.Windows.Controls;
                                                                                                  join
                                                                                                  ticket in dataBase.Ticket on session.IDSession
using static Cinema. Authorization;
                                                                          equals ticket.IDSession into ticketGroup
namespace Cinema
                                                                                                  from ticket in ticketGroup.DefaultIfEmpty()
                                                                                                  where ((session.IDMovie ==
{
  /// <summary>
                                                                          TransmittedData.idSelectedMovieAnalysis) &&
  /// Логика взаимодействия для MovieAnalysis.xaml
                                                                          ((beginDate.SelectedDate == null || session.DateAndTimeSession >=
  /// </summary>
                                                                          beginDate.SelectedDate.Value) && (endDate.SelectedDate == null ||
  public partial class MovieAnalysis: Page
                                                                          session.DateAndTimeSession <= endDate.SelectedDate.Value)))
                                                                                                  select new
     public MovieAnalysis()
                                                                                                    session.IDSession,
       InitializeComponent();
                                                                                                    session.DateAndTimeSession,
                                                                                                    TicketID = ticket == null ? 0:
                                                                          ticket.IDTicket,
    public class SessionDataAnalysis
                                                                                                    session.TicketPrice,
                                                                                                  }).ToList();
       public int idSession { get; set; }
       public string dateSession { get; set; }
                                                                                      var groupedSessionData = sessionData.GroupBy(m =>
       public string timeSession { get; set; }
                                                                          m.IDSession)
       public string ticketCountSession { get; set; }
                                                                                         .Select(g => new
       public string ticketPrice { get; set; }
       public string ticketSummarySession { get; set; }
                                                                                           g.Key,
                                                                                           TicketCount = g.Count(),
                                                                                           TotalCost = g.Sum(t => t.TicketPrice)
     List<SessionDataAnalysis> sessionDataAnalysesList = new
                                                                                         }).ToList();
List<SessionDataAnalysis>();
                                                                                      var result = groupedSessionData.Select(g =>
    private void Page_Loaded(object sender, RoutedEventArgs e)
                                                                                         var session = sessionData.FirstOrDefault(m =>
       try
                                                                          m.IDSession == g.Key);
                                                                                         return new
          SessionList.ItemsSource = sessionDataAnalysesList;
                                                                                           session.IDSession,
          LoadData(BeginDate, EndDate);
                                                                                           session.DateAndTimeSession,
                                                                                           session.TicketPrice,
       catch (Exception ex)
                                                                                           g.TicketCount,
                                                                                           g.TotalCost
                                                                                         };
```

```
}).ToList();
                     foreach (var sessionLine in result)
                                                                                                                                           private void Page_SizeChanged(object sender,
                         SessionDataAnalysis sessionDataAnalysisClass = new
                                                                                                                                  SizeChangedEventArgs e)
SessionDataAnalysis();
                                                                                                                                           {
                                                                                                                                               try
                          sessionDataAnalysisClass.idSession =
sessionLine.IDSession;
                                                                                                                                                    SessionCodeGrid.Width = 0;
                                                                                                                                                    SessionInformationGrid.Width = ActualWidth - ActualWidth
                         sessionDataAnalysisClass.dateSession =
sessionLine.DateAndTimeSession.ToString().Split(' ')[0];
                                                                                                                                   * 0.5;
                         session Data Analysis Class.time Session = \\
                                                                                                                                                    AnalyticsGrid.Width = ActualWidth - ActualWidth * 0.5;
sessionLine.DateAndTimeSession.ToString().Split(' ')[1];
                                                                                                                                                    SessionList.Height = ActualHeight - 35;
                          sessionDataAnalysisClass.ticketPrice =
session Line. Ticket Price. To String (). Remove (session Line. Ticket Price. To Albert Price) and the price of the pric
                                                                                                                                               catch (Exception ex)
String().Length - 2, 2) + " py6.";;
                                                                                                                                                   MessageBox.Show(ex.Message, "Ошибка",
                                                                                                                                   MessageBoxButton.OK, MessageBoxImage.Error);
                         if (sessionLine.TicketCount == 1)
                         {
                              var ticketCount = dataBase.Ticket.Where(w =>
                                                                                                                                           }
w.IDSession == sessionLine.IDSession).Count();
                              sessionDataAnalysisClass.ticketCountSession =
                                                                                                                                          private void GoBackPage_Click(object sender, RoutedEventArgs
ticketCount.ToString();
                                                                                                                                  e)
                              sessionDataAnalysisClass.ticketSummarySession =
                                                                                                                                           {
(sessionLine.TicketPrice *
                                                                                                                                               try
ticketCount).ToString().Remove((sessionLine.TicketPrice *
ticketCount).ToString().Length - 2, 2) + " py6.";
                                                                                                                                                    Window activeWindow = Window.GetWindow(this);
                          }
                                                                                                                                                   if (activeWindow.Title == "Панель директора")
                         else
                                                                                                                                                       DirectorsPanel directorsPanel = Window.GetWindow(this)
                              session Data Analysis Class.ticket Count Session = \\
sessionLine.TicketCount.ToString();
                                                                                                                                   as DirectorsPanel:
                              sessionDataAnalysisClass.ticketSummarySession =
                                                                                                                                                       directorsPanel.TicketAnalysisOpen();
sessionLine.TotalCost.ToString().Remove(sessionLine.TotalCost.ToStri
ng().Length - 2, 2) + " pyδ.";
                         }
                                                                                                                                                    if (activeWindow.Title = "Панель менеджера")
                                                                                                                                                       BookerPanel = Window.GetWindow(this) as
sessionDataAnalysesList.Add(sessionDataAnalysisClass);
                                                                                                                                   BookerPanel;
                                                                                                                                                       bookerPanel.TicketAnalysisOpen();
                     var fullSessionData = dataBase.Session.Where(w =>
                                                                                                                                               }
w.IDMovie == TransmittedData.idSelectedMovieAnalysis).ToList();
                                                                                                                                               catch (Exception ex)
                    FindCounterData.Text = result.Count() + "/" +
fullSessionData.Count();
                                                                                                                                                    MessageBox.Show(ex.Message, "Ошибка",
                                                                                                                                  MessageBoxButton.OK,\,MessageBoxImage.Error);
                     SessionList.Items.Refresh();
                                                                                                                                               }
                 }
                                                                                                                                           }
             catch (Exception ex)
                                                                                                                                           private void FindDateChecker_Click(object sender,
                                                                                                                                  RoutedEventArgs e)
                 MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                                                                               try
```

```
movie.Title,
         if (FindDateChecker.IsChecked == false)
                                                                                                 movie. Year Of Publication,
                                                                                                 session.DateAndTimeSession,
           BeginDate.IsEnabled = false;
                                                                                               }).FirstOrDefault();
           BeginDate.SelectedDate = null;
           EndDate.IsEnabled = false;
                                                                                   if (saveFileDialog.ShowDialog() == true)
           EndDate.SelectedDate = null;
                                                                                      Document doc = new Document();
         else
                                                                                      PdfWriter writer = PdfWriter.GetInstance(doc, new
                                                                        FileStream($@"{saveFileDialog.FileName}", FileMode.Create));
         {
                                                                                      doc.SetPageSize(PageSize.A4.Rotate());
           BeginDate.IsEnabled = true;
           BeginDate.SelectedDate = DateTime.Now;
           EndDate.IsEnabled = true;
                                                                                      BaseFont baseFont =
           EndDate.SelectedDate = DateTime.Now;
                                                                        BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf",
                                                                        BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
                                                                                      Font font = new Font(baseFont, 12);
         LoadData(BeginDate, EndDate);
                                                                                      BaseFont baseFontHead =
                                                                        BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf",
       catch (Exception ex)
                                                                        BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                      Font fontHead = new Font(baseFont, 20, Font.BOLD);
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                      doc.Open();
       }
     }
                                                                                      Paragraph mainParagraph = new Paragraph($"Отчет
                                                                        продаж по фильму {sessionData.Title + " " +
    private void FindDate_SelectedDateChanged(object sender,
SelectionChangedEventArgs e)
                                                                        sessionData.YearOfPublication}", fontHead);
                                                                                      mainParagraph.Alignment = Element.ALIGN_CENTER;
       LoadData(BeginDate, EndDate);
                                                                                      doc.Add(mainParagraph);
     }
                                                                                      PdfPTable table = new PdfPTable(4);
    private void Print_Click(object sender, RoutedEventArgs e)
                                                                                      table.SpacingBefore = 10;
     {
                                                                                      table.WidthPercentage = 100;
      try
                                                                                      table.AddCell(new Paragraph("Код сеанса", font));
         using (var dataBase = new CinemaEntities())
                                                                                      table.AddCell(new Paragraph("Дата и время
                                                                        проведения сеанса", font));
           SaveFileDialog saveFileDialog = new SaveFileDialog();
                                                                                      table.AddCell(new Paragraph("Количество проданных
           saveFileDialog.Filter = "PDF files (*.pdf)|*.pdf";
                                                                        билетов", font));
                                                                                      table.AddCell(new Paragraph("Выручка с проданных
                                                                        билетов", font));
           var sessionData = (from session in dataBase.Session
                      ioin
                      movie in dataBase.Movie on session.IDMovie
                                                                                      foreach (var line in sessionDataAnalysesList)
equals movie.IDMovie into movieGroup
                      from movie in movieGroup.DefaultIfEmpty()
                                                                                        table.AddCell(new
                      where ((session.IDMovie ==
                                                                        Paragraph(line.idSession.ToString(), font));
TransmittedData.idSelectedMovieAnalysis) &&
                                                                                        table.AddCell(new Paragraph(line.dateSession + " " +
((BeginDate.SelectedDate == null || session.DateAndTimeSession >=
                                                                        line.timeSession, font));
BeginDate.SelectedDate.Value) && (EndDate.SelectedDate == null ||
                                                                                        table.AddCell(new Paragraph(line.ticketCountSession,
session.DateAndTimeSession <= EndDate.SelectedDate.Value)))
                                                                        font));
                      select new
                                                                                        table.AddCell(new
                                                                        Paragraph(line.ticketSummarySession.ToString(), font));
                         session.IDSession,
```

```
<TextBox x:Name="FindData" Width="200"
             doc.Add(table);
                                                                     Style="{DynamicResource MainTextBoxStyle}"
                                                                     TextChanged="FindData_TextChanged"/>
             double ticketPriceSumm = 0;
                                                                                 <Rectangle Margin="10,0,10,0" Style="{DynamicResource
             foreach (var line in sessionDataAnalysesList)
                                                                     SecondColorSeparatorsRectangleStyle}"/>
                                                                                 <TextBlock Text="Найдено: " Style="{DynamicResource
               ticketPriceSumm +=
                                                                     MainTextBlockStyle \} "/>
Convert.ToDouble(line.ticketSummarySession.Replace("py6.", ""));
                                                                                <TextBlock x:Name="FindCounterData" Text="0/0"
                                                                     Style="{DynamicResource MainTextBlockStyle}"/>
                                                                              </WrapPanel>
             Paragraph paragraph = new Paragraph("Итог: " +
                                                                            </Grid>
ticketPriceSumm.ToString() + " Py6.", font);
             doc.Add(paragraph);
                                                                            <ListView x:Name="MovieCashierList" Height="400"</p>
                                                                     d:ItemsSource="{d:SampleData ItemCount=2}"
                                                                     ScrollViewer.CanContentScroll="False"
             doc.Close();
             MessageBox.Show("Файл сохранен", "Готово",
                                                                     MouseDoubleClick="MovieCashierList_MouseDoubleClick">
MessageBoxButton.OK, MessageBoxImage.Information);
                                                                              <ListView.ItemContainerStyle>
                                                                                 <Style TargetType="ListViewItem">
             Process.Start(saveFileDialog.FileName);
                                                                                   <Setter Property="HorizontalContentAlignment"</pre>
                                                                      Value="Stretch" />
           }
                                                                                 </Style>
         }
                                                                              </ListView.ItemContainerStyle>
      catch (Exception ex)
                                                                              <ListView.View>
                                                                                <GridView>
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                   <GridViewColumn x:Name="MovieCodeGrid">
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                     <Label Content="Код фильма"
                                                                     Style="{DynamicResource MainLableStyle}"/>
                                                                                     <GridViewColumn.CellTemplate>
    }
                                                                                       <DataTemplate>
  }
                                                                                          <Label Content="{Binding movieCashierID}"</pre>
}
                                                                     HorizontalAlignment="Center" Style="{DynamicResource
MovieCashierControls.xaml
                                                                     MainLableStyle \}"/>
<Page x:Class="Cinema.MovieCashierControls"
                                                                                       </DataTemplate>
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                                     </GridViewColumn.CellTemplate>
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                                   </GridViewColumn>
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                                   <GridViewColumn x:Name="ImageGrid">
compatibility/2006"
                                                                                     <Label Content="Постер" Style="{DynamicResource
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                     MainLableStyle \} "/>
   xmlns:local="clr-namespace:Cinema"
                                                                                     <GridViewColumn.CellTemplate>
   mc:Ignorable="d"
                                                                                       <DataTemplate>
   d:DesignHeight="450" d:DesignWidth="800"
                                                                                          <Image Source="{Binding movieCashierCover}"</pre>
   Title="MovieCashierControls" SizeChanged="Page_SizeChanged"
                                                                     MaxHeight="400" HorizontalAlignment="Center"/>
Loaded="Page_Loaded">
                                                                                       </DataTemplate>
                                                                                     </GridViewColumn.CellTemplate>
  <Grid Style="{DynamicResource MainColorStyle}">
                                                                                   </GridViewColumn>
    <StackPanel>
                                                                                   <GridViewColumn x:Name="MovieInfoGrid">
      <Grid Style="{DynamicResource SecondColorStyle}">
                                                                                     <Label Content="Информация о фильме"
         <WrapPanel Margin="10,0,10,0"
                                                                     Style="{DynamicResource MainLableStyle}"/>
VerticalAlignment="Center">
                                                                                     <GridViewColumn.CellTemplate>
           <TextBlock Text="Поиск:" Margin="0,0,10,0"
                                                                                       <DataTemplate>
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                                          <StackPanel>
                                                                                            <Label Content="{Binding
                                                                     movieCashierTitle}" Style="{DynamicResource MainLableStyle}"/>
```

```
<WrapPanel>
                                                                                         </DataTemplate>
                         <Label Content="Жанр(ы):"
                                                                                      </GridViewColumn.CellTemplate>
Style="{DynamicResource MainLableStyle}"/>
                                                                                    </GridViewColumn>
                         < Label Content="{Binding
                                                                                  </GridView>
movieCashierGenre}" Style="{DynamicResource MainLableStyle}"/>
                                                                               </ListView.View>
                       </WrapPanel>
                                                                             </ListView>
                                                                           </StackPanel>
                       <WrapPanel>
                         <Label Content="Год публикации:"
                                                                         </Grid>
Style="{DynamicResource MainLableStyle}"/>
                                                                      </Page>
                         <Label Content="{Binding
movieCashierYearOfPublication}" Style="{DynamicResource
                                                                      MovieCashierControls.xaml.cs
MainLableStyle \}"/>
                                                                      using System;
                       </WrapPanel>
                                                                      using System.Collections.Generic;
                       <WrapPanel>
                                                                      using System.IO;
                         <Label Content="Хронометраж:"
                                                                      using System.Linq;
Style="{DynamicResource MainLableStyle}"/>
                                                                      using System.Windows;
                         < Label Content="{Binding
                                                                      using System.Windows.Controls;
movieCashierTiming}" Style="{DynamicResource MainLableStyle}"/>
                                                                      using System.Windows.Input;
                       </WrapPanel>
                                                                      using System.Windows.Media.Imaging;
                       <WrapPanel>
                                                                      using static Cinema. Authorization;
                         <Label Content="Возрастной рейтинг:"
Style="{DynamicResource MainLableStyle}"/>
                                                                      namespace Cinema
                         <Label Content="{Binding
                                                                      {
movieCashierAgeRating}" Style="{DynamicResource
                                                                        /// <summary>
MainLableStyle \} "/>
                                                                        /// Логика взаимодействия для MovieCashierControls.xaml
                       </WrapPanel>
                                                                        /// </summary>
                       <WrapPanel>
                                                                         public partial class MovieCashierControls : Page
                         <Label Content="Страна:"
Style="{DynamicResource MainLableStyle}"/>
                                                                           public MovieCashierControls()
                         <Label Content="{Binding
movieCashierCountry}" Style="{DynamicResource
                                                                             InitializeComponent();
MainLableStyle \}"/>
                                                                           }
                       </WrapPanel>
                       <WrapPanel>
                                                                           public class MovieCashierData
                         <Label Content="Актеры:"
                                                                           {
Style="{DynamicResource MainLableStyle}"/>
                                                                             public int movieCashierID { get; set; }
                         <TextBlock Text="{Binding
                                                                             public BitmapImage movieCashierCover { get; set; }
movieCashierActors}" Margin="5,0,0,0" VerticalAlignment="Center"
                                                                             public string movieCashierTitle { get; set; }
TextWrapping="Wrap" Style="{DynamicResource
                                                                             public string movieCashierGenre { get; set; }
MainTextBlockStyle}"/>
                                                                             public int movieCashierYearOfPublication { get; set; }
                       </WrapPanel>
                                                                             public double movieCashierTiming { get; set; }
                    </StackPanel>
                                                                             public string movieCashierAgeRating { get; set; }
                  </DataTemplate>
                                                                             public string movieCashierCountry { get; set; }
                </GridViewColumn.CellTemplate>
                                                                             public string movieCashierDescription { get; set; }
              </GridViewColumn>
                                                                             public string movieCashierActors { get; set; }
              <GridViewColumn x:Name="DescriptionGrid">
                                                                           }
                <Label Content="Описание"
Style="{DynamicResource MainLableStyle}"/>
                                                                           List<MovieCashierData> moviesCashierDataList = new
                <GridViewColumn.CellTemplate>
                                                                      List<MovieCashierData>();
                  <DataTemplate>
                    <TextBlock x:Name="DescriptionText"
                                                                           private void Page_Loaded(object sender, RoutedEventArgs e)
Text="{Binding movieCashierDescription}" TextWrapping="Wrap"
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                             try
```

```
actors.Name.Contains(findLine) || actors.Patronymic.Contains(findLine)
          MovieCashierList.ItemsSource = moviesCashierDataList;
                                                                          || actors.Nickname.Contains(findLine)) &&
                                                                          (session.DateAndTimeSession >= DateTime.Now))
         LoadData(null);
                                                                                                select new
       catch (Exception ex)
                                                                                                   movie.IDMovie,
                                                                                                   movie Cover
          MessageBox.Show(ex.Message, "Ошибка",
                                                                                                   movieTitle = movie.Title,
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                                   movieGenere = genere.Title,
                                                                                                   movie.YearOfPublication,
     }
                                                                                                   movie.Timing,
                                                                                                   movie.AgeRating,
    private void LoadData(string findLine)
                                                                                                   movie.Description,
                                                                                                   movieCountry = country.Title,
                                                                                                   movieActor = actors.Surname + " " +
       try
                                                                          actors.Name + " " + actors.Patronymic + " " + actors.Nickname
         using (var dataBase = new CinemaEntities())
                                                                                                ).ToList();
            moviesCashierDataList.Clear();
                                                                                      var groupedMovieCashierData = movieData.GroupBy(m
            var movieData = (from movie in dataBase.Movie
                                                                          => m.IDMovie)
                      ioin
                                                                                         .Select(g => new)
                      movieGenre in dataBase.MovieGenre on
movie.IDMovie equals movieGenre.IDMovie into movieGenereGroup
                                                                                           g.Key,
                      from movieGenere in
                                                                                           genres = g.Select(m => m.movieGenere).Distinct(),
movieGenereGroup.DefaultIfEmpty()
                                                                                           actors = g.Select(m => m.movieActor).Distinct()
                      join
                                                                                         }).ToList();
                      genre in dataBase.Genre on
movieGenere.IDGenre equals genre.IDGenre into genereGroup
                                                                                      var result = groupedMovieCashierData.Select(g =>
                      from genere in genereGroup.DefaultIfEmpty()
                                                                                         var movie = movieData.FirstOrDefault(m => m.IDMovie
                      country in dataBase.Country on
                                                                          == g.Key);
movie.IDCountry equals country.IDCountry into countryGroup
                                                                                         return new
                      from country in countryGroup.DefaultIfEmpty()
                                                                                           movie.IDMovie,
                      actorsInMovies in dataBase.ActorsInMovies on
                                                                                           movie.Cover.
movie.IDMovie equals actorsInMovies.IDMovie into
                                                                                           movie.movieTitle,
actorsInMoviesGroup
                                                                                           movieGenres = g.genres,
                      from actorsInMovie in
                                                                                           movie.YearOfPublication,
actorsInMoviesGroup.DefaultIfEmpty()
                                                                                           movie.Timing,
                      join
                                                                                           movie.AgeRating,
                      actors in dataBase.Actor on
                                                                                           movie.Description,
actorsInMovie.IDActor equals actors.IDActor into actorsGroup
                                                                                           movie.movieCountry,
                      from actors in actorsGroup.DefaultIfEmpty()
                                                                                           movieActors = g.actors
                      join
                                                                                         };
                      session in dataBase.Session on movie.IDMovie
                                                                                      }).ToList();
equals session.IDMovie into sessionGroup
                      from session in sessionGroup.DefaultIfEmpty()
                                                                                      foreach (var movieLine in result)
                      where ((findLine == null \parallel
movie. Title. Contains (find Line) \parallel genere. Title. Contains (find Line) \parallel
                                                                                         MovieCashierData movieDataClass = new
movie.YearOfPublication.ToString().Contains(findLine) ||
                                                                          MovieCashierData();
movie. Description. Contains (find Line) \parallel
country. Title. Contains (find Line) \parallel actors. Surname. Contains (find Line) \parallel
                                                                                         movieDataClass.movieCashierID = movieLine.IDMovie;
```

```
if (movieActorCounterTemp !=
              if (movieLine.Cover != null)
                                                                        movieLine.movieActors.Count())
                BitmapImage cover = new BitmapImage();
                                                                                           movieActorTemp += actor + ", ";
                cover.BeginInit();
                                                                                        else
                cover.StreamSource = new
MemoryStream(movieLine.Cover);
                                                                                           movieActorTemp += actor;
                cover.EndInit();
                movieDataClass.movieCashierCover = cover;
                                                                                        movieActorCounterTemp++;
              }
              else
                                                                                      movieDataClass.movieCashierActors =
                                                                        movieActorTemp;
                movieDataClass.movieCashierCover = new \\
BitmapImage(new Uri("pack://application:,,,/Resource/NoImage.png",
                                                                                      moviesCashierDataList.Add(movieDataClass);
UriKind.Absolute));
                                                                                    var fullMovieCashierData = (from movie in
              movieDataClass.movieCashierTitle =
                                                                        dataBase.Movie
movieLine.movieTitle;
                                                                                                    ioin
                                                                                                    session in dataBase.Session on
              string movieGenereTemp = "";
                                                                        movie.IDMovie equals session.IDMovie into sessionGroup
              int movieGenresCounterTemp = 1;
                                                                                                    from session in
              foreach (var genere in movieLine.movieGenres)
                                                                        sessionGroup.DefaultIfEmpty()
                                                                                                    where (session.DateAndTimeSession >=
                if (movieGenresCounterTemp !=
                                                                        DateTime.Now)
movieLine.movieGenres.Count())
                                                                                                    group movie by movie. Title into
                  movieGenereTemp += genere + ", ";
                                                                        groupedMovies
                                                                                                    select new
                else
                   movieGenereTemp += genere;
                                                                                                      MovieTitle = groupedMovies.Key,
                movieGenresCounterTemp++;
                                                                                                      TotalSessions =
                                                                        groupedMovies.Count()
              movie Data Class. movie Cashier Genre = \\
                                                                                                    }).ToList();
movieGenereTemp;
                                                                                    FindCounterData.Text = result.Count() + "/" +
              movie Data Class. movie Cashier Year Of Publication = \\
                                                                        fullMovieCashierData.Count();
movieLine.YearOfPublication;
              movie Data Class. movie Cashier Timing = \\
                                                                                    MovieCashierList.Items.Refresh();
movieLine.Timing;
              movie Data Class. movie Cashier Age Rating = \\
                                                                               }
movieLine.AgeRating + "+";
                                                                               catch (Exception ex)
              movie Data Class. movie Cashier Description = \\
                                                                                  MessageBox.Show(ex.Message, "Ошибка",
movieLine.Description;
              movie Data Class. movie Cashier Country = \\
                                                                        MessageBoxButton.OK,\,MessageBoxImage.Error);
movieLine.movieCountry;
                                                                               }
                                                                             }
              string movieActorTemp = "";
              int movieActorCounterTemp = 1;
                                                                             private void Page_SizeChanged(object sender,
              foreach (var actor in movieLine.movieActors)
                                                                        SizeChangedEventArgs e)
                                                                               try
```

```
xmlns:local="clr-namespace:Cinema"
         MovieCodeGrid.Width = 0;
                                                                        mc:Ignorable="d"
                                                                        d:DesignHeight="450" d:DesignWidth="800"
         ImageGrid.Width = ActualWidth - ActualWidth * 0.8;
         MovieInfoGrid.Width = ActualWidth - ActualWidth * 0.7;
                                                                        Title="MovieControls" SizeChanged="Page_SizeChanged"
         DescriptionGrid.Width = ActualWidth - ActualWidth * 0.5;
                                                                     Loaded="Page_Loaded">
         MovieCashierList.Height = ActualHeight - 35;
                                                                       <Grid Style="{DynamicResource MainColorStyle}">
      catch (Exception ex)
                                                                          <StackPanel>
                                                                            <Grid Style="{DynamicResource SecondColorStyle}">
                                                                              <WrapPanel Margin="10,0,0,0,0" VerticalAlignment="Center">
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                <Button x:Name="AddMovie" Margin="0,0,5,0"
      }
                                                                     Content="Добавить" HorizontalAlignment="Left" Width="120"
                                                                     Style="{DynamicResource TabControlButtonStyle}"
    }
                                                                     Click="AddMovie_Click"/>
                                                                                <Button x:Name="RemoveMovie" Content="Удалить"
    private void FindData_TextChanged(object sender,
TextChangedEventArgs e)
                                                                     HorizontalAlignment="Left" Width="120" Style="{DynamicResource
                                                                     TabControlButtonStyle}" Click="RemoveMovie_Click"/>
    {
      LoadData(FindData.Text);
                                                                                <Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                     SecondColorSeparatorsRectangleStyle}"/>
    }
                                                                                <TextBlock Text="Поиск:" Margin="0,0,10,0"
    private void MovieCashierList_MouseDoubleClick(object sender,
                                                                     Style="{DynamicResource MainTextBlockStyle}"/>
MouseButtonEventArgs e)
                                                                                <TextBox x:Name="FindData" Width="200"
                                                                     Style="{DynamicResource MainTextBoxStyle}"
                                                                     TextChanged="FindData_TextChanged"/>
      try
                                                                                < Rectangle Margin="10,0,10,0" Style="{DynamicResource
         var selectedCashierMovie = MovieCashierList.SelectedItem
                                                                     SecondColorSeparatorsRectangleStyle}"/>
                                                                                <TextBlock Text="Найдено: " Style="{DynamicResource
as MovieCashierData;
         if (selectedCashierMovie != null)
                                                                     MainTextBlockStyle \}"/>
                                                                                <TextBlock x:Name="FindCounterData" Text="0/0"
           TransmittedData.idSelectedCashierMovie =
                                                                     Style="{DynamicResource MainTextBlockStyle}"/>
                                                                              </WrapPanel>
selectedCashierMovie.movieCashierID:
                                                                            </Grid>
           CashierPanel = Window.GetWindow(this) as
CashierPanel;
                                                                            <ListView x:Name="MovieList" Height="400"</pre>
           cashierPanel.SessionPageOpen();
                                                                     d:ItemsSource="{d:SampleData ItemCount=2}"
                                                                     ScrollViewer.CanContentScroll="False"
         }
                                                                     MouseDoubleClick="MovieList_MouseDoubleClick">
      }
      catch (Exception ex)
                                                                              <ListView.ItemContainerStyle>
                                                                                <Style TargetType="ListViewItem">
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                   <Setter Property="HorizontalContentAlignment"
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                     Value="Stretch" />
                                                                                </Style>
                                                                              </ListView.ItemContainerStyle>
    }
                                                                              <ListView.View>
  }
                                                                                <GridView>
                                                                                   <GridViewColumn x:Name="MovieCodeGrid">
MovieControls.xaml
                                                                                     <Label Content="Код фильма"
<Page x:Class="Cinema.MovieControls"
                                                                     Style="{DynamicResource MainLableStyle}"/>
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                                     <GridViewColumn.CellTemplate>
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                                       <DataTemplate>
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                                         <Label Content="{Binding movieID}"</pre>
compatibility/2006"
                                                                     HorizontalAlignment="Center" Style="{DynamicResource
```

MainLableStyle \}"/>

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

```
</DataTemplate>
                                                                                              <Label Content="Актеры:"
                </GridViewColumn.CellTemplate>
                                                                     Style="{DynamicResource MainLableStyle}"/>
              </GridViewColumn>
                                                                                              <TextBlock Text="{Binding movieActors}"
                                                                     Margin="5,0,0,0" VerticalAlignment="Center" TextWrapping="Wrap"
              <GridViewColumn x:Name="ImageGrid">
                <Label Content="Постер" Style="{DynamicResource
                                                                     Style="{DynamicResource MainTextBlockStyle}"/>
MainLableStyle \}"/>
                                                                                            </WrapPanel>
                <GridViewColumn.CellTemplate>
                                                                                         </StackPanel>
                  <DataTemplate>
                                                                                       </DataTemplate>
                    <Image Source="{Binding movieCover}"</pre>
                                                                                     </GridViewColumn.CellTemplate>
MaxHeight="400" HorizontalAlignment="Center"/>
                                                                                   </GridViewColumn>
                  </DataTemplate>
                                                                                   <GridViewColumn x:Name="DescriptionGrid">
                </GridViewColumn.CellTemplate>
                                                                                     <Label Content="Описание"
              </GridViewColumn>
                                                                     Style="{DynamicResource MainLableStyle}"/>
              <GridViewColumn x:Name="MovieInfoGrid">
                                                                                     <GridViewColumn.CellTemplate>
                <Label Content="Информация о фильме"
                                                                                       <DataTemplate>
Style="{DynamicResource MainLableStyle}"/>
                                                                                          <TextBlock Text="{Binding movieDescription}"
                <GridViewColumn.CellTemplate>
                                                                     TextWrapping="Wrap" Style="{DynamicResource
                                                                     MainTextBlockStyle}"/>
                  <DataTemplate>
                    <StackPanel>
                                                                                       </DataTemplate>
                      <Label Content="{Binding movieTitle}"</pre>
                                                                                     </GridViewColumn.CellTemplate>
Style="{DynamicResource MainLableStyle}"/>
                                                                                   </GridViewColumn>
                      <WrapPanel>
                                                                                 </GridView>
                                                                              </ListView.View>
                         <Label Content="Жанр(ы):"
Style="{DynamicResource MainLableStyle}"/>
                                                                            </ListView>
                         <Label Content="{Binding movieGenre}"</pre>
                                                                          </StackPanel>
Style="{DynamicResource MainLableStyle}"/>
                                                                        </Grid>
                      </WrapPanel>
                                                                     </Page>
                      <WrapPanel>
                         <Label Content="Год публикации:"
                                                                     MovieControls.xaml.cs
Style="{DynamicResource MainLableStyle}"/>
                                                                     using System;
                         < Label Content="{Binding
                                                                     using System.Collections.Generic;
movieYearOfPublication}" Style="{DynamicResource
                                                                     using System.IO;
MainLableStyle \} "/>
                                                                     using System.Ling;
                      </WrapPanel>
                                                                     using System. Windows;
                      <WrapPanel>
                                                                     using System.Windows.Controls;
                         <Label Content="Хронометраж:"
                                                                     using System.Windows.Input;
Style="{DynamicResource MainLableStyle}"/>
                                                                     using System.Windows.Media.Imaging;
                         <Label Content="{Binding movieTiming}"</pre>
                                                                     using static Cinema. Authorization;
Style="{DynamicResource MainLableStyle}"/>
                      </WrapPanel>
                                                                     namespace Cinema
                      <WrapPanel>
                         <Label Content="Возрастной рейтинг:"
                                                                       /// <summary>
Style="{DynamicResource MainLableStyle}"/>
                                                                       /// Логика взаимодействия для MovieControls.xaml
                        < Label Content="{Binding
                                                                       /// </summary>
movieAgeRating}" Style="{DynamicResource MainLableStyle}"/>
                                                                        public partial class MovieControls: Page
                      </WrapPanel>
                      <WrapPanel>
                                                                          public MovieControls()
                         <Label Content="Страна:"
                                                                          {
Style="{DynamicResource MainLableStyle}"/>
                                                                            InitializeComponent();
                         <Label Content="{Binding movieCountry}"</pre>
Style="{DynamicResource MainLableStyle}"/>
                      </WrapPanel>
                                                                          public class MovieData
                      <WrapPanel>
                                                                          {
```

```
public int movieID { get; set; }
                                                                                                 actorsInMovies in dataBase.ActorsInMovies on
       public BitmapImage movieCover { get; set; }
                                                                          movie.IDMovie equals actorsInMovies.IDMovie into
       public string movieTitle { get; set; }
                                                                          actorsInMoviesGroup
       public string movieGenre { get; set; }
                                                                                                from actorsInMovie in
       public int movieYearOfPublication { get; set; }
                                                                          actorsInMoviesGroup.DefaultIfEmpty()
       public double movieTiming { get; set; }
                                                                                                join
       public string movieAgeRating { get; set; }
                                                                                                actors in dataBase.Actor on
       public string movieCountry { get; set; }
                                                                          actorsInMovie.IDActor equals actors.IDActor into actorsGroup
       public string movieDescription { get; set; }
                                                                                                from actors in actorsGroup.DefaultIfEmpty()
       public string movieActors { get; set; }
                                                                                                 where (findLine == null \parallel
                                                                          movie.Title.Contains(findLine) || genere.Title.Contains(findLine) ||
                                                                          movie.YearOfPublication.ToString().Contains(findLine) ||
    List<MovieData> moviesDataList = new List<MovieData>();
                                                                          movie.Description.Contains(findLine) ||
                                                                          country. Title. Contains (find Line) \parallel actors. Surname. Contains (find Line) \parallel
    private void Page_Loaded(object sender, RoutedEventArgs e)
                                                                          actors.Name.Contains(findLine) || actors.Patronymic.Contains(findLine)
                                                                          \parallel actors.Nickname.Contains(findLine))
                                                                                                 select new
       try
                                                                                                   movie.IDMovie,
         MovieList.ItemsSource = moviesDataList:
                                                                                                   movie.Cover,
         LoadData(null);
                                                                                                   movieTitle = movie.Title,
                                                                                                   movieGenere = genere.Title,
       catch (Exception ex)
                                                                                                   movie.YearOfPublication,
       {
                                                                                                   movie.Timing,
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                                   movie.AgeRating,
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                                   movie.Description,
                                                                                                   movieCountry = country.Title,
     }
                                                                                                   movieActor = actors.Surname + " " +
                                                                          actors.Name + " " + actors.Patronymic + " " + actors.Nickname
    private void LoadData(string findLine)
                                                                                                ).ToList();
       try
                                                                                      var groupedMovieData = movieData.GroupBy(m =>
         using (var dataBase = new CinemaEntities())
                                                                          m.IDMovie)
                                                                                         .Select(g => new
            moviesDataList.Clear();
                                                                                           g.Key,
            var movieData = (from movie in dataBase.Movie
                                                                                           genres = g.Select(m => m.movieGenere).Distinct(),
                                                                                           actors = g.Select(m => m.movieActor).Distinct()
                      movieGenre in dataBase.MovieGenre on
                                                                                         }).ToList();
movie.IDMovie equals movieGenre.IDMovie into movieGenereGroup
                      from movieGenere in
                                                                                      var result = groupedMovieData.Select(g =>
movieGenereGroup.DefaultIfEmpty()
                                                                                         var movie = movieData.FirstOrDefault(m => m.IDMovie
                      join
                      genre in dataBase.Genre on
                                                                          == g.Key);
movieGenere.IDGenre equals genre.IDGenre into genereGroup
                                                                                         return new
                      from genere in genereGroup.DefaultIfEmpty()
                                                                                           movie.IDMovie,
                      join
                      country in dataBase.Country on
                                                                                           movie.Cover.
movie.IDCountry equals country.IDCountry into countryGroup
                                                                                           movie.movieTitle,
                      from country in countryGroup.DefaultIfEmpty()
                                                                                           movieGenres = g.genres,
                                                                                           movie.YearOfPublication,
                      join
                                                                                           movie.Timing,
```

```
movie.AgeRating,
                                                                                    movieDataClass.movieCountry =
                movie.Description,
                                                                       movieLine.movieCountry;
                movie.movieCountry,
                movieActors = g.actors
                                                                                    string movieActorTemp = "";
                                                                                    int movieActorCounterTemp = 1;
           }).ToList();
                                                                                    foreach (var actor in movieLine.movieActors)
           foreach (var movieLine in result)
                                                                                       if (movieActorCounterTemp !=
                                                                       movieLine.movieActors.Count())
              MovieData movieDataClass = new MovieData();
                                                                                         movieActorTemp += actor + ", ";
              movieDataClass.movieID = movieLine.IDMovie;
                                                                                       else
              if (movieLine.Cover != null)
                                                                                         movieActorTemp += actor;
                BitmapImage cover = new BitmapImage();
                cover.BeginInit();
                                                                                       movieActorCounterTemp++;
                cover.StreamSource = new
MemoryStream(movieLine.Cover);
                                                                                    movieDataClass.movieActors = movieActorTemp;
                cover.EndInit();
                                                                                    moviesDataList.Add(movieDataClass);
                movieDataClass.movieCover = cover;
                                                                                  }
              }
              else
                                                                                  var fullMovieData = dataBase.Movie.ToList();
                                                                                  FindCounterData.Text = result.Count() + "/" +
                movieDataClass.movieCover = new BitmapImage(new
                                                                       fullMovieData.Count();
Uri("pack://application:,,,/Resource/NoImage.png", UriKind.Absolute));
                                                                                  MovieList.Items.Refresh();
                                                                                }
              movieDataClass.movieTitle = movieLine.movieTitle;
                                                                             catch (Exception ex)
              string movieGenereTemp = "";
              int movieGenresCounterTemp = 1;
                                                                                MessageBox.Show(ex.Message, "Ошибка",
              foreach (var genere in movieLine.movieGenres)
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
                if (movieGenresCounterTemp !=
                                                                           }
movieLine.movieGenres.Count())
                  movieGenereTemp += genere + ", ";
                                                                           private void Page_SizeChanged(object sender,
                                                                       SizeChangedEventArgs e)
                else
                  movieGenereTemp += genere;
                                                                             try
                movieGenresCounterTemp++;
                                                                                MovieCodeGrid.Width = 0;
                                                                                ImageGrid.Width = ActualWidth - ActualWidth * 0.8;
              movieDataClass.movieGenre = movieGenereTemp;
                                                                                MovieInfoGrid.Width = ActualWidth - ActualWidth * 0.7;
              movieDataClass.movieYearOfPublication =
                                                                                DescriptionGrid.Width = ActualWidth - ActualWidth * 0.5;
movieLine.YearOfPublication;
                                                                                MovieList.Height = ActualHeight - 35;
              movieDataClass.movieTiming = movieLine.Timing;
              movieDataClass.movieAgeRating =
                                                                              catch (Exception ex)
movieLine.AgeRating + "+";
             movie Data Class. movie Description = \\
                                                                                MessageBox.Show(ex.Message, "Ошибка",
movieLine.Description;
                                                                       MessageBoxButton.OK,\,MessageBoxImage.Error);
```

```
private void RemoveMovie_Click(object sender, RoutedEventArgs
       }
                                                                      e)
    private void MovieList_MouseDoubleClick(object sender,
                                                                             try
MouseButtonEventArgs e)
                                                                               var selectedMovie = MovieList.SelectedItem as MovieData;
     {
      try
                                                                               if (selectedMovie != null)
         var selectedMovie = MovieList.SelectedItem as MovieData;
                                                                                  if (MessageBox.Show("Вы действительно хотите удалить
         if (selectedMovie != null)
                                                                      данный фильм?", "Внимание", MessageBoxButton. YesNo,
                                                                      MessageBoxImage.Warning) == MessageBoxResult.Yes)
           TransmittedData.idSelectedMovie =
selectedMovie.movieID;
                                                                                    using (var dataBase = new CinemaEntities())
           AddOrEditMovie\ addOrEditMovie\ =\ new
                                                                                      var removeMovie = dataBase.Movie.Where(w =>
AddOrEditMovie();
                                                                      w.IDMovie == selectedMovie.movieID).FirstOrDefault();
           addOrEditMovie.ShowDialog();
                                                                                      var removeMovieGenere =
                                                                      dataBase.MovieGenre.Where(w => w.IDMovie ==
           LoadData(FindData.Text);
                                                                      selectedMovie.movieID).ToList();
                                                                                      var removeActorsInMovies =
         }
                                                                      dataBase.ActorsInMovies.Where(w => w.IDMovie ==
       catch (Exception ex)
                                                                      selectedMovie.movieID).ToList();
                                                                                      var removeTicket = (from session in dataBase.Session
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                                 ticket in dataBase.Ticket on
                                                                      session.IDSession equals ticket.IDSession
     }
                                                                                                 where (session.IDMovie ==
                                                                      selectedMovie.movieID)
    private void FindData_TextChanged(object sender,
                                                                                                 select ticket).ToList();
TextChangedEventArgs e)
                                                                                      var removeSession = dataBase.Session.Where(w =>
                                                                      w.IDMovie == selectedMovie.movieID).ToList();
      LoadData(FindData.Text);
                                                                                      if (removeSession.Count > 0)
     }
    private void AddMovie_Click(object sender, RoutedEventArgs e)
                                                                                        if (MessageBox.Show("Удаляемый фильм
                                                                      присутсвует в сесии. Вы действительно хотите его удалить?",
                                                                      "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Warning)
       try
                                                                      == MessageBoxResult.No)
         TransmittedData.idSelectedMovie = -1;
                                                                                           return;
         AddOrEditMovie addOrEditMovie = new AddOrEditMovie();
         addOrEditMovie.ShowDialog();
                                                                                      data Base. Ticket. Remove Range (remove Ticket);\\
                                                                                      dataBase.Session.RemoveRange(removeSession);
         LoadData(FindData.Text);
                                                                      data Base. Actors In Movies. Remove Range (remove Actors In Movies); \\
       catch (Exception ex)
                                                                      dataBase.MovieGenre.RemoveRange(removeMovieGenere);
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                      dataBase.Movie.Remove(removeMovie);
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                      dataBase.SaveChanges();
                                                                                      MessageBox.Show("Данные были удаленны",
                                                                      "Готово", MessageBoxButton.OK, MessageBoxImage.Information);
```

```
<StackPanel>
             LoadData(FindData.Text);
                                                                                  <Label Content="Цена билета"
                                                                      HorizontalAlignment="Center" Style="{DynamicResource
           }
                                                                      MainLableStyle \}"/>
         }
                                                                                  <Label x:Name="TicketPrice" Content=""</pre>
       }
      catch (Exception ex)
                                                                      HorizontalAlignment="Center" Style="{DynamicResource
                                                                      MainLableStyle \}"/>
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                </StackPanel>
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                <StackPanel>
                                                                                  <Label Content="Дата и время сеанса"
                                                                      HorizontalAlignment="Center" Style="{DynamicResource
    }
  }
                                                                       MainLableStyle}"/>
                                                                                  <Label x:Name="DateTimeSession" Content=""
}
                                                                       HorizontalAlignment="Center" Style="{DynamicResource
                                                                      MainLableStyle \}"/>
PlacesCashierControls.xaml
<Page x:Class="Cinema.PlacesCashierControls"
                                                                                </StackPanel>
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                                <Button x:Name="BookingTicket" Margin="10,0,10,10"
                                                                       Content="Забронировать" Style="{DynamicResource
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                      TabControlButtonStyle}" Click="BookingTicket_Click"/>
compatibility/2006"
                                                                              </StackPanel>
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                           </Grid>
   xmlns:local="clr-namespace:Cinema"
                                                                         </Grid>
   mc:Ignorable="d"
                                                                       </Page>
   d:DesignHeight="550" d:DesignWidth="1050"
   Title="PlacesCashierControls" Background="White"
                                                                       PlacesCashierControls.xaml.cs
Loaded="Page_Loaded" SizeChanged="Page_SizeChanged">
                                                                       using iTextSharp.text;
                                                                       using iTextSharp.text.pdf;
  <Grid Style="{DynamicResource MainColorStyle}">
                                                                      using QRCoder;
     <StackPanel x:Name="Hall" Margin="0,0,198,0"
                                                                       using System;
VerticalAlignment="Center" HorizontalAlignment="Center"
                                                                       using
SizeChanged="Hall_SizeChanged"/>
                                                                       System.Data.Entity.Core.Common.CommandTrees.ExpressionBuilder;
                                                                       using System.Diagnostics;
       <Grid VerticalAlignment="Center"
HorizontalAlignment="Right" Margin="10,0,0,0,0" Width="200">
                                                                       using System.IO;
       <Rectangle Style="{DynamicResource
                                                                       using System.Linq;
SecondColorRectanglePlaceStyle}" Opacity="0.5"/>
                                                                       using System.Text.RegularExpressions;
       <StackPanel>
                                                                       using System. Threading. Tasks;
         <TextBlock Text="Бронирование билета"
                                                                       using System.Windows;
Margin="0,10,0,0" TextWrapping="Wrap" Style="{DynamicResource
                                                                       using System.Windows.Controls;
TitleTextBlockStyle}"/>
                                                                       using System.Windows.Media;
         <StackPanel>
                                                                       using System.Xml.Linq;
           <Label Content="Ряд" HorizontalAlignment="Center"
                                                                       using static Cinema. Authorization;
Style="{DynamicResource MainLableStyle}"/>
           <Label x:Name="SelectedRow" Content="Het"</pre>
                                                                      namespace Cinema
HorizontalAlignment="Center" Style="{DynamicResource
                                                                       {
MainLableStyle \} "/>
                                                                         /// <summary>
         </StackPanel>
                                                                         /// Логика взаимодействия для PlacesCashierControls.xaml
         <StackPanel>
           <Label Content="Место" HorizontalAlignment="Center"
                                                                         public partial class PlacesCashierControls: Page
Style="{DynamicResource MainLableStyle}"/>
                                                                         {
           <Label x:Name="SelectedPlace" Content="Het"</pre>
                                                                           public PlacesCashierControls()
HorizontalAlignment="Center" Style="{DynamicResource
MainLableStyle \} "/>
                                                                              InitializeComponent();
```

</StackPanel>

button.Background = Brushes.White;

```
public int selectedRowNumber;
                                                                                             button.Margin = new Thickness(0, 0, 5, 0);
    public int selectedPlaceNumber;
                                                                                             button.Click += HallButton_Click;
     public Button selectedRowPlaceButton;
                                                                                             if (settingsData.HiddenPlaces != null)
     private void Page_Loaded(object sender, RoutedEventArgs e)
                                                                                               string[] hidePlaceTemp =
       LoadData();
                                                                          settingsData.HiddenPlaces.Split('|');
                                                                                               foreach (var hidePlaceLine in hidePlaceTemp)
     }
                                                                                                  Match match = Regex.Match(hidePlaceLine,
    private void LoadData()
                                                                          @"Row(\d+)Place(\d+)");
       try
                                                                                                  if (match.Success)
         using (var dataBase = new CinemaEntities())
                                                                                                    if (row + 1 ==
            var ticketData = dataBase.Ticket.Where(w => w.IDSession
                                                                          int.Parse(match.Groups[1].Value) + 1 && place ==
                                                                          int.Parse(match.Groups[2].Value))
== TransmittedData.idSelectedCashierSession).ToList();
                                                                                                       button.Click -= HallButton_Click;
            var settingsData = dataBase.Settings.OrderByDescending(o
=> o.DateTimeChange).FirstOrDefault();
                                                                                                       button. Opacity = 0;
                                                                                                      button.IsEnabled = false;
            if (settingsData != null)
              Hall.Children.Clear();
              for (int row = 0; row < settingsData.RowHall; row++)
                                                                                             foreach (var ticketLine in ticketData)
                 WrapPanel wrapPanel = new WrapPanel();
                                                                                               if (row + 1 == ticketLine.RowNumber && place
                TextBlock textBlockBegin = new TextBlock();
                                                                          == ticketLine.PlaceNumber)
                 textBlockBegin.Text = $"Ряд {row + 1}";
                                                                                               {
                 textBlockBegin.FontSize = 10;
                                                                                                  button.Foreground = Brushes.Red;
                 textBlockBegin.Margin = new Thickness(0, 0, 5, 0);
                                                                                                  button.IsEnabled = false;
                 textBlockBegin.VerticalAlignment =
VerticalAlignment.Center;
                TextBlock textBlockEnd = new TextBlock();
                                                                                             wrapPanel.Children.Add(button);
                 textBlockEnd.Text = $"Ряд {row + 1}";
                 textBlockEnd.FontSize = 10;
                                                                                           wrapPanel.Children.Add(textBlockEnd);
                textBlockEnd.VerticalAlignment = \\
VerticalAlignment.Center;
                                                                                           StackPanel stackPanel = new StackPanel();
                                                                                           stackPanel.Margin = new Thickness(0, 0, 0, 5);
                                                                                           stack Panel. Horizontal Alignment = \\
                 wrapPanel.Children.Add(textBlockBegin);
                                                                          HorizontalAlignment.Center;
                 for (int place = 1; place <= settingsData.PlaceHall;
place++)
                                                                                           stackPanel.Children.Add(wrapPanel);
                   Button button = new Button();
                                                                                           Hall.Children.Add(stackPanel);
                   button.Content = place;
                   button.Name = $"Row{row}Place{place}";
                                                                                      }
                   button.Width = 18;
                                                                                      else
                   button.Height = 18;
                   button.FontSize = 10;
                                                                                        Label lable = new Label();
```

```
lable.FontSize = 26;
                                                                                                                                                       double newHeight = Hall.ActualHeight * scale;
                          lable.Foreground = Brushes.Red;
                          lable.Content = "Зал не размечен
                                                                                                                                                       if (newWidth > maxWidth)
администратором.\rОбратитесь с системному администратору.";
                         lable. Horizontal Alignment = \\
                                                                                                                                                           scale = maxWidth / Hall.ActualWidth;
Horizontal Alignment. Center;
                         lable. Horizontal Content Alignment = \\
                                                                                                                                                       if (newHeight > maxHeight)
Horizontal Alignment. Center;
                                                                                                                                                           scale = Math.Min(scale, maxHeight / Hall.ActualHeight);
                         Hall.Children.Add(lable);
                     }
                                                                                                                                                       ScaleTransform scaleTransform = new ScaleTransform(scale,
                     var sessionData = dataBase.Session.Where(w =>
                                                                                                                                      scale, centerX, centerY);
w.IDSession ==
                                                                                                                                                       Hall.RenderTransform = scaleTransform;
TransmittedData.idSelectedCashierSession).FirstOrDefault();
                     TicketPrice.Content = \\
                                                                                                                                                   catch (Exception ex)
session Data. Ticket Price. To String (). Remove (session Data. Ticket Price. To String (). The properties of the prop
String().Length - 2, 2) + " Py6.";
                                                                                                                                                       MessageBox.Show(ex.Message, "Ошибка",
                     DateTimeSession.Content = \\
                                                                                                                                     MessageBoxButton.OK,\,MessageBoxImage.Error);
sessionData.DateAndTimeSession;
                                                                                                                                                   }
                                                                                                                                              }
                 }
             }
                                                                                                                                              private void HallButton_Click(object sender, RoutedEventArgs e)
             catch (Exception ex)
                 MessageBox.Show(ex.Message, "Ошибка",
                                                                                                                                                  try
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                                                                                       Button clickedButton = sender as Button;
         }
                                                                                                                                                       if (clickedButton != null)
        private void Page_SizeChanged(object sender,
SizeChangedEventArgs e)
                                                                                                                                                           if (selectedRowPlaceButton != null)
        {
                                                                                                                                                           {
            ResizeWindows();
                                                                                                                                                                selectedRowPlaceButton.Background = Brushes.White;
                                                                                                                                                                selectedRowPlaceButton.Foreground = Brushes.Black;
                                                                                                                                                                selectedRowPlaceButton.BorderBrush = (Brush)(new
         private void Hall_SizeChanged(object sender,
                                                                                                                                      BrushConverter().ConvertFrom("#FF707070"));
SizeChangedEventArgs e)
            ResizeWindows();
                                                                                                                                                           Match match = Regex.Match(clickedButton.Name,
                                                                                                                                      @"Row(\d+)Place(\d+)");
        private void ResizeWindows()
                                                                                                                                                           if (match.Success)
         {
                                                                                                                                                                selectedRowNumber = int.Parse(match.Groups[1].Value)
             try
                                                                                                                                      + 1:
                 double maxWidth = this.ActualWidth - 10;
                                                                                                                                                               selectedPlaceNumber =
                 double maxHeight = this.ActualHeight - 10;
                                                                                                                                     int.Parse(match.Groups[2].Value);
                 double scale = this.ActualWidth / (Hall.ActualWidth * 1.28);
                                                                                                                                                                clickedButton.Background = Brushes.Aqua;
                 double centerX = Hall.ActualWidth / 2;
                                                                                                                                                                clickedButton.Foreground = Brushes.Black;
                 double centerY = Hall.ActualHeight / 2;
                                                                                                                                                                clickedButton.BorderBrush = Brushes.Black;
                                                                                                                                                                selectedRowPlaceButton = clickedButton;
                 double newWidth = Hall.ActualWidth * scale;
                                                                                                                                                           }
```

```
SelectedPlaceChange();
                                                                          private async Task PrintTicket(int selectedTicket)
       }
       catch (Exception ex)
       {
                                                                            try
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                              using (var dataBase = new CinemaEntities())
                                                                                 var ticketData = (from ticket in dataBase.Ticket
    private void SelectedPlaceChange()
                                                                                           session in dataBase.Session on ticket.IDSession
                                                                     equals session.IDSession into sessionGroup
                                                                                           from session in sessionGroup.DefaultIfEmpty()
       try
         if (selectedRowNumber != 0)
                                                                                           movie in dataBase.Movie on session.IDMovie
                                                                     equals movie.IDMovie into movieGroup
                                                                                           from movie in movieGroup.DefaultIfEmpty()
           SelectedRow.Content = selectedRowNumber;
         }
                                                                                          join
         else
                                                                                          employee in dataBase.Employee on
                                                                     ticket.IDEmployee equals employee.IDEmployee into employeeGroup
           SelectedRow.Content = "Het";
                                                                                           from employee in
                                                                     employee Group. Default If Empty()\\
                                                                                           where (ticket.IDTicket == selectedTicket)
         if (selectedPlaceNumber != 0)
                                                                                           select new
           SelectedPlace.Content = selectedPlaceNumber;
                                                                                             ticket.IDTicket,
         }
                                                                                             ticket.RowNumber,
                                                                                             ticket.PlaceNumber,
         else
                                                                                             ticket.DateTimeBooking,
           SelectedPlace.Content = "Het";
                                                                                             movie = movie. Title,
                                                                                             session.DateAndTimeSession.
         }
                                                                                             employee.Surname,
       catch (Exception ex)
                                                                                             employee.Name,
                                                                                             employee.Patronymic,
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                             session.TicketPrice
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                           }).FirstOrDefault();
       }
                                                                                 string tempFilePath = Path.GetTempFileName() + ".pdf";
     }
    private void ClearPlaceChange()
                                                                                 string qrCodeText =
                                                                     ession}|{ticketData.Surname} {ticketData.Name}
      try
                                                                      {ticketData.Patronymic}|{Math.Round(ticketData.TicketPrice,2)}";
         selectedRowNumber = 0;
                                                                                 QRCodeGenerator qrGenerator = new QRCodeGenerator();
         selectedPlaceNumber = 0;
                                                                                 QRCodeData qrCodeData =
         selectedRowPlaceButton = null;
                                                                     grGenerator.CreateQrCode(grCodeText,
         SelectedPlaceChange();
                                                                      QRCodeGenerator.ECCLevel.Q);
                                                                                 QRCode qrCode = new QRCode(qrCodeData);
       catch (Exception ex)
                                                                                 using (System.Drawing.Bitmap qrCodeBitmap =
                                                                     qrCode.GetGraphic(20, System.Drawing.Color.Black,
                                                                     System.Drawing.Color.White, null))
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK,\,MessageBoxImage.Error);
```

```
using (MemoryStream ms = new MemoryStream())
                                                                                                                                                  doc.Add(paragraph7);
                           qrCodeBitmap.Save(ms,
                                                                                                                                                  iTextSharp.text.Image image =
System.Drawing.Imaging.ImageFormat.Png); // Сохраняем в PNG
                                                                                                                       iTextSharp.text.Image.GetInstance(qrCodeBytes);\\
                           byte[] qrCodeBytes = ms.ToArray();
                                                                                                                                                  image.ScaleToFit(200, 200);
                                                                                                                                                  doc.Add(image);
                           Document doc = new Document();
                           PdfWriter writer = PdfWriter.GetInstance(doc, new
                                                                                                                                                  doc.Close();
FileStream($@"{tempFilePath}", FileMode.Create));
                           BaseFont baseFont =
BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf",
                                                                                                                                          Process.Start(tempFilePath);
BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
                           Font font = new Font(baseFont, 16);
                                                                                                                                   catch (Exception ex)
                           BaseFont baseFontHead =
BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf",
                                                                                                                                      MessageBox.Show(ex.Message, "Ошибка",
BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
                                                                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
                           Font fontHead = new Font(baseFont, 20, Font.BOLD);
                                                                                                                                   }
                                                                                                                               }
                           doc.Open();
                                                                                                                              private void BookingTicket_Click(object sender, RoutedEventArgs
                           Paragraph mainParagraph = new Paragraph("Билет",
                                                                                                                       e)
fontHead);
                                                                                                                               {
                           mainParagraph.Alignment =
                                                                                                                                   try
Element.ALIGN_CENTER;
                           Paragraph paragraph = new Paragraph("Homep
                                                                                                                                       if (selectedRowNumber == 0 && selectedPlaceNumber == 0)
билета: " + ticketData.IDTicket, font);
                           Paragraph paragraph1 = new Paragraph("Фильм: " +
                                                                                                                                           MessageBox.Show("Место в зале не выбранно",
                                                                                                                       "Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);
ticketData.movie. font):
                           Paragraph paragraph2 = new Paragraph("Ряд: " +
                                                                                                                                          return;
ticketData.RowNumber, font);
                                                                                                                                       }
                           Paragraph paragraph3 = new Paragraph("Mecto: " +
ticketData.PlaceNumber, font);
                                                                                                                                       using (var dataBase = new CinemaEntities())
                           Paragraph paragraph4 = new Paragraph("Дата и время
                                                                                                                                       {
начала сеанса: " + ticketData.DateAndTimeSession, font);
                                                                                                                                          if (MessageBox.Show("Забронировать место на сеанс?",
                           Paragraph paragraph5 = new Paragraph("Кассир: " +
                                                                                                                       "Внимание", MessageBoxButton. YesNo, MessageBoxImage. Warning)
ticketData.Surname + " " + ticketData.Name + " " +
                                                                                                                        == MessageBoxResult.Yes)
ticketData.Patronymic, font);
                           Paragraph paragraph6 = new Paragraph("Дата
                                                                                                                                              var newTicket = new Ticket();
продажи: " + ticketData.DateTimeBooking, font);
                           Paragraph paragraph7 = new Paragraph("Цена: " +
                                                                                                                                              newTicket.IDSession =
ticketData. TicketPrice. ToString(). Remove (ticketData. TicketPrice. ToString()) and ticketData. TicketPrice. ToString() and ticketData. TicketData
                                                                                                                       TransmittedData.idSelectedCashierSession;
ng().Length - 2, 2) + " Pyő.", font);
                                                                                                                                              newTicket.RowNumber = selectedRowNumber:
                                                                                                                                              newTicket.PlaceNumber = selectedPlaceNumber;
                           doc.Add(mainParagraph);
                                                                                                                                              newTicket.IDEmployee = TransmittedData.idEmployee;
                           doc.Add(paragraph);
                                                                                                                                              newTicket.DateTimeBooking = DateTime.Now;
                           doc.Add(paragraph1);
                           doc.Add(paragraph2);
                                                                                                                                              dataBase.Ticket.Add(newTicket);
                           doc.Add(paragraph3);
                                                                                                                                              dataBase.SaveChanges();
                           doc.Add(paragraph4);
                           doc.Add(paragraph5);
                                                                                                                                              MessageBox.Show("Место на сеанс зарезервировано",
                           doc.Add(paragraph6);
                                                                                                                       "Готово", MessageBoxButton.OK, MessageBoxImage.Information);
```

```
<DatePicker x:Name="FindSessionData"
             Task.Run(() => PrintTicket(newTicket.IDTicket));
                                                                     Style="{DynamicResource SecondDatePickerStyle}"
                                                                     SelectedDateChanged="FindSessionData_SelectedDateChanged"/>
                                                                                 <Rectangle Margin="10,0,10,0" Style="{DynamicResource
             selectedRowPlaceButton.Background = (Brush)(new
BrushConverter().ConvertFrom("#FFDDDDDD"));
                                                                     SecondColorSeparatorsRectangleStyle}"/>
             selectedRowPlaceButton.BorderBrush = (Brush)(new
                                                                                 <TextBlock Text="Найдено: " Style="{DynamicResource
BrushConverter().ConvertFrom("#FF707070"));
                                                                     MainTextBlockStyle \} "/>
             selectedRowPlaceButton.Foreground = Brushes.Red;
                                                                                <TextBlock x:Name="FindCounterData" Text="0/0"
             selectedRowPlaceButton.IsEnabled = false;
                                                                     Style="{DynamicResource MainTextBlockStyle}"/>
                                                                              </WrapPanel>
                                                                            </Grid>
             ClearPlaceChange();
           }
                                                                            <ListView x:Name="SessionCashierList" Height="400"</p>
         }
                                                                     d:ItemsSource="{d:SampleData ItemCount=2}"
                                                                     ScrollViewer.CanContentScroll="False"
      catch (Exception ex)
                                                                     MouseDoubleClick="SessionCashierList_MouseDoubleClick">
         MessageBox.Show(ex.Message, "Ошибка",
                                                                              <ListView.ItemContainerStyle>
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                 <Style TargetType="ListViewItem">
                                                                                   <Setter Property="HorizontalContentAlignment"</pre>
    }
                                                                      Value="Stretch" />
                                                                                 </Style>
  }
}
                                                                              </ListView.ItemContainerStyle>
                                                                              <ListView.View>
SessionCashierControls.xaml
                                                                                <GridView>
<Page x:Class="Cinema.SessionCashierControls"
                                                                                   <GridViewColumn x:Name="SessionCodeGrid">
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                                     <Label Content="Код сессии"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                     Style="{DynamicResource MainLableStyle}"/>
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                                     <GridViewColumn.CellTemplate>
compatibility/2006"
                                                                                       <DataTemplate>
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                                         <Label Content="{Binding sessionCashierID}"</pre>
   xmlns:local="clr-namespace:Cinema"
                                                                     HorizontalAlignment="Center" Style="{DynamicResource
   mc:Ignorable="d"
                                                                     MainLableStyle \}"/>
   d:DesignHeight="450" d:DesignWidth="800"
                                                                                       </DataTemplate>
   Title="SessionControls" SizeChanged="Page_SizeChanged"
                                                                                     </GridViewColumn.CellTemplate>
Loaded="Page_Loaded">
                                                                                   </GridViewColumn>
                                                                                   <GridViewColumn x:Name="SessionCoverGrid">
  <Grid Style="{DynamicResource MainColorStyle}">
                                                                                     <Label Content="Постер" Style="{DynamicResource
    <StackPanel>
                                                                     MainLableStyle \} "/>
      <Grid Style="{DynamicResource SecondColorStyle}">
                                                                                     <GridViewColumn.CellTemplate>
         <WrapPanel Margin="10,0,10,0"
                                                                                       <DataTemplate>
VerticalAlignment="Center">
                                                                                         <Image Source="{Binding</pre>
           <TextBlock Text="Поиск:" Margin="0,0,10,0"
                                                                     sessionCashierMovieCover}" MaxHeight="400"
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                     HorizontalAlignment="Center"/>
           <TextBox x:Name="FindData" Width="200"
                                                                                       </DataTemplate>
Style="{DynamicResource MainTextBoxStyle}"
                                                                                     </GridViewColumn.CellTemplate>
TextChanged="FindData_TextChanged"/>
                                                                                   </GridViewColumn>
           < Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                                   <GridViewColumn x:Name="SessionInfoGird">
SecondColorSeparatorsRectangleStyle}"/>
                                                                                     <Label Content="Информация о фильме"
           <CheckBox x:Name="FindDateChecker" Margin="0,4,5,0"</pre>
                                                                     Style="{DynamicResource MainLableStyle}"/>
Style="{DynamicResource MainCheckBoxStyle}"
                                                                                     <GridViewColumn.CellTemplate>
Click="FindDateChecker_Click"/>
                                                                                       <DataTemplate>
                                                                                         <StackPanel>
           <TextBlock Text="Дата сеанса:" Margin="0,0,10,0"
```

Style="{DynamicResource MainTextBlockStyle}"/>

```
<Label Content="{Binding
                                                                                     <GridViewColumn.CellTemplate>
sessionCashierMovieTitle}" Style="{DynamicResource
                                                                                       <DataTemplate>
MainLableStyle \}"/>
                                                                                         <StackPanel>
                                                                                            <WrapPanel>
                      <WrapPanel>
                        <Label Content="Жанр(ы):"
                                                                                              <Label Content="Дата начала сеанса:"
Style="{DynamicResource MainLableStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
                        < Label Content="{Binding
                                                                                              < Label Content="{Binding
sessionCashierMovieGenre}" Style="{DynamicResource
                                                                     sessionCashierDateSessionStart}" Style="{DynamicResource
MainLableStyle \}"/>
                                                                     MainLableStyle \}"/>
                      </WrapPanel>
                                                                                            </WrapPanel>
                      <WrapPanel>
                                                                                            <WrapPanel>
                        <Label Content="Год публикации:"
                                                                                              <Label Content="Время начала сеанса:"
Style="{DynamicResource MainLableStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
                        <Label Content="{Binding
                                                                                              < Label Content="{Binding
sessionCashierMovieYearOfPublication}" Style="{DynamicResource
                                                                     sessionCashierTimeSessionStart}" Style="{DynamicResource
MainLableStyle \} "/>
                                                                     MainLableStyle \}"/>
                      </WrapPanel>
                                                                                            </WrapPanel>
                      <WrapPanel>
                                                                                            <WrapPanel>
                                                                                              <Label Content="Места в зале:"
                        <Label Content="Хронометраж:"
Style="{DynamicResource MainLableStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
                        <Label Content="{Binding
                                                                                              < Label Content="{Binding
sessionCashierMovieTiming}" Style="{DynamicResource
                                                                     sessionSeatsInHall}" Style="{DynamicResource MainLableStyle}"/>
MainLableStyle \}"/>
                                                                                            </WrapPanel>
                                                                                         </StackPanel>
                      </WrapPanel>
                      <WrapPanel>
                                                                                       </DataTemplate>
                        <Label Content="Возрастной рейтинг:"
                                                                                     </GridViewColumn.CellTemplate>
Style="{DynamicResource MainLableStyle}"/>
                                                                                   </GridViewColumn>
                        <Label Content="{Binding
                                                                                   <GridViewColumn x:Name="SessionPriceGrid">
                                                                                     <Label Content="Цена билета"
sessionCashierMovieAgeRating}" Style="{DynamicResource
MainLableStyle \} "/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
                                                                                     <\!\!Grid View Column. Cell Template\!\!>
                      </WrapPanel>
                      <WrapPanel>
                                                                                       <DataTemplate>
                        <Label Content="Страна:"
                                                                                         < Label Content="{Binding
                                                                     sessionCashierTicketPrice}" Style="{DynamicResource
Style="{DynamicResource MainLableStyle}"/>
                        <Label Content="{Binding
                                                                     MainLableStyle}"/>
sessionCashierMovieCountry}" Style="{DynamicResource
                                                                                       </DataTemplate>
MainLableStyle \}"/>
                                                                                     </GridViewColumn.CellTemplate>
                      </WrapPanel>
                                                                                   </GridViewColumn>
                      <WrapPanel>
                                                                                </GridView>
                        <Label Content="Актеры:"
                                                                              </ListView.View>
                                                                            </ListView>
Style="{DynamicResource MainLableStyle}"/>
                        <TextBlock Text="{Binding
                                                                          </StackPanel>
sessionCashierActors}" Margin="5,0,0,0" VerticalAlignment="Center"
                                                                        </Grid>
TextWrapping="Wrap" Style="{DynamicResource
                                                                     </Page>
MainTextBlockStyle \}"/>
                      </WrapPanel>
                                                                     SessionCashierControls.xaml.cs
                    </StackPanel>
                                                                     using System;
                  </DataTemplate>
                                                                     using System.Collections.Generic;
               </GridViewColumn.CellTemplate>
                                                                     using System.IO;
             </GridViewColumn>
                                                                     using System.Linq;
             <GridViewColumn x:Name="SessionDateTimeGrid">
                                                                     using System. Windows;
               <Label Content="Информация о сеансе"
                                                                     using System.Windows.Controls;
Style="{DynamicResource MainLableStyle}"/>
                                                                     using System.Windows.Input;
```

```
using System.Windows.Media.Imaging;
using static Cinema. Authorization;
                                                                                    using (var dataBase = new CinemaEntities())
                                                                                      sessionCashierDataList.Clear();
namespace Cinema
                                                                                      DatePicker endDate = new DatePicker();
  /// <summary>
                                                                                      endDate.SelectedDate = null;
  /// Логика взаимодействия для SessionCashierControls.xaml
  /// </summary>
  public partial class SessionCashierControls: Page
                                                                                      if (findDate.SelectedDate != null)
    public SessionCashierControls()
                                                                                        endDate.SelectedDate =
                                                                          findDate.SelectedDate.Value.AddDays(1);
       InitializeComponent();
                                                                                      }
                                                                                      else
    public class SessionCashierData
                                                                                        endDate.SelectedDate = null;
       public int sessionCashierID { get; set; }
       public BitmapImage sessionCashierMovieCover { get; set; }
                                                                                      var sessionCashierData = (from session in dataBase.Session
       public string sessionCashierMovieTitle { get; set; }
                                                                                                     ioin
       public string sessionCashierMovieGenre { get; set; }
                                                                                                     movie in dataBase.Movie on
       public int sessionCashierMovieYearOfPublication { get; set; }
                                                                          session.IDMovie equals movie.IDMovie into movieGroup
       public double sessionCashierMovieTiming { get; set; }
                                                                                                     from movie in
       public string sessionCashierMovieAgeRating { get; set; }
                                                                          movieGroup.DefaultIfEmpty()
       public string sessionCashierMovieCountry { get; set; }
       public string sessionCashierActors { get; set; }
                                                                                                     movieGenre in dataBase.MovieGenre on
       public string sessionCashierDateSessionStart { get; set; }
                                                                          movie.IDMovie equals movieGenre.IDMovie into movieGenereGroup
       public string sessionCashierTimeSessionStart { get; set; }
                                                                                                     from movieGenre in
                                                                          movie Genere Group. Default If Empty()\\
       public string sessionCashierTicketPrice { get; set; }
       public string sessionSeatsInHall { get; set; }
                                                                                                     ioin
     }
                                                                                                     actorsInMovies in
                                                                          dataBase.ActorsInMovies on movie.IDMovie equals
     List<SessionCashierData> sessionCashierDataList = new
                                                                          actorsInMovies.IDMovie into actorsInMoviesGroup
List<SessionCashierData>();
                                                                                                     from actorsInMovies in
                                                                          actorsInMoviesGroup.DefaultIfEmpty()
    private void Page_Loaded(object sender, RoutedEventArgs e)
                                                                                                     actors in dataBase.Actor on
                                                                          actorsInMovies.IDActor equals actors.IDActor into actorsGroup
       try
                                                                                                     from actors in
                                                                          actorsGroup.DefaultIfEmpty()
         SessionCashierList.ItemsSource = sessionCashierDataList:
         LoadData(null, FindSessionData);
                                                                                                     genre in dataBase.Genre on
                                                                          movieGenre.IDGenre equals genre.IDGenre into genreGroup
       catch (Exception ex)
                                                                                                     from genre in genreGroup.DefaultIfEmpty()
       {
                                                                                                     join
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                                     country in dataBase.Country on
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                          movie.IDCountry equals country.IDCountry into countryGroup
                                                                                                     from country in
                                                                          countryGroup.DefaultIfEmpty()
    private void LoadData(string findLine, DatePicker findDate)
                                                                                                     ticket in dataBase.Ticket on
                                                                          session.IDSession equals ticket.IDSession into ticketGroup
                                                                                                     from ticket in ticketGroup.DefaultIfEmpty()
       try
```

```
where ((findLine == null \parallel
                                                                                            session.TicketPrice,
movie. Title. Contains (find Line) \parallel genre. Title. Contains (find Line) \parallel
                                                                                            movieActors = g.actors,
movie. Year Of Publication. To String (). Contains (find Line) \parallel
                                                                                            g.TicketCount,
movie. Description. Contains (find Line) \parallel
                                                                                         };
country.Title.Contains(findLine)) && ((findDate.SelectedDate == null
                                                                                       }).ToList();
&& endDate.SelectedDate == null) || (session.DateAndTimeSession >=
findDate.SelectedDate.Value && session.DateAndTimeSession <=
                                                                                       foreach (var sessionLine in result)
endDate.SelectedDate.Value)) && (session.DateAndTimeSession >=
DateTime.Now) && (session.IDMovie ==
                                                                                         SessionCashierData sessionDataClass = new
TransmittedData.idSelectedCashierMovie))
                                                                           SessionCashierData();
                            select new
                                                                                         sessionDataClass.sessionCashierID =
                              session.IDSession,
                                                                           sessionLine.IDSession;
                              movie.Cover,
                              movieTitle = movie.Title,
                                                                                         if (sessionLine.Cover != null)
                              movieGenere = genre.Title,
                              movie.YearOfPublication,
                                                                                            BitmapImage cover = new BitmapImage();
                              movie.Timing,
                              movie.AgeRating,
                                                                                            cover.BeginInit();
                              movieCountry = country.Title,
                                                                                            cover.StreamSource = new
                              movieActors = actors.Surname + " " +
                                                                           MemoryStream(sessionLine.Cover);
actors.Name + " " + actors.Patronymic + " " + actors.Nickname,
                                                                                           cover.EndInit();
                              session.DateAndTimeSession,
                              TicketID = ticket == null ? 0 :
                                                                                            sessionDataClass.sessionCashierMovieCover = cover;
ticket.IDTicket,
                              session.TicketPrice
                                                                                         else
                            }).ToList();
                                                                                            sessionDataClass.sessionCashierMovieCover = new
            var\ grouped Session Cashier Data =
                                                                           BitmapImage(new Uri("pack://application:,,,/Resource/NoImage.png",
sessionCashierData.GroupBy(m => m.IDSession)
                                                                           UriKind.Absolute));
              .Select(g => new
              {
                g.Key,
                                                                                         sessionDataClass.sessionCashierMovieTitle =
                 genres = g.Select(m => m.movieGenere).Distinct(),
                                                                           sessionLine.movieTitle;
                 actors = g.Select(m => m.movieActors).Distinct(),
                TicketCount = g.Count()
                                                                                         string sessionMovieGenereTemp = "";
              }).ToList();
                                                                                         int movieGenresCounterTemp = 1;
                                                                                         foreach (var genere in sessionLine.sessionMovieGenres)
            var result = groupedSessionCashierData.Select(g =>
                                                                                            if (movieGenresCounterTemp !=
            {
              var session = sessionCashierData.FirstOrDefault(m =>
                                                                           sessionLine.sessionMovieGenres.Count())\\
m.IDSession == g.Key);
                                                                                              sessionMovieGenereTemp += genere + ", ";
              return new
                                                                                            else
                                                                                              sessionMovieGenereTemp += genere;
                 session.IDSession,
                 session.Cover,
                                                                                            movieGenresCounterTemp++;
                 session.movieTitle,
                 sessionMovieGenres = g.genres,
                                                                                         sessionDataClass.sessionCashierMovieGenre =
                 session. Year Of Publication,
                                                                           sessionMovieGenereTemp;
                 session.Timing,
                                                                                         string movieActorTemp = "";
                 session.AgeRating,
                 session.movieCountry,
                                                                                         int movieActorCounterTemp = 1;
                 session.DateAndTimeSession,
                                                                                         foreach (var actor in sessionLine.movieActors)
```

```
else
                             if (movieActorCounterTemp !=
sessionLine.movieActors.Count())
                                                                                                                                                                   sessionDataClass.sessionSeatsInHall =
                                                                                                                                  purchasedTickets + "/" + totalTickets;
                                  movieActorTemp += actor + ", ";
                             else
                                                                                                                                                           else
                                 movieActorTemp += actor;
                                                                                                                                                               sessionDataClass.sessionSeatsInHall = "Зал не
                                                                                                                                  размечен";
                             movieActorCounterTemp++;
                                                                                                                                                           sessionCashierDataList.Add(sessionDataClass);
                         sessionDataClass.sessionCashierActors =
movieActorTemp;
                                                                                                                                                       var fullSessionCashierData = dataBase.Session.Where(w
                                                                                                                                  => w.DateAndTimeSession >= DateTime.Now && w.IDMovie ==
                                                                                                                                 TransmittedData.idSelectedCashierMovie).ToList();
                                                                                                                                                       FindCounterData.Text = result.Count() + "/" +
session Data Class. session Cashier Movie Year Of Publication = \\
sessionLine.YearOfPublication;
                                                                                                                                  fullSessionCashierData.Count();
                         session Data Class. session Cashier Movie Timing = \\
                                                                                                                                                       SessionCashierList.Items.Refresh();
sessionLine.Timing;
                         sessionDataClass.sessionCashierMovieAgeRating =
sessionLine.AgeRating + "+";
                                                                                                                                              catch (Exception ex)
                         sessionDataClass.sessionCashierMovieCountry =
sessionLine.movieCountry;
                                                                                                                                                   MessageBox.Show(ex.Message, "Ошибка",
                         session Data Class. session Cashier Date Session Start = \\
                                                                                                                                 MessageBoxButton.OK,\,MessageBoxImage.Error);
sessionLine.DateAndTimeSession.ToString().Split(' ')[0];
                         session Data Class. session Cashier Time Session Start = \\
                                                                                                                                          }
sessionLine.DateAndTimeSession.ToString().Split(' ')[1];
                         sessionDataClass.sessionCashierTicketPrice =
                                                                                                                                          private void Page_SizeChanged(object sender,
session Line. Ticket Price. To String (). Remove (session Line. Ticket Price. To String ()) and the session Line of the price of the 
                                                                                                                                 SizeChangedEventArgs e)
String().Length -2, 2) + "py6.";
                                                                                                                                          {
                                                                                                                                              try
                         var seatsHallData =
dataBase.Settings.OrderByDescending(o =>
                                                                                                                                                   SessionCodeGrid.Width = 0;
o.IDSettings).FirstOrDefault();
                                                                                                                                                   SessionCoverGrid.Width = ActualWidth - ActualWidth * 0.8;
                         if (seatsHallData != null)
                                                                                                                                                   SessionInfoGird.Width = ActualWidth - ActualWidth * 0.7;
                                                                                                                                                   SessionDateTimeGrid.Width = ActualWidth - ActualWidth *
                                                                                                                                 0.7;
                             int hidePlaceCount =
seatsHallData.HiddenPlaces.Split('|').Count();
                                                                                                                                                   SessionPriceGrid.Width = ActualWidth - ActualWidth * 0.8;
                                                                                                                                                   SessionCashierList.Height = ActualHeight - 35;
                             int purchasedTickets = sessionLine.TicketCount /
((movieGenresCounterTemp - 1) * (movieActorCounterTemp - 1));
                                                                                                                                              catch (Exception ex)
                             int totalTickets = (seatsHallData.RowHall *
seatsHallData.PlaceHall) - hidePlaceCount;
                                                                                                                                                   MessageBox.Show(ex.Message, "Ошибка",
                                                                                                                                  MessageBoxButton.OK, MessageBoxImage.Error);
                             if (purchasedTickets == 1)
                                                                                                                                              }
                                                                                                                                          }
                                 var ticketCount = dataBase.Ticket.Where(w =>
w.IDSession == sessionLine.IDSession).Count();
                                                                                                                                          private void FindData_TextChanged(object sender,
                                 session Data Class. session Seats In Hall = ticket Count \\
                                                                                                                                 TextChangedEventArgs e)
+ "/" + totalTickets;
                                                                                                                                              try
```

```
}
         if (FindDateChecker.IsChecked == false)
           Find Session Data. Selected Date = null; \\
                                                                            private void SessionCashierList_MouseDoubleClick(object sender,
                                                                       MouseButtonEventArgs e)
         LoadData(FindData.Text, FindSessionData);
                                                                            {
                                                                              try
       catch (Exception ex)
                                                                                 var selectedCashierSession = SessionCashierList.SelectedItem
         MessageBox.Show(ex.Message, "Ошибка",
                                                                        as SessionCashierData;
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                if (selectedCashierSession != null)
       }
                                                                                   Transmitted Data.id Selected Cashier Session = \\
     }
                                                                        selectedCashierSession.sessionCashierID;
    private void FindDateChecker_Click(object sender,
RoutedEventArgs e)
                                                                                   CashierPanel cashierPanel = Window.GetWindow(this) as
                                                                        CashierPanel:
     {
                                                                                   cashierPanel.PlacesPageOpen();
       try
                                                                                 }
         if (FindDateChecker.IsChecked == false)
                                                                               }
                                                                              catch (Exception ex)
           FindSessionData.IsEnabled = false;
           FindSessionData.SelectedDate = null;
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
         }
         else
           FindSessionData.IsEnabled = true;
           FindSessionData.SelectedDate = DateTime.Now.Date;
         LoadData(FindData.Text, FindSessionData);
                                                                        SessionControls.xaml
                                                                        <Page x:Class="Cinema.SessionControls"
       catch (Exception ex)
                                                                        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
         MessageBox.Show(ex.Message, "Ошибка",
                                                                        xmlns:mc="http://schemas.openxmlformats.org/markup-
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                        compatibility/2006"
                                                                        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
       }
                                                                           xmlns:local="clr-namespace:Cinema"
     }
                                                                           mc:Ignorable="d"
    private\ void\ FindSessionData\_SelectedDateChanged (object\ sender,
                                                                           d:DesignHeight="450" d:DesignWidth="800"
SelectionChangedEventArgs e)
                                                                           Title="SessionControls" SizeChanged="Page_SizeChanged"
                                                                       Loaded="Page_Loaded">
     {
       try
                                                                          <Grid Style="{DynamicResource MainColorStyle}">
         if (FindDateChecker.IsChecked == false)
                                                                            <StackPanel>
                                                                               <Grid Style="{DynamicResource SecondColorStyle}">
           FindSessionData.SelectedDate = null;
                                                                                 <WrapPanel Margin="10,0,10,0"
                                                                        VerticalAlignment="Center">
         LoadData(FindData.Text, FindSessionData);
                                                                                   <Button x:Name="AddSession" Margin="0,0,5,0"
                                                                        Content="Добавить" HorizontalAlignment="Center" Width="120"
       catch (Exception ex)
                                                                        Style="{DynamicResource TabControlButtonStyle}"
                                                                        Click="AddSession_Click"/>
         MessageBox.Show(ex.Message, "Ошибка",
```

 $MessageBoxButton.OK,\,MessageBoxImage.Error);$ 

```
<Button x:Name="RemoveSession" Content="Удалить"
                                                                                     <Label Content="Постер" Style="{DynamicResource
HorizontalAlignment="Left" Width="120" Style="{DynamicResource
                                                                     MainLableStyle \}"/>
TabControlButtonStyle}" Click="RemoveSession_Click"/>
                                                                                     <GridViewColumn.CellTemplate>
           < Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                                       <DataTemplate>
SecondColorSeparatorsRectangleStyle}"/>
                                                                                         <Image Source="{Binding sessionMovieCover}"</pre>
           <TextBlock Text="Поиск:" Margin="0,0,10,0"
                                                                     MaxHeight="400" HorizontalAlignment="Center"/>
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                                       </DataTemplate>
           <TextBox x:Name="FindData" Width="200"
                                                                                     </GridViewColumn.CellTemplate>
Style="{DynamicResource MainTextBoxStyle}"
                                                                                  </GridViewColumn>
TextChanged="FindData_TextChanged"/>
                                                                                   <GridViewColumn x:Name="SessionInfoGird">
                                                                                     <Label Content="Информация о фильме"
           <Rectangle Margin="10,0,10,0" Style="{DynamicResource
SecondColorSeparatorsRectangleStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
           <CheckBox x:Name="FindDateChecker" Margin="0,4,5,0"</p>
                                                                                    <GridViewColumn.CellTemplate>
Style="{DynamicResource MainCheckBoxStyle}"
                                                                                       <DataTemplate>
Click="FindDateChecker_Click"/>
                                                                                         <StackPanel>
           <TextBlock Text="Дата сеанса:" Margin="0,0,10,0"
                                                                                           < Label Content="{Binding
                                                                     sessionMovieTitle}" Style="{DynamicResource MainLableStyle}"/>
Style="{DynamicResource MainTextBlockStyle}"/>
           <DatePicker x:Name="FindSessionData"</pre>
                                                                                           <WrapPanel>
Style="{DynamicResource SecondDatePickerStyle}"
                                                                                             <Label Content="Жанр(ы):"
Selected Date Changed = "Find Session Data\_Selected Date Changed"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
           < Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                                             < Label Content="{Binding
SecondColorSeparatorsRectangleStyle}"/>
                                                                     sessionMovieGenre}" Style="{DynamicResource MainLableStyle}"/>
           <TextBlock Text="Найдено: " Style="{DynamicResource
                                                                                           </WrapPanel>
MainTextBlockStyle}"/>
                                                                                           <WrapPanel>
           <TextBlock x:Name="FindCounterData" Text="0/0"
                                                                                             <Label Content="Год публикации:"
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
         </WrapPanel>
                                                                                             < Label Content="{Binding
       </Grid>
                                                                     sessionMovieYearOfPublication}" Style="{DynamicResource
                                                                     MainLableStyle \} "/>
       <ListView x:Name="SessionList" Height="400"</pre>
                                                                                           </WrapPanel>
d:ItemsSource="{d:SampleData ItemCount=2}"
                                                                                           <WrapPanel>
ScrollViewer.CanContentScroll="False"
                                                                                             <Label Content="Хронометраж:"
MouseDoubleClick="SessionList MouseDoubleClick">
                                                                     Style="{DynamicResource MainLableStyle}"/>
         <ListView.ItemContainerStyle>
                                                                                             <Label Content="{Binding
           <Style TargetType="ListViewItem">
                                                                     sessionMovieTiming}" Style="{DynamicResource MainLableStyle}"/>
             <Setter Property="HorizontalContentAlignment"
                                                                                           </WrapPanel>
Value="Stretch" />
                                                                                           <WrapPanel>
                                                                                              <Label Content="Возрастной рейтинг:"
         </ListView.ItemContainerStyle>
                                                                     Style="{DynamicResource MainLableStyle}"/>
         <ListView.View>
                                                                                             <Label Content="{Binding
           <GridView>
                                                                     sessionMovieAgeRating}" Style="{DynamicResource
             <GridViewColumn x:Name="SessionCodeGrid">
                                                                     MainLableStyle \}"/>
                <Label Content="Код сессии"
                                                                                           </WrapPanel>
Style="{DynamicResource MainLableStyle}"/>
                                                                                           <WrapPanel>
                <GridViewColumn.CellTemplate>
                                                                                             <Label Content="Страна:"
                  <DataTemplate>
                                                                     Style="{DynamicResource MainLableStyle}"/>
                    <Label Content="{Binding sessionID}"
                                                                                             <Label Content="{Binding
HorizontalAlignment="Center" Style="{DynamicResource
                                                                     sessionMovieCountry}" Style="{DynamicResource
MainLableStyle \} "/>
                                                                     MainLableStyle \}"/>
                  </DataTemplate>
                                                                                           </WrapPanel>
                </GridViewColumn.CellTemplate>
                                                                                           <WrapPanel>
             </GridViewColumn>
                                                                                             <Label Content="Актеры:"
```

Style="{DynamicResource MainLableStyle}"/>

<GridViewColumn x:Name="SessionCoverGrid">

```
<TextBlock Text="{Binding
                                                                        using System.Collections.Generic;
sessionActors}" Margin="5,0,0,0" VerticalAlignment="Center"
                                                                        using System.IO;
TextWrapping="Wrap" Style="{DynamicResource
                                                                        using System.Linq;
MainTextBlockStyle}"/>
                                                                        using System. Windows;
                       </WrapPanel>
                                                                        using System.Windows.Controls;
                     </StackPanel>
                                                                        using System.Windows.Input;
                   </DataTemplate>
                                                                        using System.Windows.Media.Imaging;
                </GridViewColumn.CellTemplate>
                                                                        using static Cinema. Authorization;
              </GridViewColumn>
              <GridViewColumn
                                                                        namespace Cinema
x:Name="SessionInformationSeasonGrid">
                                                                        {
                <Label Content="Информация"
                                                                          /// <summary>
Style="{DynamicResource MainLableStyle}"/>
                                                                          /// Логика взаимодействия для SessionControls.xaml
                <GridViewColumn.CellTemplate>
                                                                          /// </summary>
                                                                          public partial class SessionControls: Page
                   <DataTemplate>
                     <StackPanel>
                       <WrapPanel>
                                                                            public SessionControls()
                          <Label Content="Дата начала сеанса:"
Style = "\{DynamicResource\ MainLableStyle\}"/>
                                                                               InitializeComponent();
                         < Label Content="{Binding
sessionDateSessionStart}" Style="{DynamicResource
MainLableStyle \}"/>
                                                                            public class SessionData
                       </WrapPanel>
                       <WrapPanel>
                                                                               public int sessionID { get; set; }
                          <Label Content="Время начала сеанса:"
                                                                               public BitmapImage sessionMovieCover { get; set; }
Style="{DynamicResource MainLableStyle}"/>
                                                                               public string sessionMovieTitle { get; set; }
                          <Label Content="{Binding
                                                                               public string sessionMovieGenre { get; set; }
sessionTimeSessionStart}" Style="{DynamicResource
                                                                               public int sessionMovieYearOfPublication { get; set; }
MainLableStyle \} "/>
                                                                               public double sessionMovieTiming { get; set; }
                       </WrapPanel>
                                                                               public string sessionMovieAgeRating { get; set; }
                     </StackPanel>
                                                                               public string sessionMovieCountry { get; set; }
                   </DataTemplate>
                                                                               public string sessionActors { get; set; }
                </GridViewColumn.CellTemplate>
                                                                               public string sessionDateSessionStart { get; set; }
              </GridViewColumn>
                                                                               public string sessionTimeSessionStart { get; set; }
              <GridViewColumn x:Name="SessionPriceGrid">
                                                                               public string sessionTicketPrice { get; set; }
                <Label Content="Цена билета"
Style="{DynamicResource MainLableStyle}"/>
                <GridViewColumn.CellTemplate>
                                                                            List<SessionData> sessionDataList = new List<SessionData>();
                  <DataTemplate>
                     <Label Content="{Binding sessionTicketPrice}"</pre>
                                                                            private void Page_Loaded(object sender, RoutedEventArgs e)
Style="{DynamicResource MainLableStyle}"/>
                                                                            {
                  </DataTemplate>
                                                                               try
                </GridViewColumn.CellTemplate>
              </GridViewColumn>
                                                                                 SessionList.ItemsSource = sessionDataList;
            </GridView>
         </ListView.View>
                                                                                 LoadData(null, FindSessionData);
       </ListView>
     </StackPanel>
                                                                               catch (Exception ex)
   </Grid>
</Page>
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
SessionControls.xaml.cs
using System;
                                                                            }
```

```
movie.Description.Contains(findLine) ||
     private void LoadData(string findLine, DatePicker findDate)
                                                                           country.Title.Contains(findLine)) && ((findDate.SelectedDate == null
                                                                            && endDate.SelectedDate == null) || (session.DateAndTimeSession >=
                                                                           findDate.SelectedDate.Value && session.DateAndTimeSession <=
       try
                                                                           endDate.SelectedDate.Value)))
          using (var dataBase = new CinemaEntities())
                                                                                                   select new
            sessionDataList.Clear();
                                                                                                      session.IDSession,
                                                                                                      movie.Cover,
            DatePicker endDate = new DatePicker();
                                                                                                      movieTitle = movie.Title,
            endDate.SelectedDate = null;
                                                                                                      movieGenere = genere.Title,
                                                                                                      movieActors = actors.Surname + " " +
            if (findDate.SelectedDate != null)
                                                                           actors.Name + " " + actors.Patronymic + " " + actors.Nickname,
                                                                                                      movie. Year Of Publication,
              endDate.SelectedDate =
                                                                                                      movie. Timing,
find Date. Selected Date. Value. Add Days (1);\\
                                                                                                      movie.AgeRating,
                                                                                                      movieCountry = country.Title,
            }
                                                                                                      session.DateAndTimeSession,
            else
                                                                                                      session.TicketPrice
              endDate.SelectedDate = null;
                                                                                                   }).ToList();
                                                                                       var groupedSessionData = sessionData.GroupBy(m =>
            var sessionData = (from session in dataBase.Session
                                                                           m.IDSession)
                                                                                          .Select(g => new)
                        movie in dataBase.Movie on session.IDMovie
equals movie.IDMovie into movieGroup
                                                                                            g.Key,
                        from movie in movieGroup.DefaultIfEmpty()
                                                                                            genres = g.Select(m => m.movieGenere).Distinct(),
                                                                                            actors = g.Select(m => m.movieActors).Distinct(),
                        actorsInMovies in dataBase.ActorsInMovies on
                                                                                          }).ToList();
movie.IDMovie equals actorsInMovies.IDMovie into
actorsInMoviesGroup
                                                                                        var result = groupedSessionData.Select(g =>
                        from actorsInMovie in
                                                                                       {
actorsInMoviesGroup.DefaultIfEmpty()
                                                                                          var session = sessionData.FirstOrDefault(m =>
                                                                           m.IDSession == g.Key);
                        actors in dataBase.Actor on
                                                                                          return new
actorsInMovie.IDActor equals actors.IDActor into actorsGroup
                        from actors in actorsGroup.DefaultIfEmpty()
                                                                                             session.IDSession,
                                                                                             session.Cover,
                        movieGenre in dataBase.MovieGenre on
                                                                                             session.movieTitle,
movie.IDMovie equals movieGenre.IDMovie into movieGenereGroup
                                                                                             sessionMovieGenres = g.genres,
                        from movieGenere in
                                                                                             session. Year Of Publication,
movie Genere Group. Default If Empty()\\
                                                                                             session.Timing,
                                                                                             session.AgeRating,
                       join
                        genre in dataBase.Genre on
                                                                                             session.movieCountry,
movieGenere.IDGenre equals genre.IDGenre into genereGroup
                                                                                             session.DateAndTimeSession,
                        from genere in genereGroup.DefaultIfEmpty()
                                                                                             session.TicketPrice,
                                                                                             movieActors = g.actors,
                        country in dataBase.Country on
                                                                                          };
movie.IDCountry equals country.IDCountry into countryGroup
                                                                                        }).ToList();
                        from country in countryGroup.DefaultIfEmpty()
                        where ((findLine == null \parallel
                                                                                       foreach (var sessionLine in result)
movie. Title. Contains (find Line) \parallel genere. Title. Contains (find Line) \parallel
movie. Year Of Publication. To String (). Contains (find Line) \parallel
                                                                                          SessionData sessionDataClass = new SessionData();
```

```
movieActorCounterTemp++;
                          sessionDataClass.sessionID = sessionLine.IDSession;
                                                                                                                                                             sessionDataClass.sessionActors = movieActorTemp;
                         if (sessionLine.Cover != null)
                                                                                                                                                             session Data Class. session Movie Year Of Publication = \\
                              BitmapImage cover = new BitmapImage();
                                                                                                                                    sessionLine.YearOfPublication;
                                                                                                                                                             session Data Class. session Movie Timing = \\
                             cover.BeginInit();
                                                                                                                                    sessionLine.Timing;
                             cover.StreamSource = new
                                                                                                                                                             sessionDataClass.sessionMovieAgeRating =
MemoryStream(sessionLine.Cover);
                                                                                                                                    sessionLine.AgeRating + "+";
                             cover.EndInit();
                                                                                                                                                             sessionDataClass.sessionMovieCountry =
                                                                                                                                    sessionLine.movieCountry;
                              sessionDataClass.sessionMovieCover = cover;
                                                                                                                                                             sessionDataClass.sessionDateSessionStart = \\
                          }
                                                                                                                                    sessionLine.DateAndTimeSession.ToString().Split(' ')[0];
                         else
                                                                                                                                                             sessionDataClass.sessionTimeSessionStart = \\
                                                                                                                                    sessionLine.DateAndTimeSession.ToString().Split(' ')[1];
                                                                                                                                                             session Data Class. session Ticket Price = \\
                              sessionDataClass.sessionMovieCover = new
BitmapImage(new Uri("pack://application:,,,/Resource/NoImage.png",
                                                                                                                                    session Line. Ticket Price. To String (). Remove (session Line. Ticket Price. To String ()) and the session Line of the price of the 
UriKind.Absolute));
                                                                                                                                    String().Length -2, 2) + "py6.";
                         }
                                                                                                                                                             sessionDataList.Add(sessionDataClass);
                         sessionDataClass.sessionMovieTitle =
                                                                                                                                                         }
sessionLine.movieTitle;
                                                                                                                                                         var fullSessionData = dataBase.Session.ToList();
                          string sessionMovieGenereTemp = "";
                                                                                                                                                         FindCounterData.Text = result.Count() + "/" +
                         int movieGenresCounterTemp = 1;
                                                                                                                                    fullSessionData.Count();
                          foreach (var genere in sessionLine.sessionMovieGenres)
                                                                                                                                                         SessionList.Items.Refresh();
                              if (movieGenresCounterTemp !=
                                                                                                                                                     }
sessionLine.sessionMovieGenres.Count())
                                  sessionMovieGenereTemp += genere + ", ";
                                                                                                                                                 catch (Exception ex)
                             else
                                  sessionMovieGenereTemp += genere;
                                                                                                                                                     MessageBox.Show(ex.Message, "Ошибка",
                                                                                                                                   MessageBoxButton.OK,\,MessageBoxImage.Error);
                              movieGenresCounterTemp++;
                                                                                                                                                 }
                                                                                                                                            }
                          sessionDataClass.sessionMovieGenre =
sessionMovieGenereTemp;
                                                                                                                                            private void Page_SizeChanged(object sender,
                                                                                                                                   SizeChangedEventArgs e)
                          string movieActorTemp = "";
                                                                                                                                            {
                         int movieActorCounterTemp = 1;
                                                                                                                                                try
                         foreach (var actor in sessionLine.movieActors)
                                                                                                                                                     SessionCodeGrid.Width = 0;
                             if (movieActorCounterTemp !=
                                                                                                                                                     SessionCoverGrid.Width = ActualWidth - ActualWidth * 0.8;
sessionLine.movieActors.Count())
                                                                                                                                                     SessionInfoGird.Width = ActualWidth - ActualWidth * 0.6;
                                                                                                                                                     SessionInformationSeasonGrid.Width = ActualWidth -
                                  movieActorTemp += actor + ", ";
                                                                                                                                    ActualWidth * 0.8;
                                                                                                                                                     SessionPriceGrid.Width = ActualWidth - ActualWidth * 0.8;
                                                                                                                                                     SessionList.Height = ActualHeight - 35;
                              else
                                  movieActorTemp += actor;
                                                                                                                                                 catch (Exception ex)
```

```
MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                   FindSessionData.SelectedDate = null;
                                                                                 LoadData(FindData.Text, FindSessionData);
    private void FindData_TextChanged(object sender,
                                                                              catch (Exception ex)
TextChangedEventArgs e)
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
       try
       {
         if (FindDateChecker.IsChecked == false)
                                                                            }
           FindSessionData.SelectedDate = null;
                                                                            private void SessionList_MouseDoubleClick(object sender,
                                                                        MouseButtonEventArgs e)
         LoadData(FindData.Text, FindSessionData);
                                                                              try
       catch (Exception ex)
                                                                                 var selectedSession = SessionList.SelectedItem as
         MessageBox.Show(ex.Message, "Ошибка",
                                                                        SessionData:
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                 if (selectedSession != null)
                                                                                   TransmittedData.idSelectedSession =
     }
                                                                        selectedSession.sessionID;
    private void FindDateChecker_Click(object sender,
RoutedEventArgs e)
                                                                                   AddOrEditSession addOrEditSession = new
     {
                                                                        AddOrEditSession();
       try
                                                                                   addOrEditSession.ShowDialog();
         if (FindDateChecker.IsChecked == false)
                                                                                   if (FindDateChecker.IsChecked == false)
           FindSessionData.IsEnabled = false;
                                                                                      FindSessionData.SelectedDate = null;
           FindSessionData.SelectedDate = null:
                                                                                   LoadData(FindData.Text, FindSessionData);
         }
         else
           FindSessionData.IsEnabled = true;
                                                                               catch (Exception ex)
           Find Session Data. Selected Date = Date Time. Now. Date; \\
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
         LoadData(FindData.Text, FindSessionData);
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
       catch (Exception ex)
         MessageBox.Show(ex.Message, "Ошибка",
                                                                            private void AddSession_Click(object sender, RoutedEventArgs e)
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                              try
     }
                                                                                 TransmittedData.idSelectedSession = -1;
    private void FindSessionData_SelectedDateChanged(object sender,
SelectionChangedEventArgs e)
                                                                                 AddOrEditSession addOrEditSession = new
    {
                                                                        AddOrEditSession();
                                                                                 addOrEditSession.ShowDialog();
       try
                                                                                 if (FindDateChecker.IsChecked == false)
         if (FindDateChecker.IsChecked == false)
```

```
LoadData(FindData.Text, FindSessionData);
           FindSessionData.SelectedDate = null;
                                                                               }
         LoadData(FindData.Text, FindSessionData);
                                                                             catch (Exception ex)
      catch (Exception ex)
                                                                             {
                                                                               MessageBox.Show(ex.Message, "Ошибка",
         MessageBox.Show(ex.Message, "Ошибка",
                                                                      MessageBoxButton.OK, MessageBoxImage.Error);
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                          }
    }
    private void RemoveSession_Click(object sender,
RoutedEventArgs e)
                                                                      SettingsControls.xaml
                                                                      <Page x:Class="Cinema.Pages.SettingsControls"
                                                                      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      try
                                                                      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
         var selectedSession = SessionList.SelectedItem as
                                                                      xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                      compatibility/2006"
SessionData:
         if (selectedSession != null)
                                                                      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                         xmlns:local="clr-namespace:Cinema.Pages"
           if (MessageBox.Show("Вы действительно хотите удалить
                                                                         mc:Ignorable="d"
                                                                         d:DesignHeight="450" d:DesignWidth="800"
данный сеанс?", "Внимание", MessageBoxButton. YesNo,
MessageBoxImage.Warning) == MessageBoxResult.Yes)
                                                                         Title="SettingsControls" SizeChanged="Page_SizeChanged">
             using (var dataBase = new CinemaEntities())
                                                                        <Grid Style="{DynamicResource MainColorStyle}">
                                                                          <StackPanel x:Name="Hall" Margin="0,0,198,0"
                var removeSession = dataBase.Session.Where(w =>
                                                                      VerticalAlignment="Center" HorizontalAlignment="Center"
w.IDSession == selectedSession.sessionID).FirstOrDefault();
                                                                      SizeChanged="Hall_SizeChanged"/>
                                                                          <Grid VerticalAlignment="Center" HorizontalAlignment="Right"
                var removeTicket = dataBase.Ticket.Where(w =>
w.IDSession == selectedSession.sessionID).ToList();
                                                                      Width="200">
                                                                             <Rectangle Style="{DynamicResource
                if (removeTicket.Count > 0)
                                                                      SecondColorRectanglePlaceStyle}"/>
                                                                             <StackPanel>
                  if (MessageBox.Show("Удаляемый сеанс
                                                                               <TextBlock Margin="0,10,0,0" Text="Настройка зала"
присутсвует в билетах. Вы действительно хотите его удалить?",
                                                                      Style="{DynamicResource TitleTextBlockStyle}"/>
"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Warning)
                                                                               <Grid Margin="0,0,0,10" HorizontalAlignment="Center">
== MessageBoxResult.No)
                                                                                 <Label Content="Ряды" Style="{DynamicResource
                    return;
                                                                      MainLableStyle \}"/>
                                                                                 <ComboBox x:Name="SelectedRowHall"
                                                                      Margin="70,0,0,0" HorizontalAlignment="Left" Width="100"
                                                                      Style="{DynamicResource MainComboBoxStyle}"
                dataBase.Ticket.RemoveRange(removeTicket);
                dataBase.Session.Remove(removeSession);
                                                                      ItemContainerStyle="{DynamicResource
                                                                      MainSelectedComboBoxStyle}"
                                                                      SelectionChanged="SelectedRowHall_SelectionChanged"/>
                dataBase.SaveChanges();
               MessageBox.Show("Данные были удаленны",
                                                                               </Grid>
"Готово", MessageBoxButton.OK, MessageBoxImage.Information);
                                                                               <Grid Margin="0,0,0,10" HorizontalAlignment="Center">
                                                                                 <Label Content="Mecтa" FontSize="16"/>
             }
                                                                                 <ComboBox x:Name="SelectedPlaceHall"</p>
             if (FindDateChecker.IsChecked == false)
                                                                      Margin="70,0,0,0" HorizontalAlignment="Left" Width="100"
                                                                      Style="{DynamicResource MainComboBoxStyle}"
                FindSessionData.SelectedDate = null;
                                                                      ItemContainerStyle="{DynamicResource
```

```
MainSelectedComboBoxStyle}"
                                                                                     numericPlaceList.Add(i);
SelectionChanged="SelectedPlaceHall_SelectionChanged"/>
         </Grid>
         <Button x:Name="SaveHallSettings" Content="Сохранить"
                                                                                   SelectedRowHall.ItemsSource = numericRowList;
Margin="10,0,10,10" Style="{DynamicResource
                                                                                   SelectedPlaceHall.ItemsSource = numericPlaceList;
TabControlButtonStyle}" Click="SaveHallSettings_Click"/>
       </StackPanel>
                                                                                   using (var dataBase = new CinemaEntities())
     </Grid>
                                                                                     var loadData = dataBase.Settings.OrderByDescending(o =>
  </Grid>
</Page>
                                                                         o.DateTimeChange).FirstOrDefault();
SettingsControls.xaml.cs
                                                                                     if (loadData != null)
using System;
using System.Collections.Generic;
                                                                                        SelectedRowHall.SelectedItem = loadData.RowHall;
                                                                                        SelectedPlaceHall.SelectedItem = loadData.PlaceHall;
using
System. Data. Entity. Core. Common. Command Trees. Expression Builder; \\
                                                                                        if (loadData.HiddenPlaces != null)
using System.Linq;
                                                                                          Create Hall (load Data. Hidden Places. Split ('|'));\\
using System.Text.RegularExpressions;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
namespace Cinema.Pages
                                                                                catch (Exception ex)
{
  /// <summary>
                                                                                   MessageBox.Show(ex.Message, "Ошибка",
                                                                         MessageBoxButton.OK,\,MessageBoxImage.Error);
  /// Логика взаимодействия для SettingsControls.xaml
  /// </summary>
  public partial class SettingsControls : Page
                                                                              }
     public SettingsControls()
                                                                              private void Page_SizeChanged(object sender,
                                                                         SizeChangedEventArgs e)
       InitializeComponent();
                                                                              {
       LoadData();
                                                                                ResizeWindows();
     public int selectedRowHall;
                                                                              private void Hall_SizeChanged(object sender,
     public int selectedPlaceHall;
                                                                         SizeChangedEventArgs e)
     public List<string> hidePlace = new List<string>();
                                                                                ResizeWindows();
     private void LoadData()
                                                                              private void ResizeWindows()
       try
         List<int> numericRowList = new List<int>();
                                                                                try
         List<int> numericPlaceList = new List<int>();
                                                                                   double maxWidth = this.ActualWidth - 10;
         for (int i = 1; i < 21; i++)
                                                                                   double maxHeight = this.ActualHeight - 10;
            numericRowList.Add(i);
                                                                                   double scale = this.ActualWidth / (Hall.ActualWidth * 1.28);
                                                                                   double centerX = Hall.ActualWidth / 2;
                                                                                   double centerY = Hall.ActualHeight / 2;
         for (int i = 1; i < 31; i++)
                                                                                   double newWidth = Hall.ActualWidth * scale;
```

```
double newHeight = Hall.ActualHeight * scale;
                                                                                try
         if (newWidth > maxWidth)
                                                                                  if (selectedRowHall != 0 && selectedPlaceHall != 0)
            scale = maxWidth / Hall.ActualWidth;
                                                                                     Hall.Children.Clear();
                                                                                    hidePlace.Clear();
         if (newHeight > maxHeight)
            scale = Math.Min(scale, maxHeight / Hall.ActualHeight);
                                                                                     for (int row = 0; row < selectedRowHall; row++)
                                                                                       WrapPanel wrapPanel = new WrapPanel();
         ScaleTransform scaleTransform = new ScaleTransform(scale,
scale, centerX, centerY);
                                                                                       TextBlock textBlockBegin = new TextBlock();
                                                                                       textBlockBegin.Text = $"Ряд {row + 1}";
         Hall.RenderTransform = scaleTransform;
                                                                                       textBlockBegin.FontSize = 10;
                                                                                       textBlockBegin.Margin = new Thickness(0, 0, 5, 0);
       catch (Exception ex)
                                                                                       textBlockBegin. Vertical Alignment = \\
                                                                         VerticalAlignment.Center;
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                       TextBlock textBlockEnd = new TextBlock();
       }
                                                                                       textBlockEnd.Text = $"Ряд {row + 1}";
     }
                                                                                       textBlockEnd.FontSize = 10;
     private void SelectedRowHall_SelectionChanged(object sender,
                                                                                       textBlockEnd.VerticalAlignment =
SelectionChangedEventArgs e)
                                                                         VerticalAlignment.Center;
       try
                                                                                       wrapPanel.Children.Add(textBlockBegin);
                                                                                       for (int place = 1; place <= selectedPlaceHall; place++)
         selectedRowHall = (int)SelectedRowHall.SelectedItem;
         CreateHall(null);
                                                                                         Button button = new Button();
                                                                                         button.Content = place;
                                                                                         button.Name = $"Row\{row\}Place\{place\}";
       catch (Exception ex)
                                                                                         button.Click += HidePlace_Click;
       {
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                         button. Width = 18;
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                          button.Height = 18;
                                                                                          button.FontSize = 10;
     }
                                                                                          button.Background = Brushes.White;
                                                                                          button.Margin = new Thickness(0, 0, 5, 0);
     private void SelectedPlaceHall_SelectionChanged(object sender,
SelectionChangedEventArgs e)
                                                                                         if (hidePlaceLoadData != null)
     {
                                                                                            foreach (var hidePlaceLine in hidePlaceLoadData)
       try
         selectedPlaceHall = (int)SelectedPlaceHall.SelectedItem;
                                                                                              Match match = Regex.Match(hidePlaceLine,
         CreateHall(null);
                                                                         @"Row(\d+)Place(\d+)");
       catch (Exception ex)
                                                                                              if (match.Success)
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                                if (row + 1 ==
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                         int.Parse(match.Groups[1].Value) + 1 && place ==
                                                                         int.Parse(match.Groups[2].Value))
                                                                                                   button.Tag = button.Name;
    private void CreateHall(string[] hidePlaceLoadData)
                                                                                                   button.Content = "";
```

```
button. Opacity = 0.2;
                                                                                  MessageBox.Show(ex.Message, "Ошибка",
                          hidePlace.Add(button.Tag.ToString());
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                }
                                                                             }
                }
                                                                             private void SaveHallSettings_Click(object sender,
                wrapPanel.Children.Add(button);
                                                                        RoutedEventArgs e)
                                                                             {
              wrapPanel.Children.Add(textBlockEnd);
                                                                                try
              StackPanel stackPanel = new StackPanel();
                                                                                  using (var dataBase = new CinemaEntities())
              stackPanel.Margin = new Thickness(0, 0, 0, 5);
              stackPanel.HorizontalAlignment =
                                                                                     var newHall = new Settings();
HorizontalAlignment.Center;
              stackPanel.Children.Add(wrapPanel);
                                                                                    newHall.RowHall = (int)SelectedRowHall.SelectedItem; \\
                                                                                    newHall.PlaceHall = (int)SelectedPlaceHall.SelectedItem; \\
              Hall.Children.Add(stackPanel);
                                                                                    newHall.DateTimeChange = DateTime.Now; \\
                                                                                    string hidePlaceTemp = null;
         }
                                                                                    foreach (var hidePlaceLine in hidePlace)
       catch (Exception ex)
                                                                                       if (hidePlaceLine != hidePlace.LastOrDefault())
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                         hidePlaceTemp += hidePlaceLine + "|";
     }
                                                                                       else
    private void HidePlace_Click(object sender, RoutedEventArgs e)
                                                                                         hidePlaceTemp += hidePlaceLine;
     {
       try
         Button button = (Button)sender;
                                                                                    newHall.HiddenPlaces = hidePlaceTemp;
         if (button.Content.ToString() != "")
                                                                                    dataBase.Settings.Add(newHall);
                                                                                    dataBase.SaveChanges();
           button.Tag = button.Name;
           button.Content = "";\\
                                                                                    MessageBox.Show("Нстройка зала была сохранена",
           button. Opacity = 0.2;
                                                                         "Готово", MessageBoxButton.OK, MessageBoxImage.Information);
           hidePlace.Add(button.Tag.ToString());
         }
                                                                                }
                                                                                catch (Exception ex)
         else
           button.Name = button.Tag.ToString();
                                                                                  MessageBox.Show(ex.Message, "Ошибка",
           button. Content = int. Parse (Regex. Match (button. Name, \\
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
@ "Row(\d+)Place(\d+)").Groups[2].Value);
           button. Tag = null;
                                                                             }
           button. Opacity = 1;
                                                                           }
           hidePlace.Remove(hidePlace.FirstOrDefault(w => w ==
button.Name.ToString()));
                                                                        TicketAnalysis.xaml
                                                                         <Page x:Class="Cinema.TicketAnalysis"
       }
       catch (Exception ex)
```

```
ScrollViewer.CanContentScroll="False"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                     MouseDoubleClick="TicketList_MouseDoubleClick">
   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                              <ListView.ItemContainerStyle>
   xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                                 <Style TargetType="ListViewItem">
compatibility/2006"
                                                                                   <Setter Property="HorizontalContentAlignment"
   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                     Value="Stretch" />
                                                                                 </Style>
   xmlns:local="clr-namespace:Cinema"
   mc:Ignorable="d"
                                                                              </ListView.ItemContainerStyle>
   d:DesignHeight="450" d:DesignWidth="800"
                                                                              <ListView.View>
   Title="TicketAnalysis" SizeChanged="Page_SizeChanged"
                                                                                 <GridView>
                                                                                   <GridViewColumn x:Name="MovieCodeGrid">
Loaded="Page_Loaded">
                                                                                     <Label Content="Код фильма"
  <Grid Style="{DynamicResource MainColorStyle}">
                                                                     Style="{DynamicResource MainLableStyle}"/>
    <StackPanel>
                                                                                     <GridViewColumn.CellTemplate>
       <Grid Style="{DynamicResource SecondColorStyle}">
                                                                                       <DataTemplate>
         <WrapPanel VerticalAlignment="Center" Margin="10,0,0,0">
                                                                                          <Label Content="{Binding ticketMovieID}"</pre>
           <TextBlock Text="Поиск:" Margin="0,0,10,0"
                                                                     HorizontalAlignment="Center" Style="{DynamicResource
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                     MainLableStyle \}"/>
           <TextBox x:Name="FindData" Width="200"
                                                                                       </DataTemplate>
Style="{DynamicResource MainTextBoxStyle}"
                                                                                     </GridViewColumn.CellTemplate>
TextChanged="FindData_TextChanged"/>
                                                                                   </GridViewColumn>
           < Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                                   <GridViewColumn x:Name="ImageGrid">
                                                                                     <Label Content="Постер" Style="{DynamicResource
SecondColorSeparatorsRectangleStyle \} "/>
           <CheckBox x:Name="FindDateChecker" Margin="0,4,5,0"</pre>
                                                                     MainLableStyle \}"/>
Style="{DynamicResource MainCheckBoxStyle}"
                                                                                     <GridViewColumn.CellTemplate>
Click="FindDateChecker_Click"/>
                                                                                       <DataTemplate>
                                                                                         <Image Source="{Binding ticketMovieCover}"</pre>
           <TextBlock Text="C:" Margin="0,0,10,0"
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                     MaxHeight="400" HorizontalAlignment="Center"/>
           <DatePicker x:Name="BeginDate" Margin="0,0,10,0"</pre>
                                                                                       </DataTemplate>
Style="{DynamicResource SecondDatePickerStyle}"
                                                                                     </GridViewColumn.CellTemplate>
SelectedDateChanged="BeginDate_SelectedDateChanged"/>
                                                                                   </GridViewColumn>
           <TextBlock Text="по" Margin="0,0,10,0"
                                                                                   <GridViewColumn x:Name="MovieInfoGrid">
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                                     <Label Content="Информация о фильме"
           <DatePicker x:Name="EndDate"
                                                                     Style="{DynamicResource MainLableStyle}"/>
Style="{DynamicResource SecondDatePickerStyle}"
                                                                                     <GridViewColumn.CellTemplate>
SelectedDateChanged="EndDate_SelectedDateChanged"/>
                                                                                       <DataTemplate>
           < Rectangle Margin="10,0,10,0" Style="{DynamicResource
                                                                                         <StackPanel>
SecondColorSeparatorsRectangleStyle}"/>
                                                                                            <Label Content="{Binding ticketMovieTitle}"</pre>
           <TextBlock Text="Найдено: " Style="{DynamicResource
                                                                     Style="{DynamicResource MainLableStyle}"/>
MainTextBlockStyle \}"/>
                                                                                            <WrapPanel>
           <TextBlock x:Name="FindCounterData" Text="0/0"
                                                                                              <Label Content="Жанр(ы):"
Style="{DynamicResource MainTextBlockStyle}"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
         </WrapPanel>
                                                                                              < Label Content="{Binding
         <WrapPanel HorizontalAlignment="Right"</p>
                                                                     ticketMovieGenre}" Style="{DynamicResource MainLableStyle}"/>
VerticalAlignment="Center" Margin="0,0,10,0">
                                                                                            </WrapPanel>
           <Button x:Name="Print" Content="Печать отчета"
                                                                                            <WrapPanel>
Width="120" Style="{DynamicResource TabControlButtonStyle}"
                                                                                              <Label Content="Год публикации:"
Click="Print_Click"/>
                                                                     Style="{DynamicResource MainLableStyle}"/>
         </WrapPanel>
                                                                                              < Label Content="{Binding
       </Grid>
                                                                     ticketMovieYearOfPublication}" Style="{DynamicResource
                                                                     MainLableStyle \}"/>
       <ListView x:Name="TicketList" Height="400"</p>
                                                                                            </WrapPanel>
```

<WrapPanel>

d:ItemsSource="{d:SampleData ItemCount=2}"

```
<Label Content="Хронометраж:"
                                                                                 </ListView.View>
Style="{DynamicResource MainLableStyle}"/>
                                                                              </ListView>
                         <Label Content="{Binding
                                                                            </StackPanel>
ticketMovieTiming}" Style="{DynamicResource MainLableStyle}"/>
                                                                          </Grid>
                       </WrapPanel>
                                                                       </Page>
                       <WrapPanel>
                         <Label Content="Возрастной рейтинг:"
                                                                       TicketAnalysis.xaml.cs
Style="{DynamicResource MainLableStyle}"/>
                                                                       using iTextSharp.text;
                         <Label Content="{Binding
                                                                       using iTextSharp.text.pdf;
ticketMovieAgeRating}" Style="{DynamicResource
                                                                       using Microsoft.Win32;
MainLableStyle \} "/>
                                                                       using System;
                       </WrapPanel>
                                                                       using System.Collections.Generic;
                       <WrapPanel>
                                                                       using System.Diagnostics;
                         <Label Content="Страна:"
                                                                       using System.IO;
Style="{DynamicResource MainLableStyle}"/>
                                                                       using System.Linq;
                         < Label Content="{Binding
                                                                       using System.Windows;
ticketMovieCountry}" Style="{DynamicResource MainLableStyle}"/>
                                                                       using System.Windows.Controls;
                       </WrapPanel>
                                                                       using System.Windows.Input;
                       <WrapPanel>
                                                                       using System.Windows.Media.Imaging;
                         <Label Content="Актеры:"
                                                                       using static Cinema. Authorization;
Style="{DynamicResource MainLableStyle}"/>
                         <TextBlock Text="{Binding
                                                                       namespace Cinema
ticketMovieActors}" Margin="5,0,0,0" VerticalAlignment="Center"
TextWrapping="Wrap" Style="{DynamicResource
                                                                         /// <summary>
MainTextBlockStyle \}"/>
                                                                         /// Логика взаимодействия для TicketAnalysis.xaml
                       </WrapPanel>
                                                                         /// </summary>
                     </StackPanel>
                                                                          public partial class TicketAnalysis: Page
                  </DataTemplate>
                </GridViewColumn.CellTemplate>
                                                                            public TicketAnalysis()
              </GridViewColumn>
              <GridViewColumn x:Name="AnalyticsGrid">
                                                                              InitializeComponent();
                <Label Content="Аналитика продаж"
                                                                            }
Style="{DynamicResource MainLableStyle}"/>
                <GridViewColumn.CellTemplate>
                                                                            public class TicketData
                  <DataTemplate>
                                                                            {
                     <StackPanel>
                                                                              public int ticketMovieID { get; set; }
                       <WrapPanel>
                                                                              public BitmapImage ticketMovieCover { get; set; }
                         <Label Content="Количество проданных
                                                                              public string ticketMovieTitle { get; set; }
билетов:" Style="{DynamicResource MainLableStyle}"/>
                                                                              public string ticketMovieGenre { get; set; }
                         <Label Content="{Binding ticketCount}"</pre>
                                                                              public int ticketMovieYearOfPublication { get; set; }
Style="{DynamicResource MainLableStyle}"/>
                                                                              public double ticketMovieTiming { get; set; }
                       </WrapPanel>
                                                                              public string ticketMovieAgeRating { get; set; }
                       <WrapPanel>
                                                                              public string ticketMovieCountry { get; set; }
                         <Label Content="Суммарные сборы:"
                                                                              public string ticketMovieActors { get; set; }
Style="{DynamicResource MainLableStyle}"/>
                                                                              public string ticketCount { get; set; }
                         < Label Content="{Binding
                                                                              public string ticketSummaryCost { get; set; }
ticketSummaryCost}" Style="{DynamicResource MainLableStyle}"/>
                       </WrapPanel>
                     </StackPanel>
                                                                            List<TicketData> ticketDataList = new List<TicketData>();
                  </DataTemplate>
                </GridViewColumn.CellTemplate>
                                                                            private void Page_Loaded(object sender, RoutedEventArgs e)
              </GridViewColumn>
            </GridView>
                                                                              try
```

```
where ((findLine == null ||
         TicketList.ItemsSource = ticketDataList;
                                                                          movie.Title.Contains(findLine) || genre.Title.Contains(findLine) ||
                                                                          movie. Year Of Publication. To String (). Contains (find Line) \parallel
         LoadData(null, BeginDate, EndDate);
                                                                          movie. Description. Contains (find Line) \parallel
                                                                          country.Title.Contains(findLine)) && ((beginDate.SelectedDate == null
       catch (Exception ex)
                                                                          \parallel session. Date And Time Session > begin Date. Selected Date. Value) \ \&\& \\
                                                                          (endDate.SelectedDate == null || session.DateAndTimeSession <
         MessageBox.Show(ex.Message, "Ошибка",
                                                                          endDate.SelectedDate.Value)))
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                                 select new
     }
                                                                                                   movie.IDMovie,
                                                                                                   movie.Cover.
    private void LoadData(string findLine, DatePicker beginDate,
                                                                                                   movieTitle = movie.Title,
DatePicker endDate)
                                                                                                   movieGenre = genre. Title,
                                                                                                   movie.YearOfPublication,
       try
                                                                                                   movie.Timing,
                                                                                                   movie.AgeRating,
         using (var dataBase = new CinemaEntities())
                                                                                                   movieCountry = country.Title,
                                                                                                   movieActors = actors.Surname + " " +
            ticketDataList.Clear();
                                                                          actors.Name + " " + actors.Patronymic + " " + actors.Nickname,
                                                                                                   session.TicketPrice
            var ticketData = (from ticket in dataBase.Ticket
                                                                                                ).ToList();
                       session in dataBase.Session on ticket.IDSession
equals session.IDSession into SessionGroup
                                                                                      var groupedMovieData = ticketData.GroupBy(m =>
                      from session in SessionGroup.DefaultIfEmpty()
                                                                          m.IDMovie)
                      join
                                                                                         .Select(g => new
                      movie in dataBase.Movie on session.IDMovie
equals movie.IDMovie into movieGroup
                                                                                           g.Key,
                      from movie in movieGroup.DefaultIfEmpty()
                                                                                           genres = g.Select(m => m.movieGenre).Distinct(),
                                                                                           actors = g.Select(m => m.movieActors).Distinct(),
                      actorsInMovies in dataBase.ActorsInMovies on
                                                                                           TicketCount = g.Count(),
movie.IDMovie equals actorsInMovies.IDMovie into
                                                                                           TotalCost = g.Sum(t => t.TicketPrice)
actorsInMoviesGroup
                                                                                         }).ToList();
                       from actorsInMovie in
actorsInMoviesGroup.DefaultIfEmpty()
                                                                                      var result = groupedMovieData.Select(g =>
                       actors in dataBase.Actor on
                                                                                         var movie = ticketData.FirstOrDefault(m => m.IDMovie
actorsInMovie.IDActor equals actors.IDActor into actorsGroup
                                                                          == g.Key);
                       from actors in actorsGroup.DefaultIfEmpty()
                                                                                         return new
                      movieGenre in dataBase.MovieGenre on
                                                                                           movie.IDMovie,
movie.IDMovie equals movieGenre.IDMovie into movieGenereGroup
                                                                                           movie.Cover,
                      from movieGenere in
                                                                                           movie.movieTitle,
movieGenereGroup.DefaultIfEmpty()
                                                                                           movieGenres = g.genres,
                      join
                                                                                           movie.YearOfPublication,
                      genre in dataBase.Genre on
                                                                                           movie.Timing,
movieGenere.IDGenre equals genre.IDGenre into genreGroup
                                                                                           movie.AgeRating,
                       from genre in genreGroup.DefaultIfEmpty()
                                                                                           movie.movieCountry,
                                                                                           movieActors = g.actors,
                      country in dataBase.Country on
                                                                                           g.TicketCount,
movie.IDCountry equals country.IDCountry into countryGroup
                                                                                           g.TotalCost,
                       from country in countryGroup.DefaultIfEmpty()
                                                                                           movie.TicketPrice,
```

```
};
                                                                                         else
            }).ToList();
                                                                                            movieActorTemp += actor;
           foreach (var ticketLine in result)
              TicketData ticketDataClass = new TicketData();
                                                                                         movieActorCounterTemp++;
              ticketDataClass.ticketMovieID = ticketLine.IDMovie;
                                                                                       ticketDataClass.ticketMovieActors = movieActorTemp;
              if (ticketLine.Cover != null)
                                                                                       ticketDataClass.ticketMovieYearOfPublication =
                                                                         ticketLine.YearOfPublication;
              {
                BitmapImage cover = new BitmapImage();
                                                                                       ticketDataClass.ticketMovieTiming = ticketLine.Timing;
                                                                                       ticketDataClass.ticketMovieAgeRating =
                                                                         ticketLine.AgeRating + "+";
                cover.BeginInit();
                cover.StreamSource = new \\
                                                                                       ticketDataClass.ticketMovieCountry = \\
MemoryStream(ticketLine.Cover);
                                                                         ticketLine.movieCountry;
                cover.EndInit();
                                                                                       ticketDataClass.ticketCount = (ticketLine.TicketCount /
                ticketDataClass.ticketMovieCover = cover;
                                                                         ((movieGenresCounterTemp - 1) * (movieActorCounterTemp -
              }
                                                                         1))).ToString();
              else
                                                                                       ticketDataClass.ticketSummaryCost =
              {
                                                                         (ticketLine.TotalCost / ((movieGenresCounterTemp - 1) *
                ticketDataClass.ticketMovieCover = new
                                                                         (movieActorCounterTemp -
BitmapImage(new Uri("pack://application:,,,/Resource/NoImage.png",
                                                                         1))).ToString().Remove((ticketLine.TotalCost /
UriKind.Absolute));
                                                                         ((movieGenresCounterTemp - 1) * (movieActorCounterTemp -
                                                                         1))).ToString().Length - 2, 2) + " py6.";
              ticketDataClass.ticketMovieTitle =
                                                                                       ticketDataList.Add(ticketDataClass);
ticketLine.movieTitle;
                                                                                     }
                                                                                     var fullMovieData = (from ticket in dataBase.Ticket
              string movieGenereTemp = "";
              int movieGenresCounterTemp = 1;
                                                                                                 ioin
              foreach (var genere in ticketLine.movieGenres)
                                                                                                 session in dataBase.Session on
                                                                         ticket.IDSession equals session.IDSession into sessionGroup
                if (movieGenresCounterTemp !=
                                                                                                 from session in
ticketLine.movieGenres.Count())
                                                                         sessionGroup.DefaultIfEmpty()
                   movieGenereTemp += genere + ", ";
                else
                                                                                                 movie in dataBase.Movie on session.IDMovie
                   movieGenereTemp += genere;
                                                                         equals movie.IDMovie into movieGroup
                                                                                                 from movie in movieGroup.DefaultIfEmpty()
                                                                                                 group ticket by movie into movieTickets
                movieGenresCounterTemp++;
                                                                                                 where movieTickets.Count() > 1
              ticketDataClass.ticketMovieGenre = movieGenereTemp;
                                                                                                 select new
              string movieActorTemp = "";
                                                                                                   Movie = movieTickets.Key,
              int movieActorCounterTemp = 1;
                                                                                                    TicketCount = movieTickets.Count()
              foreach (var actor in ticketLine.movieActors)
                                                                                                 }).ToList();
                if (movieActorCounterTemp !=
                                                                                     FindCounterData.Text = result.Count() + "/" +
ticketLine.movieActors.Count())
                                                                         fullMovieData.Count();
                   movieActorTemp += actor + ", ";
                                                                                     TicketList.Items.Refresh();
```

```
BeginDate.SelectedDate = DateTime.Now;
       catch (Exception ex)
                                                                                  EndDate.IsEnabled = true;
                                                                                  EndDate.SelectedDate = DateTime.Now;
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                LoadData(FindData.Text, BeginDate, EndDate);
     }
                                                                              catch (Exception ex)
    private void Page_SizeChanged(object sender,
SizeChangedEventArgs e)
                                                                                MessageBox.Show(ex.Message, "Ошибка",
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
    {
       try
                                                                              }
                                                                            }
         MovieCodeGrid.Width = 0;
         ImageGrid.Width = ActualWidth - ActualWidth * 0.8;
                                                                            private void FindData_TextChanged(object sender,
         MovieInfoGrid.Width = ActualWidth - ActualWidth * 0.6;
                                                                       TextChangedEventArgs e)
         AnalyticsGrid.Width = ActualWidth - ActualWidth * 0.6;
                                                                            {
         TicketList.Height = ActualHeight - 35;
                                                                              LoadData(FindData.Text, BeginDate, EndDate);
                                                                            }
       catch (Exception ex)
                                                                           private void TicketList_MouseDoubleClick(object sender,
         MessageBox.Show(ex.Message, "Ошибка",
                                                                       MouseButtonEventArgs e)
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                            {
                                                                              try
                                                                                var selectedSession = TicketList.SelectedItem as TicketData;
    private void BeginDate_SelectedDateChanged(object sender,
                                                                                if (selectedSession != null)
SelectionChangedEventArgs e)
                                                                                  Transmitted Data. id Selected Movie Analysis = \\
       LoadData(FindData.Text,\,BeginDate,\,EndDate);\\
                                                                       selectedSession.ticketMovieID;
                                                                                   Window activeWindow = Window.GetWindow(this);
    private void EndDate_SelectedDateChanged(object sender,
SelectionChangedEventArgs e)
                                                                                  if (activeWindow.Title == "Панель директора")
     {
       LoadData(FindData.Text, BeginDate, EndDate);
                                                                                     DirectorsPanel directorsPanel =
                                                                       Window.GetWindow(this) as DirectorsPanel;
                                                                                     directorsPanel.MovieAnalysisOpen();
    private void FindDateChecker_Click(object sender,
RoutedEventArgs e)
                                                                                  if (activeWindow.Title == "Панель менеджера")
     {
       try
                                                                                     BookerPanel = Window.GetWindow(this)
         if (FindDateChecker.IsChecked == false)
                                                                       as BookerPanel;
                                                                                     bookerPanel.MovieAnalysisOpen();
           BeginDate.IsEnabled = false;
           BeginDate.SelectedDate = null;
           EndDate.IsEnabled = false;
           EndDate.SelectedDate = null;
                                                                              catch (Exception ex)
         }
                                                                                MessageBox.Show(ex.Message, "Ошибка",
         else
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
           BeginDate.IsEnabled = true;
```

```
table.AddCell(new Paragraph("Количество проданных
     }
                                                                         билетов", font));
    private void Print_Click(object sender, RoutedEventArgs e)
                                                                                       table.AddCell(new Paragraph("Выручка с проданных
                                                                         билетов", font));
       try
                                                                                       foreach (var line in ticketDataList)
         using (var dataBase = new CinemaEntities())
                                                                                         table.AddCell(new
           string findLine = FindData.Text;
                                                                         Paragraph(line.ticketMovieID.ToString(), font));
                                                                                         table.AddCell(new Paragraph(line.ticketMovieTitle,
           SaveFileDialog saveFileDialog = new SaveFileDialog();
                                                                         font));
           saveFileDialog.Filter = "PDF files (*.pdf)|*.pdf";
                                                                                         table.AddCell(new Paragraph(line.ticketMovieGenre,
                                                                         font));
           if (saveFileDialog.ShowDialog() == true)
                                                                                         table.AddCell(new
                                                                         Paragraph(line.ticketMovieYearOfPublication.ToString(), font));
              Document doc = new Document();
                                                                                         table.AddCell(new
              PdfWriter writer = PdfWriter.GetInstance(doc, new
                                                                        Paragraph(line.ticketMovieTiming.ToString(), font));
FileStream (\$@"\{saveFileDialog.FileName\}", FileMode.Create));\\
                                                                                         table.AddCell(new
              doc. SetPageSize(PageSize.A4.Rotate());\\
                                                                        Paragraph (line.ticket Movie Age Rating, \ font)); \\
                                                                                         table.AddCell(new
              BaseFont baseFont =
                                                                        Paragraph(line.ticketMovieCountry, font));
BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf",
                                                                                         table.AddCell(new Paragraph(line.ticketMovieActors,
BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
                                                                         font));
              Font font = new Font(baseFont, 12);
                                                                                         table.AddCell(new Paragraph(line.ticketCount, font));
                                                                                         table.AddCell(new Paragraph(line.ticketSummaryCost,
              BaseFont baseFontHead =
                                                                         font));
BaseFont.CreateFont("C:\\Windows\\Fonts\\arial.ttf",
BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
                                                                                       doc.Add(table);
              Font fontHead = new Font(baseFont, 20, Font.BOLD);
                                                                                       double ticketPriceSumm = 0;
              doc.Open();
                                                                                       foreach (var line in ticketDataList)
                                                                                       {
              Paragraph mainParagraph = new Paragraph("Отчет
                                                                                         ticketPriceSumm +=
продаж", fontHead);
                                                                         Convert.ToDouble(line.ticketSummaryCost.Replace("py6.", ""));
              mainParagraph.Alignment = Element.ALIGN_CENTER;
                                                                                       }
              doc.Add(mainParagraph);
                                                                                       Paragraph paragraph = new Paragraph("Итог: " +
              PdfPTable table = new PdfPTable(10);
                                                                         ticketPriceSumm.ToString() + " Py6.", font);
              table.SpacingBefore = 10;
                                                                                      doc.Add(paragraph);
              table.WidthPercentage = 100;
                                                                                       doc.Close();
              table.AddCell(new Paragraph("Код фильма", font));
                                                                                       MessageBox.Show("Файл сохранен", "Готово",
              table.AddCell(new Paragraph("Название фильма",
                                                                        MessageBoxButton.OK, MessageBoxImage.Information);
font));
              table.AddCell(new Paragraph("Жанры", font));
                                                                                       Process.Start(saveFileDialog.FileName);
              table.AddCell(new Paragraph("Год публикации", font));
                                                                                    }
              table.AddCell(new Paragraph("Длительность фильма",
                                                                                  }
font));
              table.AddCell(new Paragraph("Возрастной рейтинг",
                                                                                catch (Exception ex)
font));
              table.AddCell(new Paragraph("Страна", font));
                                                                                  MessageBox.Show(ex.Message, "Ошибка",
              table.AddCell(new Paragraph("Актеры", font));
                                                                         MessageBoxButton.OK, MessageBoxImage.Error);
```

```
</Grid>
                                                                                 <Grid Margin="0,0,0,10">
  }
                                                                                   <StackPanel>
                                                                                     <Grid>
AddOrEditActor.xaml
                                                                                        <Label Content="Фотография:"
<Window x:Class="Cinema.AddOrEditActor"
                                                                     Style="{DynamicResource MainLableStyle}"/>
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                                        <Image x:Name="ActorPhoto"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                     HorizontalAlignment="Left" Margin="170,0,0,5" Width="200"
                                                                     Height="200" Visibility="Collapsed"/>
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                                     </Grid>
compatibility/2006"
                                                                                     <Button x:Name="LoadPhoto"
    xmlns:local="clr-namespace:Cinema"
                                                                      HorizontalAlignment="Left" Margin="170,0,0,0" Content="Загрузить"
    mc:Ignorable="d"
                                                                      Width="200" Style="{DynamicResource MainButtonStyle}"
    Title="Актер" Height="370" Width="500"
                                                                      Click="LoadPhoto_Click"/>
Loaded="Window_Loaded" ResizeMode="NoResize"
                                                                                   </StackPanel>
                                                                                 </Grid>
WindowStartupLocation="CenterScreen">
  <Grid Style="{DynamicResource MainColorStyle}">
                                                                               </StackPanel>
                                                                             </Grid>
    <StackPanel>
       <Grid>
                                                                             <Button x:Name="Save" Margin="0,0,0,0"
         <Rectangle Style="{DynamicResource
                                                                      Content="Coxpaнить" Width="450" Style="{DynamicResource
SecondColorRectangleStyle}"/>
                                                                      MainButtonStyle \} " Click="Save_Click"/>
         <TextBlock Text="AKTEP" Style="{DynamicResource
                                                                          </StackPanel>
TitleTextBlockStyle \} "/>
                                                                        </Grid>
      </Grid>
                                                                      </Window>
       <Grid HorizontalAlignment="Center" Margin="10">
                                                                      AddOrEditActor.xaml.cs
         <StackPanel>
           <Grid Margin="0,0,0,10">
                                                                      using Microsoft.Win32;
             <Label Content="Фамилия:" Style="{DynamicResource
                                                                      using System;
MainLableStyle \}"/>
                                                                      using System.IO;
             <TextBox x:Name="ActorSurname"
                                                                      using System.Linq;
HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
                                                                      using System.Windows;
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                      using System.Windows.Media.Imaging;
           </Grid>
                                                                      using static Cinema. Authorization;
           <Grid Margin="0,0,0,10">
             <Label Content="Имя:" Style="{DynamicResource
                                                                     namespace Cinema
MainLableStyle \}"/>
                                                                      {
             <TextBox x:Name="ActorName"
                                                                        /// <summary>
HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
                                                                        /// Логика взаимодействия для AddOrEditActor.xaml
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                        /// </summary>
           </Grid>
                                                                        public partial class AddOrEditActor: Window
           <Grid Margin="0,0,0,10">
             <Label Content="Отчество:" Style="{DynamicResource
                                                                          public AddOrEditActor()
MainLableStyle \}"/>
                <TextBox x:Name="ActorPatronymic"
                                                                            InitializeComponent();
HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
                                                                          }
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                          public string actorPhotoPath;
           <Grid Margin="0,0,0,10">
             <Label Content="Прозвище:"
                                                                          private void Window_Loaded(object sender, RoutedEventArgs e)
Style="{DynamicResource MainLableStyle}"/>
                  <TextBox x:Name="ActorNickname"
                                                                            ActorPhoto. Visibility = Visibility. Collapsed;
HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
                                                                            LoadData();
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                          }
```

```
ActorPhoto.Source = new BitmapImage(new
    private void LoadData()
                                                                        Uri(openFileDialog.FileName));
      try
                                                                               catch (Exception ex)
         using (var dataBase = new CinemaEntities())
                                                                               {
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
           if (TransmittedData.idSelectedActor != -1)
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
                                                                            }
              var actorData = dataBase.Actor.Where(w => w.IDActor
== TransmittedData.idSelectedActor).FirstOrDefault();
                                                                            private void Save_Click(object sender, RoutedEventArgs e)
              ActorSurname.Text = actorData.Surname;
              ActorName.Text = actorData.Name;
                                                                               try
              ActorPatronymic.Text = actorData.Patronymic;
                                                                                 if (ActorSurname.Text == "" && ActorName.Text == "")
              ActorNickname.Text = actorData.Nickname;
                                                                                   MessageBox.Show("Данные не были заполнены",
              if (actorData.Image != null)
                                                                        "Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);
              {
                BitmapImage bitmapImage = new BitmapImage();
                                                                                   return:
                bitmapImage.BeginInit();
                bitmapImage.StreamSource = new
                                                                                 using (var dataBase = new CinemaEntities())
MemoryStream(actorData.Image);
                bitmapImage.EndInit();
                                                                                   if (TransmittedData.idSelectedActor != -1)
                this.Height = 535;
                ActorPhoto.Visibility = Visibility.Visible;
                                                                                      var actorData = dataBase.Actor.Where(w => w.IDActor
                ActorPhoto.Source = bitmapImage;
                                                                        == TransmittedData.idSelectedActor).FirstOrDefault();
                                                                                      actorData.Surname = ActorSurname.Text;
           }
         }
                                                                                      actorData.Name = ActorName.Text;
                                                                                      actorData.Patronymic = ActorPatronymic.Text;
       catch (Exception ex)
                                                                                      actorData.Nickname = ActorNickname.Text;
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                      if (actorPhotoPath != null)
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                        actorData.Image = File.ReadAllBytes(actorPhotoPath);
    private void LoadPhoto_Click(object sender, RoutedEventArgs e)
                                                                                     dataBase.SaveChanges();
                                                                                   }
                                                                                   else
      try
         OpenFileDialog openFileDialog = new OpenFileDialog();
                                                                                      var newActorData = new Actor();
         openFileDialog.Filter = "PNG (*.png)|*.png|JPG
(*.jpg)|*.jpg";
                                                                                      newActorData.Surname = ActorSurname.Text;
                                                                                      newActorData.Name = ActorName.Text;
         if (openFileDialog.ShowDialog() == true)
                                                                                      newActorData.Patronymic = ActorPatronymic.Text;
                                                                                      newActorData.Nickname = ActorNickname.Text;
           actor Photo Path = open File Dialog. File Name; \\
                                                                                      if (actorPhotoPath != null)
           this. Height = 535;
           ActorPhoto.Visibility = Visibility.Visible;
```

```
newActorData.Image =
                                                                                  <TextBox x:Name="EmployeeName"
File.ReadAllBytes(actorPhotoPath);
                                                                    HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
                                                                    Style="{DynamicResource MainTextBoxStyle}"/>
                                                                                </Grid>
                                                                                <Grid Margin="0,0,0,10">
             dataBase.Actor.Add(newActorData);
             dataBase.SaveChanges();
                                                                                  <Label Content="Отчество:" Style="{DynamicResource
                                                                     MainLableStyle \}"/>
                                                                                  <TextBox x:Name="EmployeePatronymic"
                                                                    HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
           this.Close();
                                                                     Style="{DynamicResource MainTextBoxStyle}"/>
         }
                                                                                </Grid>
       }
       catch (Exception ex)
                                                                                <Grid Margin="0,0,0,10">
                                                                                  <Label Content="Телефон:" Style="{DynamicResource
         MessageBox.Show(ex.Message, "Ошибка",
                                                                     MainLableStyle \}"/>
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                  <TextBox x:Name="EmployeePhone"
                                                                    HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
                                                                     Style="{DynamicResource MainTextBoxStyle}"/>
    }
                                                                                </Grid>
  }
                                                                                <Grid Margin="0,0,0,10">
}
                                                                                  <Label Content="Эл. Почта:"
AddOrEditEmployee.xaml
                                                                     Style="{DynamicResource MainLableStyle}"/>
<Window x:Class="Cinema.AddOrEditEmployee"
                                                                                  <TextBox x:Name="EmployeeEmail"
                                                                    HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                     Style="{DynamicResource MainTextBoxStyle}"/>
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                                </Grid>
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                                <Grid Margin="0,0,0,10">
compatibility/2006"
                                                                                  <Label Content="Должность:"
    xmlns:local="clr-namespace:Cinema"
                                                                     Style="{DynamicResource MainLableStyle}"/>
    mc:Ignorable="d"
                                                                                  <ComboBox x:Name="EmployeeRole"
    Title="Сотрудник" Height="530" Width="500"
                                                                    Margin="100,0,0,0" Width="350" Style="{DynamicResource
Loaded="Window_Loaded" ResizeMode="NoResize"
                                                                     MainComboBoxStyle}" ItemContainerStyle="{DynamicResource
WindowStartupLocation="CenterScreen">
                                                                     MainSelectedComboBoxStyle}"/>
                                                                                </Grid>
  <Grid Style="{DynamicResource MainColorStyle}">
                                                                                <Grid Margin="0,0,0,10">
    <StackPanel>
                                                                                  <Label Content="Логин:" Style="{DynamicResource
       <Grid>
                                                                     MainLableStyle \}"/>
         <Rectangle Style="{DynamicResource
                                                                                  <TextBox x:Name="EmployeeLoggin"
SecondColorRectangleStyle}"/>
                                                                    HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
         <TextBlock Text="СОТРУДНИК"
                                                                     Style="{DynamicResource MainTextBoxStyle}"/>
Style="{DynamicResource TitleTextBlockStyle}"/>
                                                                                </Grid>
       </Grid>
                                                                                <Grid Margin="0,0,0,10">
       <Grid HorizontalAlignment="Center" Margin="10">
                                                                                  <Label Content="Пароль:" Style="{DynamicResource
         <StackPanel>
                                                                    MainLableStyle \}"/>
           <Grid Margin="0,0,0,10">
                                                                                  <TextBox x:Name="EmployeePassword"
             <Label Content="Фамилия:" Style="{DynamicResource
                                                                     HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
MainLableStyle \}"/>
                                                                     Style="{DynamicResource MainTextBoxStyle}"/>
             <TextBox x:Name="EmployeeSurname"
HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
                                                                                <Grid Margin="0,0,0,10">
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                                  <StackPanel>
           </Grid>
                                                                                    <Grid>
           <Grid Margin="0,0,0,10">
                                                                                       <Label Content="Фотография:"
             <Label Content="Имя:" Style="{DynamicResource
                                                                    Style="{DynamicResource MainLableStyle}"/>
MainLableStyle \} "/>
```

```
<Image x:Name="EmployeePhoto"
HorizontalAlignment="Left" Margin="170,0,0,5" Width="200"
                                                                                  var roleData = dataBase.Role.Select(s => s.Title).ToList();
Height="200" Visibility="Collapsed"/>
                                                                                  EmployeeRole.ItemsSource = roleData;
                </Grid>
                                                                                  if (TransmittedData.idSelectedEmployee != -1)
                <Button x:Name="LoadPhoto"
HorizontalAlignment="Left" Margin="170,0,0,0" Content="Загрузить"
                                                                                  {
Width="200" Style="{DynamicResource MainButtonStyle}"
Click="LoadPhoto_Click"/>
                                                                                     var employeeData = (from employee in
              </StackPanel>
                                                                       dataBase.Employee
           </Grid>
                                                                                               join
         </StackPanel>
                                                                                                role in dataBase.Role on employee.IDRole
       </Grid>
                                                                       equals role.IDRole into roleGroup
       <Button x:Name="Save" Margin="0,0,0,0"
                                                                                                from role in roleGroup.DefaultIfEmpty()
                                                                                                where (employee.IDEmployee ==
Content="Coxpaнить" Width="450" Style="{DynamicResource
MainButtonStyle}" Click="Save_Click"/>
                                                                       TransmittedData.idSelectedEmployee)
     </StackPanel>
                                                                                                select new
  </Grid>
</Window>
                                                                                                  employee.IDEmployee,
                                                                                                  employee.Photo,
AddOrEditEmployee.xaml.cs
                                                                                                  employee.Surname,
using Microsoft.Win32;
                                                                                                  employee.Name,
using System;
                                                                                                  employee.Patronymic,
using System.IO;
                                                                                                  employee.Phone,
using System.Ling;
                                                                                                  employee.Email,
using System.Text.RegularExpressions;
                                                                                                  employee.Login,
using System. Windows;
                                                                                                  employee.Password,
                                                                                                  employeeRole = role.Title,
using System.Windows.Media.Imaging;
using static Cinema. Authorization;
                                                                                                }).FirstOrDefault();
namespace Cinema
                                                                                     EmployeeSurname.Text = employeeData.Surname;
{
                                                                                     EmployeeName.Text = employeeData.Name;
                                                                                     EmployeePatronymic.Text = employeeData.Patronymic;
  /// <summary>
  /// Логика взаимодействия для AddOrEditEmployee.xaml
                                                                                     EmployeePhone.Text = employeeData.Phone;
  /// </summary>
                                                                                     EmployeeEmail.Text = employeeData.Email;
  public partial class AddOrEditEmployee: Window
                                                                                     EmployeeRole.SelectedItem = \\
                                                                       employeeData.employeeRole;
    public AddOrEditEmployee()
                                                                                     EmployeeLoggin.Text = employeeData.Login;
                                                                                     Employee Password. Text = employee Data. Password; \\
       InitializeComponent();
                                                                                     if (employeeData.Photo != null)
     }
                                                                                       this. Height = 695;
    public string employeePhotoPath;
                                                                                       EmployeePhoto.Visibility = Visibility.Visible;
     private void Window_Loaded(object sender, RoutedEventArgs e)
                                                                                       BitmapImage bitmapImage = new BitmapImage();
     {
       LoadData();
                                                                                       bitmapImage.BeginInit();
                                                                                       bitmapImage.StreamSource = new
                                                                       MemoryStream(employeeData.Photo);
     private void LoadData()
                                                                                       bitmapImage.EndInit();
                                                                                       EmployeePhoto.Source = bitmapImage;
       try
         using (var dataBase = new CinemaEntities())
                                                                                  }
```

```
if (!Regex.IsMatch(EmployeePhone.Text, phonePattern))
         }
       catch (Exception ex)
                                                                                  MessageBox.Show("Номер телефона заполняеться в
                                                                       формате +7(999)999-99-99", "Внимание", MessageBoxButton.OK,
         MessageBox.Show(ex.Message, "Ошибка",
                                                                      Message Box Image. Warning);\\
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                  return:
       }
    }
                                                                                if (!Regex.IsMatch(EmployeeEmail.Text, mailPattern) &&
    private void LoadPhoto_Click(object sender, RoutedEventArgs e)
                                                                       !Regex.IsMatch(EmployeeEmail.Text, gmailPattern) &&
                                                                       !Regex.IsMatch(EmployeeEmail.Text, yandexMailPattern))
      try
                                                                                  MessageBox.Show("Почта должна содержать доменный
       {
                                                                       адрес (@mail.ru / @gmail.com / @yandex.ru)", "Внимание",
         OpenFileDialog openFileDialog = new OpenFileDialog();
        openFileDialog.Filter = "PNG (*.png)|*.png|JPG
                                                                       MessageBoxButton.OK, MessageBoxImage.Warning);
(*.jpg)|*.jpg";
                                                                                  return;
                                                                                }
         if (openFileDialog.ShowDialog() == true)
                                                                                using (var dataBase = new CinemaEntities())
         {
           employeePhotoPath = openFileDialog.FileName;
                                                                                  if (TransmittedData.idSelectedEmployee != -1)
           this. Height = 695;
           EmployeePhoto.Visibility = Visibility.Visible;
                                                                                    var employeeData = dataBase.Employee.Where(w =>
           EmployeePhoto.Source = new BitmapImage(new
                                                                       w.IDEmployee ==
Uri(openFileDialog.FileName));
                                                                      TransmittedData.idSelectedEmployee).FirstOrDefault();
         }
       }
                                                                                    employeeData.Surname = EmployeeSurname.Text;
       catch (Exception ex)
                                                                                    employeeData.Name = EmployeeName.Text;
                                                                                    employeeData.Patronymic = EmployeePatronymic.Text;
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                    employeeData.Phone = EmployeePhone.Text;
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                    employeeData.Email = EmployeeEmail.Text;
    }
                                                                                    var roleData = dataBase.Role.Where(w => w.Title ==
                                                                      EmployeeRole.SelectedItem.ToString()).FirstOrDefault();
    private void Save_Click(object sender, RoutedEventArgs e)
                                                                                    employeeData.IDRole = roleData.IDRole;
                                                                                    employeeData.Login = EmployeeLoggin.Text;
      try
                                                                                    employeeData. Password = EmployeePassword. Text; \\
         string phonePattern = @"^\+7\(\d{3}\)\d{3}-\d{2}-\d{2}$";
         string mailPattern = @ "^[a-zA-Z0-9._%+-]+@mail\.ru$";
                                                                                    if (employeePhotoPath != null)
         string gmailPattern = @"^[a-zA-Z0-9._\%+-]+@gmail\.com$";
         string yandexMailPattern = @"^[a-zA-Z0-9._%+-
                                                                                      employee Data. Photo = \\
]+@yandex\.ru$";
                                                                      File. Read All Bytes (employee Photo Path);\\
         if (EmployeeSurname.Text == "" && EmployeeName.Text
== "" || EmployeePatronymic.Text == "" || EmployeeLoggin.Text == ""
                                                                                    dataBase.SaveChanges();
|| EmployeePassword.Text == "" || EmployeeRole.SelectedIndex < 0)
                                                                                  }
                                                                                  else
           MessageBox.Show("Данные не были заполнены",
                                                                                  {
"Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);
                                                                                    var newEmployeeData = new Employee();
           return;
                                                                                    newEmployeeData.Surname = EmployeeSurname.Text; \\
         }
                                                                                    newEmployeeData.Name = EmployeeName.Text;\\
```

```
newEmployeeData.Patronymic =
                                                                               <Rectangle Style="{DynamicResource
EmployeePatronymic.Text;
                                                                      SecondColorRectangleStyle}"/>
             newEmployeeData.Phone = EmployeePhone.Text;\\
                                                                               <TextBlock Text="ЖАНР" Style="{DynamicResource
             newEmployeeData.Email = EmployeeEmail.Text;
                                                                      TitleTextBlockStyle}"/>
                                                                             </Grid>
             var roleData = dataBase.Role.Where(w => w.Title ==
                                                                             <Grid HorizontalAlignment="Center" Margin="10">
EmployeeRole.SelectedItem.ToString()).FirstOrDefault();
                                                                               <StackPanel>
                                                                                 <Grid Margin="0,0,0,10">
                                                                                   <Label Content="Название:" Style="{DynamicResource
             newEmployeeData.IDRole = roleData.IDRole;
             newEmployeeData.Login = EmployeeLoggin.Text;
                                                                      MainLableStyle \}"/>
             newEmployeeData.Password = EmployeePassword.Text;
                                                                                   <TextBox x:Name="GenreTitle"
                                                                      HorizontalAlignment="Left" Margin="100,0,0,0" Width="350"
             if (employeePhotoPath != null)
                                                                      Style="{DynamicResource MainTextBoxStyle}"/>
                                                                                 </Grid>
                                                                               </StackPanel>
                newEmployeeData.Photo =
File.ReadAllBytes(employeePhotoPath);
                                                                             </Grid>
                                                                             <Button x:Name="Save" Margin="0,0,0,0"
             }
                                                                      Content="Coxpaнить" Width="450" Style="{DynamicResource
                                                                      MainButtonStyle}" Click="Save_Click"/>
             data Base. Employee. Add (new Employee Data);\\
                                                                          </StackPanel>
             dataBase.SaveChanges();
                                                                        </Grid>
           }
                                                                      </Window>
           MessageBox.Show("Данные были сохранены", "Готово",
                                                                      AddOrEditGenre.xaml.cs
MessageBoxButton.OK, MessageBoxImage.Information);
                                                                      using System;
           this.Close();
                                                                      using System.Linq;
         }
                                                                      using System.Windows;
       }
                                                                      using static Cinema. Authorization;
       catch (Exception ex)
                                                                      namespace Cinema
         MessageBox.Show(ex.Message, "Ошибка",
                                                                      {
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                        /// <summary>
                                                                        /// Логика взаимодействия для AddOrEditGenre.xaml
       }
                                                                        /// </summary>
     }
                                                                        public partial class AddOrEditGenre: Window
  }
                                                                          public AddOrEditGenre()
AddOrEditGenre.xaml
<Window x:Class="Cinema.AddOrEditGenre"
                                                                             InitializeComponent();
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                          }
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                          private void Window_Loaded(object sender, RoutedEventArgs e)
xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
                                                                             LoadData();
    xmlns:local="clr-namespace:Cinema"
                                                                          }
     mc:Ignorable="d"
     Title="Жанр" Height="180" Width="500"
                                                                          private void LoadData()
ResizeMode="NoResize" Loaded="Window_Loaded"
                                                                          {
WindowStartupLocation="CenterScreen">
                                                                            try
  <Grid Style="{DynamicResource MainColorStyle}">
     <StackPanel>
                                                                              if (TransmittedData.idSelectedGenre != -1)
       <Grid>
                                                                                 using (var dataBase = new CinemaEntities())
```

```
MessageBox.Show(ex.Message, "Ошибка",
             var genreData = dataBase.Genre.Where(w => w.IDGenre
                                                                     MessageBoxButton.OK, MessageBoxImage.Error);
== TransmittedData.idSelectedGenre).FirstOrDefault();
                                                                          }
             GenreTitle.Text = genreData.Title;
                                                                      }
         }
                                                                      AddOrEditMovie.xaml
      }
      catch (Exception ex)
                                                                      <Window x:Class="Cinema.AddOrEditMovie"
                                                                      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
         MessageBox.Show(ex.Message, "Ошибка",
                                                                      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                      xmlns:mc="http://schemas.openxmlformats.org/markup-
    }
                                                                      compatibility/2006"
                                                                          xmlns:local="clr-namespace:Cinema"
    private void Save_Click(object sender, RoutedEventArgs e)
                                                                          mc:Ignorable="d"
                                                                          Title="Фильм" Height="620" Width="1170"
    {
                                                                      Loaded="Window_Loaded" ResizeMode="NoResize"
      try
                                                                      WindowStartupLocation="CenterScreen">
         if (GenreTitle.Text == "")
                                                                        <Grid Style="{DynamicResource MainColorStyle}">
                                                                          <StackPanel>
           MessageBox.Show("Данные не были заполнены",
                                                                             <Grid>
"Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);
                                                                               <Rectangle Style="{DynamicResource
                                                                      SecondColorRectangleStyle \}"/>
           return;
                                                                               <TextBlock Text="ФИЛЬМ" Style="{DynamicResource
                                                                     TitleTextBlockStyle}"/>
                                                                             </Grid>
         using (var dataBase = new CinemaEntities())
                                                                            <Grid HorizontalAlignment="Center" Margin="10">
           if (TransmittedData.idSelectedGenre != -1)
                                                                               <StackPanel>
                                                                                 <Grid Margin="0,0,0,10">
             var\ genre Data = data Base. Genre. Where (w => w.IDGenre
                                                                                   <Label Content="Название:" Style="{DynamicResource
== TransmittedData.idSelectedGenre).FirstOrDefault();
                                                                     MainLableStyle \}"/>
                                                                                   <TextBox x:Name="MovieTitle"
             genreData.Title = GenreTitle.Text;
                                                                      HorizontalAlignment="Left" Margin="200,0,0,0" Width="350"
                                                                      Style="{DynamicResource MainTextBoxStyle}"/>
             dataBase.SaveChanges();
                                                                                 </Grid>
                                                                                 <Grid Margin="0,0,0,10">
           else
                                                                                   <Label Content="Страна производства:"
                                                                      Style="{DynamicResource MainLableStyle}"/>
                                                                                   <ComboBox x:Name="MovieCountry"
             var newGenreData = new Genre();
                                                                     HorizontalAlignment="Left" Margin="200,0,0,0" Width="350"
             newGenreData.Title = GenreTitle.Text;
                                                                      Style="{DynamicResource MainComboBoxStyle}"
                                                                      ItemContainerStyle="{DynamicResource
             dataBase.Genre.Add(newGenreData);
                                                                      MainSelectedComboBoxStyle}"/>
             dataBase.SaveChanges();
                                                                                 </Grid>
                                                                                 <Grid Margin="0,0,0,10">
                                                                                   <Label Content="Год публикации:"
           this.Close();
                                                                      Style="{DynamicResource MainLableStyle}"/>
                                                                                   <TextBox x:Name="MovieYearOfPublication"
       }
                                                                      HorizontalAlignment="Left" Margin="200,0,0,0" Width="350"
                                                                      Style="{DynamicResource MainTextBoxStyle}"/>
      catch (Exception ex)
                                                                                 </Grid>
                                                                                 <Grid Margin="0,0,0,10">
```

```
<Label Content="Хронометраж (мин):"
                                                                                        <Button x:Name="ClearGenre" Content="X"
Style="{DynamicResource MainLableStyle}"/>
                                                                    Height="31.28" Style="{DynamicResource MainClearButtonStyle}"
             <TextBox x:Name="MovieTiming"
                                                                    Click="ClearGenre_Click"/>
HorizontalAlignment="Left" Margin="200,0,0,0" Width="350"
                                                                                      </WrapPanel>
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                                    </Grid>
           </Grid>
                                                                                    <DataGrid x:Name="MovieGenreGrid" Height="120"</p>
           <Grid Margin="0,0,0,10">
                                                                    Margin="0,0,0,10" d:ItemsSource="{d:SampleData ItemCount=5}"
             <Label Content="Возрастной рейтинг:"
                                                                    Style="{DynamicResource MainDataGridStyle}"
Style="{DynamicResource MainLableStyle}"/>
                                                                    MouseDoubleClick="MovieGenreGrid MouseDoubleClick"/>
             <TextBox x:Name="MovieAgeRating"
                                                                                    <WrapPanel HorizontalAlignment="Center">
HorizontalAlignment="Left" Margin="200,0,0,0" Width="350"
                                                                                      <Button x:Name="AddGenre" Margin="0,0,5,0"
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                    Content="Добавить" Width="160" Style="{DynamicResource
           </Grid>
                                                                    MainButtonStyle}" Click="AddGenre_Click"/>
           <Grid Margin="0,0,0,10">
                                                                                      <Button x:Name="EditGenre" Margin="0,0,5,0"
             <Label Content="Описание:"
                                                                    Content="Редактировать" Width="160" Style="{DynamicResource
                                                                    MainButtonStyle}" Click="EditGenre_Click"/>
Style="{DynamicResource MainLableStyle}"/>
             <TextBox x:Name="MovieDescription"
                                                                                      <Button x:Name="RemoveGenre"
HorizontalAlignment="Left" TextWrapping="Wrap"
                                                                    Content="Удалить" Width="160" Style="{DynamicResource
Margin="200,0,0,0" Width="350" Height="200" MaxHeight="200"
                                                                    MainButtonStyle}" Click="RemoveGenre_Click"/>
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                                    </WrapPanel>
           </Grid>
                                                                                 </StackPanel>
           <Grid>
                                                                               </Grid>
             <StackPanel>
                                                                               <Grid>
               <Grid>
                                                                                 <StackPanel>
                 <Label Content="Обложка:"
                                                                                    <Grid Margin="0,0,0,10">
Style="{DynamicResource MainLableStyle}"/>
                                                                                      <Label Content="Актеры:"
                 <Grid Margin="270,0,0,5"
                                                                    Style="{DynamicResource MainLableStyle}"/>
HorizontalAlignment="Left">
                                                                                      <WrapPanel>
                    <Image x:Name="MovieCover" Width="200"</pre>
                                                                                        <TextBox x:Name="MovieActor"
                                                                    HorizontalAlignment="Left" TextWrapping="Wrap" Margin="80,0,0,0"
Height="200" HorizontalAlignment="Center" Visibility="Collapsed"/>
                 </Grid>
                                                                    Width="455" Style="{DynamicResource MainTextBoxStyle}"
               </Grid>
                                                                    IsReadOnly="True"/>
               <Button x:Name="LoadCover" Content="Загрузить"
                                                                                        <Button x:Name="ClearActor" Content="X"
Margin="270,0,0,0" Width="200" HorizontalAlignment="Left"
                                                                    Height="31.28" Style="{DynamicResource MainClearButtonStyle}"
Style="{DynamicResource MainButtonStyle}"
                                                                    Click="ClearActor_Click"/>
Click="LoadCover Click"/>
                                                                                      </WrapPanel>
             </StackPanel>
                                                                                    </Grid>
           </Grid>
                                                                                    <DataGrid x:Name="MovieActorGrid" Height="120"</p>
         </StackPanel>
                                                                    Margin="0,0,0,10" ColumnWidth="*" d:ItemsSource="{d:SampleData
         <Rectangle Margin="558,0,560,0" Fill="Black" Width="2"
                                                                    ItemCount=5}" Background="White"
Opacity="0.2"/>
                                                                    MouseDoubleClick="MovieActorGrid_MouseDoubleClick"/>
         <StackPanel Margin="580,0,0,0">
                                                                                    <WrapPanel HorizontalAlignment="Center">
           <Grid Margin="0,0,0,10">
                                                                                      <Button x:Name="AddActor" Margin="0,0,5,0"
             <StackPanel>
                                                                    Content="Добавить" Width="160" Style="{DynamicResource
               <Grid Margin="0,0,0,10">
                                                                    MainButtonStyle}" Click="AddActor_Click"/>
                 <Label Content="Жанры:"
                                                                                      <Button x:Name="EditActor" Margin="0,0,5,0"
Style="{DynamicResource MainLableStyle}"/>
                                                                    Content="Редактировать" Width="160" Style="{DynamicResource
                 <WrapPanel>
                                                                    MainButtonStyle}" Click="EditActor_Click"/>
                    <TextBox x:Name="MovieGenere"
                                                                                      <Button x:Name="RemoveActor"
HorizontalAlignment="Left" TextWrapping="Wrap" Margin="80,0,0,0"
                                                                    Content="Удалить" Width="160" Style="{DynamicResource
Width="455" Style="{DynamicResource MainTextBoxStyle}"
                                                                    MainButtonStyle}" Click="RemoveActor_Click"/>
IsReadOnly="True"/>
                                                                                    </WrapPanel>
                                                                                 </StackPanel>
```

```
</Grid>
                                                                                    var actorAllData = dataBase.Actor.Select(s => new {
         </StackPanel>
                                                                        s.IDActor, s.Surname, s.Name, s.Patronymic, s.Nickname }).ToList();
       </Grid>
       <Button x:Name="Save" Margin="0,10,0,0"
                                                                                    Movie Genre Grid. Items Source = genre All Data; \\
Content="Coxpaнить" Width="250" Style="{DynamicResource
                                                                                    MovieActorGrid.ItemsSource = actorAllData;
MainButtonStyle}" Click="Save_Click"/>
                                                                                 }
     </StackPanel>
                                                                               }
  </Grid>
                                                                               catch (Exception ex)
</Window>
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
AddOrEditMovie.xaml.cs
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
using Microsoft.Win32;
                                                                               }
using System;
                                                                             }
using System.Collections.Generic;
using System.IO;
                                                                             private void LoadUserData()
using System.Linq;
using System. Windows;
                                                                               try
using System.Windows.Controls;
using System.Windows.Input;
                                                                                 using (var dataBase = new CinemaEntities())
using System.Windows.Media.Imaging;
using static Cinema. Authorization;
                                                                                    var countryAllData = dataBase.Country.Select(s =>
                                                                        s.Title).ToList();
namespace Cinema
                                                                                    MovieCountry.ItemsSource = countryAllData;
{
  /// <summary>
  /// Логика взаимодействия для AddOrEditMovie.xaml
                                                                                    if (TransmittedData.idSelectedMovie != -1)
  /// </summary>
                                                                                    {
  public partial class AddOrEditMovie: Window
                                                                                      var movieData = (from movie in dataBase.Movie
    public AddOrEditMovie()
                                                                                                join
                                                                                                country in dataBase.Country on
                                                                        movie.IDCountry equals country.IDCountry into countryGroup
      InitializeComponent();
                                                                                                from country in
     }
                                                                        countryGroup.DefaultIfEmpty()
    List<int> selectedGenre = new List<int>();
                                                                                                where (movie.IDMovie ==
    List<int> selectedActor = new List<int>();
                                                                        TransmittedData.idSelectedMovie)
    public string coverPath;
                                                                                                select new
    private void Window_Loaded(object sender, RoutedEventArgs e)
                                                                                                  movie.IDMovie,
                                                                                                  movie Cover.
     {
      LoadGridData();
                                                                                                  movieTitle = movie.Title,
      LoadUserData();
                                                                                                  movie.YearOfPublication,
                                                                                                  movie.Timing,
     }
                                                                                                  movie.AgeRating,
    private void LoadGridData()
                                                                                                  movie.Description,
     {
                                                                                                  movieCountry = country.Title,
      try
                                                                                                ).FirstOrDefault();
         using (var dataBase = new CinemaEntities())
                                                                                      var genreData = (from genre in dataBase.Genre
            var genreAllData = dataBase.Genre.Select(s => new {
                                                                                                join
s.IDGenre, s.Title }).ToList();
                                                                                                movieGenre in dataBase.MovieGenre on
                                                                        genre.IDGenre equals movieGenre.IDGenre into movieGenreGroup
```

```
from movieGenre in
                                                                                          bitmapImage.StreamSource = new
movieGenreGroup.DefaultIfEmpty()
                                                                         MemoryStream(movieData.Cover);
                        where (movieGenre.IDMovie ==
                                                                                          bitmapImage.EndInit();
Transmitted Data. id Selected Movie) \\
                        select new
                                                                                          MovieCover.Source = bitmapImage;
                                                                                        }
                           genre.IDGenre,
                           genre.Title,
                                                                                        foreach (var genreLine in genreData)
                        }).ToList();
                                                                                          MovieGenere.Text += "|" + genreLine.Title + "|";
              foreach (var genreLine in genreData)
                 selectedGenre.Add(genreLine.IDGenre);
                                                                                        foreach (var actorLine in actorData)
                                                                                          MovieActor.Text += "|" + actorLine.Surname + " " +
                                                                          actorLine.Name + " " + actorLine.Patronymic + " " +
              var actorData = (from actor in dataBase.Actor
                                                                          actorLine.Nickname + "|";
                        join
                        actorInMovie in dataBase.ActorsInMovies on
actor. IDActor\ equals\ actor In Movie. IDActor\ into\ actor In Movie Group
                        from actorInMovie in
                                                                                   }
actorInMovieGroup.DefaultIfEmpty()
                        where (actorInMovie.IDMovie ==
                                                                                 catch (Exception ex)
TransmittedData.idSelectedMovie)
                                                                                   MessageBox.Show(ex.Message, "Ошибка",
                        select new
                                                                         MessageBoxButton.OK, MessageBoxImage.Error);
                           actor.IDActor,
                           actor.Surname,
                                                                              }
                           actor.Name,
                                                                              private void ClearGenre_Click(object sender, RoutedEventArgs e)
                           actor.Patronymic,
                           actor.Nickname,
                        }).ToList();
                                                                                 selectedGenre.Clear();
                                                                                 MovieGenere.Clear();
              foreach (var actorLine in actorData)
                 selectedActor.Add(actorLine.IDActor);
                                                                              private void ClearActor_Click(object sender, RoutedEventArgs e)
                                                                                 selectedActor.Clear();
              MovieTitle.Text = movieData.movieTitle;
                                                                                 MovieActor.Clear();
              MovieCountry.SelectedItem = movieData.movieCountry;
              MovieYearOfPublication.Text =
movieData.YearOfPublication.ToString();
                                                                              private void LoadCover_Click(object sender, RoutedEventArgs e)
              Movie Timing. Text = movie Data. Timing. To String(); \\
              MovieAgeRating.Text =
                                                                                 try
movieData.AgeRating.ToString();
              MovieDescription.Text = movieData.Description;
                                                                                   OpenFileDialog openFileDialog = new OpenFileDialog();
                                                                                   openFileDialog.Filter = "JPG (*.jpg)|*.jpg|PNG
              if (movieData.Cover != null)
                                                                         (*.png)|*.png";
              {
                 this. Height = 780;
                                                                                   if (openFileDialog.ShowDialog() == true)
                 MovieCover.Visibility = Visibility.Visible;
                                                                                     coverPath = openFileDialog.FileName;
                 BitmapImage bitmapImage = new BitmapImage();
                 bitmapImage.BeginInit();
                                                                                     this. Height = 780;
```

```
MovieCover.Visibility = Visibility.Visible;
                                                                                       var removeGenre = dataBase.MovieGenre.Where(w =>
           MovieCover.Source = new BitmapImage(new
                                                                         w.IDMovie == TransmittedData.idSelectedMovie).ToList();
Uri(openFileDialog.FileName));
                                                                                       if (removeGenre != null)
                                                                                         data Base. Movie Genre. Remove Range (remove Genre);\\
         }
       }
       catch (Exception ex)
                                                                                       foreach (var genreLine in selectedGenre)
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                         var newMovieGenre = new MovieGenre();
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                         newMovieGenre.IDMovie = movieData.IDMovie;
                                                                                         newMovieGenre.IDGenre = genreLine;
       }
     }
                                                                                         dataBase.MovieGenre.Add(newMovieGenre);
    private void Save_Click(object sender, RoutedEventArgs e)
       try
                                                                                       var removeActorInMovie =
                                                                         dataBase.ActorsInMovies.Where(w => w.IDMovie ==
         if (MovieCountry.SelectedIndex <0 \parallel MovieTitle.Text == "" \parallel
                                                                        TransmittedData.idSelectedMovie).ToList();
!int.TryParse(MovieYearOfPublication.Text, out \_) \parallel
                                                                                       if (removeActorInMovie != null)
!double.TryParse(MovieTiming.Text,\,out\,\_) \parallel
!int.TryParse(MovieAgeRating.Text, out \_) \parallel MovieDescription.Text ==
                                                                         dataBase.ActorsInMovies.RemoveRange(removeActorInMovie);
"" || MovieGenere.Text == "" || MovieActor.Text == "")
                                                                                       foreach (var actorLine in selectedActor)
           MessageBox.Show("Данные заполнены неправильно",
"Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);
                                                                                         var newMovieActor = new ActorsInMovies();
           return:
                                                                                         newMovieActor.IDMovie = movieData.IDMovie;
                                                                                         newMovieActor.IDActor = actorLine;
         using (var dataBase = new CinemaEntities())
                                                                                         dataBase.ActorsInMovies.Add(newMovieActor);
           if (TransmittedData.idSelectedMovie != -1)
                                                                                       dataBase.SaveChanges();
              var movieData = dataBase.Movie.Where(w =>
                                                                                    }
w.IDMovie == TransmittedData.idSelectedMovie).FirstOrDefault();
                                                                                    else
              movieData.Title = MovieTitle.Text;
                                                                                       var newMovieData = new Movie();
              var selectedCountry = dataBase.Country.Where(w =>
                                                                                       newMovieData.Title = MovieTitle.Text;
w. Title == Movie Country. Selected Item. To String()). First Or Default(); \\
              movie Data. ID Country = selected Country. ID Country; \\
                                                                                       var selectedCountry = dataBase.Country.Where(w =>
                                                                        w.Title == MovieCountry.SelectedItem.ToString()).FirstOrDefault();
                                                                                       newMovieData.IDCountry = selectedCountry.IDCountry;
              movieData.YearOfPublication =
Convert.ToInt32(MovieYearOfPublication.Text);
              movieData.Timing =
                                                                                       newMovieData.YearOfPublication =
Convert.ToInt32(MovieTiming.Text);
                                                                         Convert.ToInt32(MovieYearOfPublication.Text);
              movie Data. Age Rating = \\
                                                                                       newMovieData.Timing =
Convert.ToInt32(MovieAgeRating.Text);
                                                                         Convert.ToInt32(MovieTiming.Text);
              movieData.Description = MovieDescription.Text;
                                                                                       newMovieData.AgeRating =
                                                                         Convert.ToInt32(MovieAgeRating.Text);
              if (coverPath != null)
                                                                                       newMovieData.Description = MovieDescription.Text;
                movie Data. Cover = File. Read All Bytes (cover Path); \\
                                                                                       if (coverPath != null)
```

```
newMovieData.Cover =
                                                                                 DataGridCell cellTitle =
File.ReadAllBytes(coverPath);
                                                                        MovieGenreGrid.Columns[1].GetCellContent(row).Parent as
                                                                        DataGridCell;
                                                                                 int\ idGenre =
                                                                        Convert.ToInt32(((TextBlock)cellId.Content).Text);
              dataBase.Movie.Add(newMovieData);
              dataBase.SaveChanges();
                                                                                 string titleGenre = ((TextBlock)cellTitle.Content).Text;
              foreach (var genreLine in selectedGenre)
                                                                                 bool availabilityGenre = false;
              {
                                                                                 foreach (var genreLine in selectedGenre)
                var newMovieGenre = new MovieGenre();
                newMovieGenre.IDMovie = newMovieData.IDMovie;
                                                                                   if (genreLine == idGenre)
                newMovieGenre.IDGenre = genreLine;
                                                                                      availabilityGenre = true;
                dataBase.MovieGenre.Add(newMovieGenre);
              foreach (var actorLine in selectedActor)
                                                                                 if (!availabilityGenre)
                var newMovieActor = new ActorsInMovies():
                                                                                    selectedGenre.Add(idGenre):
                newMovieActor.IDMovie = newMovieData.IDMovie;
                                                                                    MovieGenere.Text += "|" + titleGenre + "|";
                newMovieActor.IDActor = actorLine;
                dataBase.ActorsInMovies.Add(newMovieActor);
                                                                               catch (Exception ex)
              }
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
              dataBase.SaveChanges();
                                                                             }
           MessageBox.Show("Данные были сохранены", "Готово",
MessageBoxButton.OK, MessageBoxImage.Information);
                                                                            private void MovieActorGrid_MouseDoubleClick(object sender,
           this.Close();
                                                                        MouseButtonEventArgs e)
                                                                             {
         }
       }
                                                                               try
       catch (Exception ex)
                                                                                 DataGridRow row =
         MessageBox.Show(ex.Message, "Ошибка",
                                                                        (DataGridRow)MovieActorGrid.ItemContainerGenerator.ContainerFro
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                        mIndex(MovieActorGrid.SelectedIndex);
       }
                                                                                 DataGridCell cellId =
                                                                        Movie Actor Grid. Columns [0]. Get Cell Content (row). Parent\ as
     }
                                                                        DataGridCell:
    private void MovieGenreGrid_MouseDoubleClick(object sender,
                                                                                 DataGridCell cellSurname =
MouseButtonEventArgs e)
                                                                        Movie Actor Grid. Columns \hbox{\tt [1]}. Get Cell Content (row). Parent\ as
                                                                        DataGridCell;
                                                                                 DataGridCell cellName =
       try
                                                                        MovieActorGrid.Columns[2].GetCellContent(row).Parent as
       {
         DataGridRow row =
                                                                        DataGridCell;
(DataGridRow)MovieGenreGrid.ItemContainerGenerator.ContainerFro
                                                                                 DataGridCell cellPatronymic =
mIndex(MovieGenreGrid.SelectedIndex);
                                                                        MovieActorGrid.Columns[3].GetCellContent(row).Parent as
         DataGridCell cellId =
                                                                        DataGridCell;
Movie Genre Grid. Columns [0]. Get Cell Content (row). Parent\ as
                                                                                 DataGridCell cellNickname =
DataGridCell;
                                                                        MovieActorGrid.Columns[4].GetCellContent(row).Parent as
                                                                        DataGridCell;
```

```
int idActor =
                                                                                 if (MovieGenreGrid.SelectedIndex >= 0)
Convert.ToInt32(((TextBlock)cellId.Content).Text);
         string surnameActor =
                                                                                   DataGridRow row =
((TextBlock)cellSurname.Content).Text;
                                                                        (Data Grid Row) Movie Genre Grid. Item Container Generator. Container Fro\\
         string nameActor = ((TextBlock)cellName.Content).Text;
                                                                        mIndex(MovieGenreGrid.SelectedIndex) as DataGridRow;
         string patronymicActor =
                                                                                    DataGridCell cell =
((TextBlock)cellPatronymic.Content).Text;
                                                                        MovieGenreGrid.Columns[0].GetCellContent(row).Parent as
         string nicknameActor =
                                                                        DataGridCell;
((TextBlock)cellNickname.Content).Text;
                                                                                   TransmittedData.idSelectedGenre =
                                                                        Convert.ToInt32(((TextBlock)cell.Content).Text);
         bool availabilityActor = false;
         foreach (var actorLine in selectedActor)
                                                                                   AddOrEditGenre addOrEditGenre = new
                                                                        AddOrEditGenre();
                                                                                   addOrEditGenre.ShowDialog();
           if (actorLine == idActor)
                                                                                   LoadGridData();
              availabilityActor = true;
                                                                                 else
                                                                                   MessageBox.Show("Выбирите строку для
         if (!availabilityActor)
                                                                        редактирования", "Внимание", MessageBoxButton.OK,
                                                                        MessageBoxImage.Warning);
           selectedActor.Add(idActor);
                                                                                   return;
           MovieActor.Text += "|" + surnameActor + " " + nameActor
+ " " + patronymicActor + " " + nicknameActor + " |";
         }
                                                                               catch (Exception ex)
       }
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
       catch (Exception ex)
                                                                        MessageBoxButton.OK, MessageBoxImage.Error);
         MessageBox.Show(ex.Message, "Ошибка",
Message Box Button. OK, \, Message Box Image. Error); \,
                                                                            }
                                                                            private void RemoveGenre_Click(object sender, RoutedEventArgs
     }
                                                                       e)
    private void AddGenre_Click(object sender, RoutedEventArgs e)
     {
                                                                               try
      try
                                                                                 if (MovieGenreGrid.SelectedIndex \geq= 0)
         TransmittedData.idSelectedGenre = -1;
         AddOrEditGenre addOrEditGenre = new AddOrEditGenre();
                                                                                   if (MessageBox.Show("Вы действительно хотите удалить
         addOrEditGenre.ShowDialog();
                                                                        выбранный жанр?", "Внимание", MessageBoxButton.YesNo,
         LoadGridData();
                                                                        MessageBoxImage.Warning) == MessageBoxResult.Yes)
       catch (Exception ex)
                                                                                      using (var dataBase = new CinemaEntities())
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                        DataGridRow row =
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                        (Data Grid Row) Movie Genre Grid. Item Container Generator. Container Fro\\
                                                                        mIndex(MovieGenreGrid.SelectedIndex) as DataGridRow;
       }
                                                                                        DataGridCell cell =
     }
                                                                        MovieGenreGrid.Columns[0].GetCellContent(row).Parent as
    private void EditGenre_Click(object sender, RoutedEventArgs e)
                                                                        DataGridCell;
                                                                                        int selectedGenre =
                                                                        Convert.ToInt32(((TextBlock)cell.Content).Text);
      try
```

```
var removeGenre = dataBase.Genre.Where(w =>
w.IDGenre == selectedGenre).FirstOrDefault();
                              var removeMovieGenre =
dataBase.MovieGenre.Where(w => w.IDGenre ==
                                                                                                                                              private void AddActor_Click(object sender, RoutedEventArgs e)
selectedGenre).ToList();
                                                                                                                                                   try
                              if (removeMovieGenre.Count != 0)
                                                                                                                                                       TransmittedData.idSelectedActor = -1;
                                                                                                                                                       AddOrEditActor addOrEditActor = new AddOrEditActor();
                                  if (MessageBox.Show("Удаляемый жанр
используеться в фильмах. Вы действительно хотите его удалить?",
                                                                                                                                                       addOrEditActor.ShowDialog();
"Внимание", MessageBoxButton. YesNo, MessageBoxImage. Warning)
                                                                                                                                                       LoadGridData();
== MessageBoxResult.Yes)
                                                                                                                                                   catch (Exception ex)
                                                                                                                                                       MessageBox.Show(ex.Message, "Ошибка",
dataBase.MovieGenre.RemoveRange(removeMovieGenre);
                                      dataBase.Genre.Remove(removeGenre);
                                                                                                                                     MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                                                                                                   }
                                                                                                                                              }
                                       dataBase.SaveChanges();
                                      MessageBox.Show("Жанр был удален",
                                                                                                                                              private void EditActor_Click(object sender, RoutedEventArgs e)
"Готово", MessageBoxButton.OK, MessageBoxImage.Information);
                                      LoadGridData();
                                                                                                                                                  try
                                   }
                                                                                                                                                      if (MovieActorGrid.SelectedIndex >= 0)
                                  else
                                       return;
                                                                                                                                                           DataGridRow row =
                                                                                                                                      (Data Grid Row) Movie Actor Grid. Item Container Generator. Container Fround From Container Generator. Container Container
                              }
                                                                                                                                      mIndex(MovieActorGrid.SelectedIndex) as DataGridRow;
                                                                                                                                                           DataGridCell cell =
                              else
                                                                                                                                     MovieActorGrid.Columns[0].GetCellContent(row).Parent as
                                                                                                                                      DataGridCell;
                                  dataBase.Genre.Remove(removeGenre);
                                                                                                                                                            TransmittedData.idSelectedActor =
                                  dataBase.SaveChanges();
                                                                                                                                      Convert.ToInt32(((TextBlock)cell.Content).Text);
                                  MessageBox.Show("Жанр был удален", "Готово",
                                                                                                                                                           AddOrEditActor addOrEditActor = new AddOrEditActor();
MessageBoxButton.OK, MessageBoxImage.Information);
                                                                                                                                                           addOrEditActor.ShowDialog();
                                  LoadGridData();
                                                                                                                                                           LoadGridData();
                                                                                                                                                       else
                 }
                                                                                                                                                           MessageBox.Show("Выбирите строку для
                                                                                                                                      редактирования", "Внимание", MessageBoxButton.OK,
                 else
                                                                                                                                      MessageBoxImage.Warning);
                     MessageBox.Show("Выбирите строку для
                                                                                                                                                           return:
редактирования", "Внимание", MessageBoxButton.OK,
                                                                                                                                                       }
MessageBoxImage.Warning);
                                                                                                                                                  catch (Exception ex)
                     return;
                 }
                                                                                                                                                       MessageBox.Show(ex.Message, "Ошибка",
             catch (Exception ex)
                                                                                                                                     MessageBoxButton.OK,\,MessageBoxImage.Error);
                 MessageBox.Show(ex.Message, "Ошибка",
                                                                                                                                              }
MessageBoxButton.OK,\,MessageBoxImage.Error);
```

```
private void RemoveActor_Click(object sender, RoutedEventArgs
                                                                                        dataBase.SaveChanges();
e)
                                                                                        MessageBox.Show("Актер был удален", "Готово",
                                                                      MessageBoxButton.OK, MessageBoxImage.Information);
      try
                                                                                       LoadGridData();
        if (MovieActorGrid.SelectedIndex >= 0)
           if (MessageBox.Show("Вы действительно хотите удалить
                                                                                 }
выбранный жанр?", "Внимание", MessageBoxButton. YesNo,
MessageBoxImage.Warning) == MessageBoxResult.Yes)
                                                                              else
             using (var dataBase = new CinemaEntities())
                                                                                 MessageBox.Show("Выбирите строку для
                                                                      редактирования", "Внимание", MessageBoxButton.OK,
                                                                      MessageBoxImage.Warning);
                DataGridRow row =
(Data Grid Row) Movie Actor Grid. Item Container Generator. Container Fro\\
                                                                                 return;
mIndex(MovieActorGrid.SelectedIndex) as DataGridRow;
                DataGridCell cell =
Movie Actor Grid. Columns [0]. Get Cell Content (row). Parent \ as
                                                                            catch (Exception ex)
DataGridCell;
               int selectedActor =
                                                                              MessageBox.Show(ex.Message, "Ошибка",
Convert.ToInt32(((TextBlock)cell.Content).Text);
                                                                     MessageBoxButton.OK, MessageBoxImage.Error);
                var removeActor = dataBase.Actor.Where(w =>
w.IDActor == selectedActor).FirstOrDefault();
                var removeActorInMovie =
                                                                        }
dataBase.ActorsInMovies.Where(w => w.IDActor ==
selectedActor).ToList();
                                                                      AddOrEditSession.xaml
                if (removeActorInMovie.Count != 0)
                                                                      <Window x:Class="Cinema.AddOrEditSession"
                                                                      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                  if (MessageBox.Show("Удаляемый актер
присутствует в других фильмах. Вы действительно хотите его
                                                                      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
удалить?", "Внимание", MessageBoxButton. YesNo,
                                                                      xmlns:mc="http://schemas.openxmlformats.org/markup-
MessageBoxImage.Warning) == MessageBoxResult.Yes)
                                                                      compatibility/2006"
                                                                          xmlns:local="clr-namespace:Cinema"
                                                                          mc:Ignorable="d"
dataBase.ActorsInMovies.RemoveRange(removeActorInMovie);
                                                                          Title="Ceaнc" Height="300" Width="520"
                    dataBase.Actor.Remove(removeActor);
                                                                     Loaded="Window_Loaded" ResizeMode="NoResize"
                                                                      WindowStartupLocation="CenterScreen">
                                                                        <Grid Style="{DynamicResource MainColorStyle}">
                    dataBase.SaveChanges();
                                                                          <StackPanel>
                    MessageBox.Show("Актер был удален",
                                                                            <Grid>
"Готово", MessageBoxButton.OK, MessageBoxImage.Information);
                                                                               <Rectangle Style="{DynamicResource
                    LoadGridData();
                                                                     SecondColorRectangleStyle}"/>
                                                                               <TextBlock Text="CEAHC" Style="{DynamicResource
                  }
                                                                     TitleTextBlockStyle}"/>
                  else
                  {
                                                                            </Grid>
                                                                            <Grid HorizontalAlignment="Center" Margin="10">
                    return;
                                                                               <StackPanel>
                                                                                 <Grid Margin="0,0,0,10">
                else
                                                                                   <Label Content="Фильм:" Style="{DynamicResource
                                                                      MainLableStyle \}"/>
                                                                                   <ComboBox x:Name="SessionMovie"
                  dataBase.Actor.Remove(removeActor);
                                                                     HorizontalAlignment="Left" Margin="120,0,0,0" Width="350"
```

```
Style="{DynamicResource MainComboBoxStyle}"
ItemContainerStyle="{DynamicResource
                                                                            private void Window_Loaded(object sender, RoutedEventArgs e)
MainSelectedComboBoxStyle}"/>
           </Grid>
                                                                              LoadData();
           <Grid Margin="0,0,0,10">
              <Label Content="Дата сеанса:"
Style="{DynamicResource MainLableStyle}"/>
                                                                            private void LoadData()
              <DatePicker x:Name="SessionDate"
HorizontalAlignment="Left" Margin="120,0,0,0" Width="350"
                                                                              try
Style="{DynamicResource MainDatePickerStyle}"/>
                                                                              {
           </Grid>
                                                                                 using (var dataBase = new CinemaEntities())
            <Grid Margin="0,0,0,10">
              <Label Content="Время сеанса:"
                                                                                   var movieData = dataBase.Movie.ToList();
Style="{DynamicResource MainLableStyle}"/>
                                                                                   List<string> movieDataList = new List<string>();
              <TextBox x:Name="SessionTime"
HorizontalAlignment="Left" Margin="120,0,0,0" Width="350"
                                                                                   foreach (var movieLine in movieData)
Style="{DynamicResource MainTextBoxStyle}"/>
            </Grid>
                                                                                     movieDataList.Add(movieLine.IDMovie + "|" +
           <Grid Margin="0,0,0,10">
                                                                       movieLine.Title);
              <Label Content="Цена билета:"
Style="{DynamicResource MainLableStyle}"/>
              <TextBox x:Name="SessionTicketPrice"
                                                                                   SessionMovie.ItemsSource = movieDataList;
Horizontal Alignment="Left" Margin="120,0,0,0" Width="350"
Style="{DynamicResource MainTextBoxStyle}"/>
                                                                                   if (TransmittedData.idSelectedSession != -1)
            </Grid>
         </StackPanel>
                                                                                     var sessionData = (from session in dataBase.Session
       </Grid>
                                                                                                join
       <Button x:Name="Save" Margin="0,0,0,0"
                                                                                                movie in dataBase.Movie on
Content="Coxpaнить" Width="470" Style="{DynamicResource
                                                                       session.IDMovie equals movie.IDMovie into movieGroup
MainButtonStyle}" Click="Save_Click"/>
                                                                                                from movie in movieGroup.DefaultIfEmpty()
     </StackPanel>
                                                                                                where (session.IDSession ==
   </Grid>
                                                                       TransmittedData.idSelectedSession)
</Window>
                                                                                                select new
AddOrEditSession.xaml.cs
                                                                                                  movie.IDMovie,
                                                                                                  movie.Title,
using System;
using System.Collections.Generic;
                                                                                                  session.DateAndTimeSession,
using System.Linq;
                                                                                                  session.TicketPrice
using System.Text.RegularExpressions;
                                                                                                }).FirstOrDefault();
using System. Windows;
using static Cinema. Authorization;
                                                                                     Session Movie. Selected Item = session Data. ID Movie + "|"
                                                                        + sessionData.Title;
namespace Cinema
                                                                                     SessionDate.SelectedDate = \\
                                                                       sessionData.DateAndTimeSession.Date:
                                                                                     SessionTime.Text =
  /// <summary>
  /// Логика взаимодействия для AddOrEditSession.xaml
                                                                       session Data. Date And Time Session. Time Of Day. To String(); \\
                                                                                     SessionTicketPrice.Text =
  /// </summary>
  public partial class AddOrEditSession: Window
                                                                       sessionData.TicketPrice.ToString();
    public AddOrEditSession()
                                                                                 }
       InitializeComponent();
                                                                              catch (Exception ex)
```

```
MessageBox.Show(ex.Message, "Ошибка",
                                                                                     string[] sessionMovieTemp =
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                       SessionMovie.SelectedItem.ToString().Split('|');
                                                                                     string[] sessionDateTemp =
                                                                       SessionDate.SelectedDate.Value.ToString().Split(' ');
     }
    private void Save_Click(object sender, RoutedEventArgs e)
                                                                                     newSessionData.IDMovie =
                                                                       Convert.ToInt32(sessionMovieTemp[0]);
                                                                                     newSessionData.DateAndTimeSession = \\
       try
                                                                       Convert.ToDateTime(sessionDateTemp[0] + " " + SessionTime.Text);
                                                                                     newSessionData.TicketPrice =
         string timePattern = @ ^{(2)}(2:[01]\d2[0-3]):[0-5]\d:[0-5]\d);
                                                                       Convert.ToDecimal(SessionTicketPrice.Text);
         if (SessionMovie.SelectedIndex < 0 ||
SessionDate.SelectedDate == null || SessionTicketPrice.Text == null ||
!decimal.TryParse(SessionTicketPrice.Text, out _))
                                                                                     dataBase.Session.Add(newSessionData);
           MessageBox.Show("Данные были заполнены неверно",
                                                                                     dataBase.SaveChanges();
"Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);
           return;
                                                                                   MessageBox.Show("Данные были сохранены", "Готово",
                                                                       MessageBoxButton.OK, MessageBoxImage.Information);
         if (!Regex.IsMatch(SessionTime.Text, timePattern))
                                                                                   this.Close();
           MessageBox.Show("Время указываеться в формате
00:00:00", "Внимание", MessageBoxButton.OK,
                                                                              catch (Exception ex)
MessageBoxImage.Warning);
           return;
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                       Message Box Button. OK, Message Box Image. Error);\\
         using (var dataBase = new CinemaEntities())
                                                                            }
                                                                          }
           if (TransmittedData.idSelectedSession != -1)
              var sessionData = dataBase.Session.Where(w =>
                                                                       AdminPanel.xaml
w.IDSession == TransmittedData.idSelectedSession).FirstOrDefault();
                                                                        <Window x:Class="Cinema.AdminPanel"
                                                                       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
              string[] sessionMovieTemp =
SessionMovie.SelectedItem.ToString().Split('|');
                                                                        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
              string[] sessionDateTemp =
                                                                       xmlns:mc="http://schemas.openxmlformats.org/markup-
Session Date. Selected Date. Value. To String (). Split ('\ ');
                                                                       compatibility/2006"
                                                                            xmlns:local="clr-namespace:Cinema"
              sessionData.IDMovie =
                                                                            mc:Ignorable="d"
                                                                            Title="Панель администратора" Height="550" Width="1050"
Convert.ToInt32(sessionMovieTemp[0]);
              session Data. Date And Time Session = \\
                                                                       MinHeight="550" MinWidth="1050" Loaded="Window_Loaded"
Convert.ToDateTime(sessionDateTemp[0] + " " + SessionTime.Text);
                                                                       Closing="Window_Closing" WindowStartupLocation="CenterScreen">
              sessionData.TicketPrice =
Convert.ToDecimal(SessionTicketPrice.Text);
                                                                          <Grid Style="{DynamicResource MainColorStyle}">
                                                                            <Grid VerticalAlignment="Top">
                                                                               <WrapPanel Style="{DynamicResource
              dataBase.SaveChanges();
                                                                       SecondColorWrapPanelStyle}">
            }
           else
                                                                                 <Button x:Name="EmployeePage" Content="Сотрудники"
                                                                       Style="{DynamicResource TabOffButtonStyle}"
                                                                       Click="SelectedPage_Click"/>
              var newSessionData = new Session();
```

```
<Button x:Name="SettingsPage" Content="Настройки зала"
                                                                                 PageManager.Navigate(new EmployeeControls());
Style="{DynamicResource TabOffButtonStyle}"
Click="SelectedPage_Click"/>
                                                                                currentSelectedButton = EmployeePage;
         <Button x:Name="BackUpPage" Content="Опции"
                                                                                current Selected Button. Style = \\
Style="{DynamicResource TabOffButtonStyle}"
                                                                       (Style)Application.Current.Resources["TabOnButtonStyle"];
Click="SelectedPage_Click"/>
       </WrapPanel>
                                                                              catch (Exception ex)
       <WrapPanel HorizontalAlignment="Right">
         <Button x:Name="Help" Content="Справка" Width="100"
                                                                                MessageBox.Show(ex.Message, "Ошибка",
Style="{DynamicResource TabHelpButtonStyle}"
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
Click="Help_Click"/>
         <Button x:Name="Exit" Content="Выйти" Width="100"
                                                                            }
Style="{DynamicResource TabExitButtonStyle}" Click="Exit_Click"/>
       </WrapPanel>
                                                                            private void SelectedPage_Click(object sender, RoutedEventArgs
     </Grid>
                                                                       e)
     <Frame x:Name="PageManager" Content="MainFrame"</p>
Margin="0,25,0,0" NavigationUIVisibility="Hidden"
                                                                              try
Background="White"/>
   </Grid>
                                                                                if (currentSelectedButton != null)
</Window>
                                                                                   currentSelectedButton.Style = \\
AdminPanel.xaml.cs
                                                                       (Style)Application.Current.Resources["TabOffButtonStyle"];
using Cinema.Pages;
using System;
using System.Diagnostics;
                                                                                 currentSelectedButton = sender as Button;
using System.IO;
using System.Reflection;
                                                                                current Selected Button. Style = \\
using System.Reflection.Emit;
                                                                       (Style) Application. Current. Resources ["TabOnButtonStyle"];\\
using System. Windows;
using System.Windows.Controls;
                                                                                 switch (currentSelectedButton.Name)
using System.Windows.Media;
                                                                                   case "EmployeePage":
using System.Windows.Media.Media3D;
using static Cinema. Authorization;
                                                                                     PageManager.Navigate(new EmployeeControls());
                                                                                     break;
namespace Cinema
                                                                                   case "SettingsPage":
  /// <summary>
                                                                                     PageManager.Navigate(new SettingsControls());
  /// Логика взаимодействия для AdminPanel.xaml
                                                                                     break;
  /// </summary>
  public partial class AdminPanel: Window
                                                                                   case "BackUpPage":
                                                                                     PageManager.Navigate(new BackUpControls());
     public AdminPanel()
                                                                                     break:
                                                                                 }
       InitializeComponent();
                                                                              catch (Exception ex)
     public bool exitMode;
                                                                                MessageBox.Show(ex.Message, "Ошибка",
     public Button currentSelectedButton;
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
    private void Window_Loaded(object sender, RoutedEventArgs e)
                                                                            }
                                                                            private void Exit_Click(object sender, RoutedEventArgs e)
       try
       {
                                                                            {
```

```
try
                                                                                 resourceStream.CopyTo(fileStream);
         exitMode = true;
                                                                               Process.Start(new ProcessStartInfo(tempFileName) {
        TransmittedData.idEmployee = 0;
                                                                      UseShellExecute = true });
         TransmittedData.idSelectedMovie = 0;
         TransmittedData.idSelectedGenre = 0;
         TransmittedData.idSelectedActor = 0;
                                                                             catch (Exception ex)
         TransmittedData.idSelectedSession = 0;
         TransmittedData.idSelectedEmployee = 0;
                                                                               MessageBox.Show(ex.Message, "Ошибка",
                                                                      MessageBoxButton.OK, MessageBoxImage.Error);
         Authorization authorization = new Authorization();
         authorization.Show();
                                                                           }
         this.Close();
       catch (Exception ex)
                                                                      Authorization.xaml
                                                                      <Window x:Class="Cinema.Authorization"
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                           xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    }
                                                                           xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                           xmlns:mc="http://schemas.openxmlformats.org/markup-
    private void Window_Closing(object sender,
                                                                      compatibility/2006"
System.ComponentModel.CancelEventArgs e)
                                                                           xmlns:local="clr-namespace:Cinema"
                                                                           mc:Ignorable="d"
                                                                           Title="Авторизация" Height="240" Width="400"
       try
                                                                      ResizeMode="NoResize" Loaded="Window_Loaded"
                                                                      Closing="Window_Closing" WindowStartupLocation="CenterScreen">
         if (!exitMode)
                                                                        <Grid Style="{DynamicResource MainColorStyle}">
           Application.Current.Shutdown();
                                                                           <StackPanel>
                                                                             <Grid Margin="0,0,0,10">
                                                                               <Rectangle Style="{DynamicResource
       catch (Exception ex)
                                                                      SecondColorRectangleStyle}"/>
                                                                               <TextBlock Text="АВТОРИЗАЦИЯ"
         MessageBox.Show(ex.Message, "Ошибка",
                                                                      Style="{DynamicResource TitleTextBlockStyle}"/>
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                             </Grid>
       }
                                                                             <StackPanel HorizontalAlignment="Center">
    }
                                                                               <Grid>
                                                                                  <Label Content="Имя БД" HorizontalAlignment="Left"
                                                                      Margin="0,0,0,0" VerticalAlignment="Center"
    private void Help_Click(object sender, RoutedEventArgs e)
                                                                      Style="{DynamicResource MainLableStyle}"/>
                                                                                  <TextBox x:Name="DataBaseConnection"
       try
                                                                      HorizontalAlignment="Left" Margin="80,0,0,0"
         Uri resourceUri = new
                                                                      VerticalAlignment="Center" Width="250" Style="{DynamicResource
Uri("pack://application:,,,/Resource/HelpDocument.pdf",
                                                                      MainTextBoxStyle}" KeyDown="DataBaseConnection_KeyDown"/>
UriKind.Absolute);
                                                                               </Grid>
                                                                               <Grid>
         Stream resourceStream =
                                                                                  <Label Content="Логин" HorizontalAlignment="Left"
Application.GetResourceStream(resourceUri)?.Stream;
                                                                      Margin="0,0,0,0" VerticalAlignment="Center"
         string tempFileName = Path.GetTempFileName() + ".pdf";
                                                                      Style="{DynamicResource MainLableStyle}"/>
         using (FileStream fileStream = new
                                                                                 <TextBox x:Name="Login" HorizontalAlignment="Left"
FileStream(tempFileName, FileMode.Create, FileAccess.Write))
                                                                      Margin="80,0,0,0" VerticalAlignment="Center" Width="250"
```

```
Style="{DynamicResource MainTextBoxStyle}"
                                                                              // Передоваемые данные
KeyDown="Login_KeyDown"/>
                                                                              public static class TransmittedData
         </Grid>
         <Grid>
                                                                                public static int idEmployee { get; set; }
                                                                                public static int idSelectedMovie { get; set; }
           <Label Content="Пароль" HorizontalAlignment="Left"
Margin="0,0,0,0" VerticalAlignment="Center"
                                                                                public static int idSelectedGenre { get; set; }
Style="{DynamicResource MainLableStyle}"/>
                                                                                public static int idSelectedActor { get; set; }
            <TextBox x:Name="Password"
                                                                                public static int idSelectedSession { get; set; }
HorizontalAlignment="Left" Margin="80,0,0,0"
                                                                                public static int idSelectedEmployee { get; set; }
VerticalAlignment="Center" Width="250" Style="{DynamicResource
                                                                                public static int idSelectedCashierMovie { get; set; }
MainTextBoxStyle}" KeyDown="Password_KeyDown"/>
                                                                                public static int idSelectedCashierSession { get; set; }
         </Grid>
                                                                                public static int idSelectedMovieAnalysis { get; set; }
         <Button x:Name="Enter" Content="Войти"
Margin="0,10,0,0" Width="200" FontSize="16"
HorizontalAlignment="Center" Style="{DynamicResource
                                                                              private void Window_Loaded(object sender, RoutedEventArgs e)
MainButtonStyle}" Click="Enter_Click"/>
       </StackPanel>
                                                                                try
     </StackPanel>
                                                                                  if (File.Exists("ConnectionData.ini"))
     <Grid VerticalAlignment="Bottom">
       <Label Content="v1.0.5" HorizontalAlignment="Left"/>
                                                                                     DataBaseConnection.Text =
       <Label Content="Early access" Visibility="Collapsed"</pre>
                                                                         File.ReadAllText("ConnectionData.ini");
HorizontalAlignment="Right"/>
    </Grid>
                                                                                }
                                                                                catch (Exception ex)
     <ProgressBar x:Name="LoadView" VerticalAlignment="Bottom"</p>
Visibility="Collapsed" Style="{DynamicResource
                                                                                  MessageBox.Show(ex.Message, "Ошибка",
MainProgressBarStyle \}"/>
                                                                         MessageBoxButton.OK, MessageBoxImage.Error);
  </Grid>
</Window>
                                                                              }
Authorization.xaml.cs
                                                                              private async void AuthorizationUser()
using System;
using System.Configuration;
                                                                                try
using System.IO;
using System.Linq;
                                                                                  LoadView. Visibility = Visibility. Visible;
using System. Threading. Tasks;
using System. Windows;
                                                                                  string password = Password.Text;
using System.Windows.Controls;
                                                                                  string login = Login.Text;
using System.Windows.Input;
                                                                                  string dataSource = DataBaseConnection.Text;
                                                                                  var userData = await Task.Run(() =>
namespace Cinema
                                                                         LoadDataBase(password, login, dataSource));
  /// <summary>
                                                                                  LoadView. Visibility = Visibility. Collapsed;
  /// Логика взаимодействия для Authorization.xaml
  /// </summary>
  public partial class Authorization: Window
                                                                                  if (userData != null)
     public Authorization()
                                                                                     switch (userData)
       InitializeComponent();
                                                                                       case "Букер":
                                                                                         BookerPanel bookerPanel = new BookerPanel();
                                                                                         bookerPanel.Show();
```

```
this.Hide();
                                                                                                  emploee.IDEmployee,
                break;
                                                                                                  Role = role.Title
                                                                                               }).FirstOrDefault();
              case "Кассир":
                CashierPanel cashierPanel = new CashierPanel();
                                                                                     if (userData != null)
                cashierPanel.Show();
                                                                                        Transmitted Data. id Employee = user Data. ID Employee; \\
                this.Hide();
                break;
                                                                                        return userData.Role;
              case "Директор":
                                                                                     }
                DirectorsPanel directorsPanel = new DirectorsPanel();
                                                                                     else
                directorsPanel.Show();
                this.Hide();
                                                                                        return null;
                break;
              case "Администратор":
                AdminPanel adminPanel = new AdminPanel();
                                                                                 catch (Exception ex)
                adminPanel.Show();
                this.Hide();
                                                                                   MessageBox.Show(ex.Message, "Ошибка",
                break;
                                                                         MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                   return null;
            }
         }
                                                                                 }
         else
                                                                              }
         {
            MessageBox.Show("Неверный логин или пароль",
                                                                              private void ConnectionStringChanged(string newDataSource)
"Внимание", MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
                                                                                 try
                                                                                   string connectionStringName = "CinemaEntities";
       catch (Exception ex)
                                                                                   Configuration config =
         MessageBox.Show(ex.Message, "Ошибка",
                                                                          Configuration Manager. Open Exe Configuration (Configuration User Level.\\
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                         None);
                                                                                   ConnectionStringSettings settings =
       }
                                                                          config.ConnectionStrings.ConnectionStrings[connectionStringName];
     }
    public async Task<string> LoadDataBase(string password, string
                                                                                   if (settings != null)
login, string newDataSource)
                                                                                     string currentConnectionString = settings.ConnectionString;
       try
                                                                                     string currentConnectionName = "";
         ConnectionStringChanged(newDataSource);
                                                                                     string[]\ tempData = currentConnectionString.Split(';');
                                                                                     for (int i = 0; i < tempData.Length; i++)
         using (var dataBase = new CinemaEntities())
         {
                                                                                        if (tempData[i].Contains("data source="))
            var userData = (from emploee in dataBase.Employee
                                                                                          currentConnectionName = tempData[i].Remove(0,
                     role in dataBase.Role on emploee.IDRole equals
                                                                          40);
role.IDRole
                                                                                          break;
                     where (emploee.Login == login &&
emploee.Password == password)
                     select new
                     {
```

```
string newConnectionString =
                                                                                MessageBox.Show(ex.Message, "Ошибка",
current Connection String. Replace (current Connection Name,\\
                                                                      MessageBoxButton.OK, MessageBoxImage.Error);
newDataSource);
                                                                           }
           settings.ConnectionString = newConnectionString;
           config. Save (Configuration Save Mode. Modified);\\
                                                                           private void Password_KeyDown(object sender, KeyEventArgs e)
ConfigurationManager.RefreshSection("connectionStrings");
                                                                              try
           File.WriteAllText("ConnectionData.ini", newDataSource);
                                                                               if (e.Key == Key.Enter)
         }
                                                                                  AuthorizationUser();
       catch (Exception ex)
         MessageBox.Show(ex.Message, "Ошибка",
                                                                              catch (Exception ex)
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                MessageBox.Show(ex.Message, "Ошибка",
       }
                                                                      MessageBoxButton.OK, MessageBoxImage.Error);
     }
    private void Enter_Click(object sender, RoutedEventArgs e)
                                                                           }
       AuthorizationUser();
                                                                           private void Window_Closing(object sender,
                                                                      System.ComponentModel.CancelEventArgs e)
                                                                           {
     private void DataBaseConnection_KeyDown(object sender,
                                                                             try
KeyEventArgs e)
    {
                                                                                Application.Current.Shutdown();
       try
                                                                              catch (Exception ex)
         if (e.Key == Key.Enter)
                                                                                MessageBox.Show(ex.Message, "Ошибка",
           Login.Focus();
                                                                      MessageBoxButton.OK, MessageBoxImage.Error);
       catch (Exception ex)
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                       BookerPanel.xaml
                                                                       <Window x:Class="Cinema.BookerPanel"
     }
                                                                       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                           xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    private void Login_KeyDown(object sender, KeyEventArgs e)
                                                                           xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
     {
                                                                           xmlns:mc="http://schemas.openxmlformats.org/markup-
       try
                                                                      compatibility/2006"
         if (e.Key == Key.Enter)
                                                                           xmlns:local="clr-namespace:Cinema"
                                                                           mc:Ignorable="d"
           Password.Focus();
                                                                           Title="Панель менеджера" Height="550" Width="1050"
                                                                       MinHeight="550" MinWidth="1050" Loaded="Window_Loaded"
                                                                       Closing="Window_Closing" WindowStartupLocation="CenterScreen">
       catch (Exception ex)
                                                                         <Grid Style="{DynamicResource MainColorStyle}">
                                                                           <Grid VerticalAlignment="Top">
```

```
<WrapPanel Style="{DynamicResource
SecondColorWrapPanelStyle}">
                                                                                 PageManager.Navigate(new MovieControls());
         <Button x:Name="MoviePage" Content="Фильмы"
Style="{DynamicResource TabOffButtonStyle}"
                                                                                currentSelectedButton = MoviePage;
Click="SelectedPage_Click"/>
                                                                                current Selected Button. Style = \\
         <Button x:Name="SessionPage" Content="Сеансы"
                                                                       (Style) Application. Current. Resources ["TabOnButtonStyle"];\\
Style="{DynamicResource TabOffButtonStyle}"
Click="SelectedPage_Click"/>
                                                                              catch (Exception ex)
         <Button x:Name="TicketPage" Content="Продажи"
Style="{DynamicResource TabOffButtonStyle}"
                                                                                MessageBox.Show(ex.Message, "Ошибка",
Click="SelectedPage_Click"/>
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
       </WrapPanel>
                                                                              }
       <WrapPanel HorizontalAlignment="Right">
         <Button x:Name="Help" Content="Справка" Width="100"
Style="{DynamicResource TabHelpButtonStyle}"
                                                                            private void SelectedPage_Click(object sender, RoutedEventArgs
Click="Help_Click"/>
                                                                       e)
         <Button x:Name="Exit" Content="Выйти" Width="100"
Style="{DynamicResource TabExitButtonStyle}" Click="Exit_Click"/>
                                                                              try
       </WrapPanel>
    </Grid>
                                                                                if (currentSelectedButton != null)
    <Frame x:Name="PageManager" Content="MainFrame"</p>
Margin="0,25,0,0" NavigationUIVisibility="Hidden"
                                                                                   currentSelectedButton.Style =
Background="White"/>
                                                                       (Style) Application. Current. Resources ["TabOffButtonStyle"];\\
  </Grid>
</Window>
                                                                                 currentSelectedButton = sender as Button;
BookerPanel.xaml.cs
using System;
                                                                                currentSelectedButton.Style =
                                                                       (Style) Application. Current. Resources ["TabOnButtonStyle"];\\
using System.Diagnostics;
using System.IO;
                                                                                 switch (currentSelectedButton.Name)
using System. Windows;
using System.Windows.Controls;
                                                                                 {
using System.Windows.Media;
                                                                                   case "MoviePage":
using static Cinema. Authorization;
                                                                                     PageManager.Navigate(new MovieControls());
                                                                                     break;
namespace Cinema
                                                                                   case "SessionPage":
{
  /// <summary>
                                                                                     PageManager.Navigate(new SessionControls());
  /// Логика взаимодействия для BookerPanel.xaml
                                                                                     break;
  /// </summary>
                                                                                   case "TicketPage":
  public partial class BookerPanel: Window
                                                                                     Page Manager. Navigate (new\ Ticket Analysis ());
    public BookerPanel()
                                                                                     break;
       InitializeComponent();
                                                                              catch (Exception ex)
    }
    public bool exitMode;
                                                                                MessageBox.Show(ex.Message, "Ошибка",
    public Button currentSelectedButton;
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
    private void Window_Loaded(object sender, RoutedEventArgs e)
                                                                            private void Exit_Click(object sender, RoutedEventArgs e)
       try
```

```
{
                                                                            }
       try
                                                                            private void Window_Closing(object sender,
                                                                       System.ComponentModel.CancelEventArgs e)
         exitMode = true;
         TransmittedData.idEmployee = 0;
                                                                              try
         TransmittedData.idSelectedMovie = 0;
         TransmittedData.idSelectedGenre = 0;
                                                                                if (!exitMode)
         TransmittedData.idSelectedActor = 0;
         TransmittedData.idSelectedSession = 0;
                                                                                   Application.Current.Shutdown();
         TransmittedData.idSelectedEmployee = 0;
         Authorization authorization = new Authorization();
                                                                              catch (Exception ex)
         authorization.Show();
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
         this.Close();
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
                                                                              }
       catch (Exception ex)
                                                                            }
         MessageBox.Show(ex.Message, "Ошибка",
                                                                            private void Help_Click(object sender, RoutedEventArgs e)
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                              try
                                                                                 Uri resourceUri = new
    public void TicketPage_Click(object sender, RoutedEventArgs e)
                                                                       Uri("pack://application:,,,/Resource/HelpDocument.pdf",
                                                                        UriKind.Absolute);
       TicketAnalysisOpen();
                                                                                 Stream resourceStream =
                                                                        Application.GetResourceStream(resourceUri)?.Stream;
                                                                                 string tempFileName = Path.GetTempFileName() + ".pdf";
    public void TicketAnalysisOpen()
                                                                                 using (FileStream fileStream = new
                                                                       FileStream(tempFileName, FileMode.Create, FileAccess.Write))
       try
         Page Manager. Navigate (new\ Ticket Analysis ());
                                                                                   resourceStream.CopyTo(fileStream);
       catch (Exception ex)
                                                                                 Process.Start(new ProcessStartInfo(tempFileName) {
         MessageBox.Show(ex.Message, "Ошибка",
                                                                       UseShellExecute = true });
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                              catch (Exception ex)
     }
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
    public void MovieAnalysisOpen()
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
                                                                            }
       try
         PageManager.Navigate(new MovieAnalysis());
       catch (Exception ex)
                                                                        CashierPanel.xaml
                                                                        <Window x:Class="Cinema.CashierPanel"
         MessageBox.Show(ex.Message, "Ошибка",
                                                                        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                                                                        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
```

```
xmlns:mc="http://schemas.openxmlformats.org/markup-
                                                                           public CashierPanel()
compatibility/2006"
    xmlns:local="clr-namespace:Cinema"
                                                                              InitializeComponent();
    mc:Ignorable="d"
    Title="Панель кассира" Height="550" Width="1050"
MinHeight="550" MinWidth="1050" Loaded="Window_Loaded"
                                                                           public bool exitMode;
Closing="Window_Closing" WindowStartupLocation="CenterScreen">
                                                                           public Button currentSelectedButton;
                                                                           public string currentSelectedPage;
  <Grid Style="{DynamicResource MainColorStyle}">
    <Grid VerticalAlignment="Top">
                                                                           private void Window_Loaded(object sender, RoutedEventArgs e)
       <WrapPanel Style="{DynamicResource
SecondColorWrapPanelStyle}">
                                                                              try
         <Button x:Name="MainPage" Content="Главная страница"
                                                                              {
Style="{DynamicResource TabOffButtonStyle}"
                                                                                MoviePageOpen();
Click="SelectedPage_Click"/>
         <Button x:Name="CurrentMoviesPage" Content="Фильмы
                                                                                currentSelectedButton = MainPage;
сегодня" Style="{DynamicResource TabOffButtonStyle}"
                                                                                current Selected Button. Style = \\
                                                                       (Style) Application. Current. Resources ["TabOnButtonStyle"];\\
Click="SelectedPage_Click"/>
         <Button x:Name="GoBack" Content="Назад"
Style="{DynamicResource TabOffButtonStyle}"
                                                                              catch (Exception ex)
Click="SelectedBackPage_Click"/>
       </WrapPanel>
                                                                                MessageBox.Show(ex.Message, "Ошибка",
       <WrapPanel HorizontalAlignment="Right">
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
         <Button x:Name="Help" Content="Справка" Width="100"
Style="{DynamicResource TabHelpButtonStyle}"
Click="Help_Click"/>
         <Button x:Name="Exit" Content="Выйти" Width="100"
                                                                           private void SelectedPage_Click(object sender, RoutedEventArgs
Style="{DynamicResource TabExitButtonStyle}" Click="Exit_Click"/>
                                                                       e)
       </WrapPanel>
    </Grid>
                                                                              try
    <Frame x:Name="PageManager" Content="MainFrame"</p>
Margin="0,25,0,0" NavigationUIVisibility="Hidden"
                                                                                if (currentSelectedButton != null)
Background="White"/>
  </Grid>
                                                                                  currentSelectedButton.Style = \\
</Window>
                                                                       (Style)Application.Current.Resources["TabOffButtonStyle"];
CashierPanel.xaml.cs
using Cinema.Pages;
                                                                                currentSelectedButton = sender as Button;
using System;
using System.Diagnostics;
                                                                                current Selected Button. Style = \\
using System.IO;
                                                                       (Style) Application. Current. Resources ["TabOnButtonStyle"];\\
using System.Windows;
using System. Windows. Controls;
                                                                                switch (currentSelectedButton.Name)
using System.Windows.Media;
                                                                                  case "CurrentMoviesPage":
using static Cinema. Authorization;
                                                                                     CurrentMoviesPageOpen();
namespace Cinema
                                                                                     break;
{
                                                                                  case "MainPage":
  /// <summarv>
  /// Логика взаимодействия для CashierPanel.xaml
                                                                                     MoviePageOpen();
  /// </summary>
                                                                                     break;
  public partial class CashierPanel: Window
```

```
catch (Exception ex)
                                                                              try
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                 PageManager.Navigate(new MovieCashierControls());
                                                                                 GoBack.Style =
     }
                                                                        (Style) Application. Current. Resources ["TabOffButtonStyle"]; \\
    private void SelectedBackPage_Click(object sender,
                                                                                 MainPage.Style =
RoutedEventArgs e)
                                                                        (Style)Application.Current.Resources["TabOnButtonStyle"];
                                                                                 currentSelectedButton = MainPage;
                                                                                 GoBack.Content = "";
       try
         Page currentPage = PageManager.Content as Page;
                                                                              catch (Exception ex)
         if (currentPage.GetType().Name == "PlacesCashierControls")
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
           SessionPageOpen();
                                                                              }
                                                                            }
         else
           if (currentPage.GetType().Name ==
                                                                            public void SessionPageOpen()
"SessionCashierControls")
         {
                                                                              try
           MoviePageOpen();
                                                                                 PageManager.Navigate(new SessionCashierControls());
         }
       catch (Exception ex)
                                                                                 MainPage.Style =
                                                                        (Style) Application. Current. Resources ["TabOffButtonStyle"]; \\
         MessageBox.Show(ex.Message, "Ошибка",
                                                                                 CurrentMoviesPage.Style =
MessageBoxButton.OK,\,MessageBoxImage.Error);
                                                                        (Style)Application.Current.Resources["TabOffButtonStyle"];
                                                                                 GoBack.Style =
                                                                        (Style)Application.Current.Resources["TabOnButtonStyle"];
     }
                                                                                 currentSelectedButton = GoBack;
    public void CurrentMoviesPageOpen()
                                                                                 GoBack.Content = "Фильмы";
                                                                              catch (Exception ex)
       try
         PageManager.Navigate(new CurrentMovieCashierControls());
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
         GoBack.Style =
(Style)Application.Current.Resources["TabOffButtonStyle"];
                                                                            }
         Current Movies Page. Style = \\
(Style) Application. Current. Resources ["TabOnButtonStyle"];\\
                                                                            public void PlacesPageOpen()
         currentSelectedButton = CurrentMoviesPage;
         GoBack.Content = "";
                                                                              try
       catch (Exception ex)
                                                                                 PageManager.Navigate(new PlacesCashierControls());
                                                                                 GoBack.Content = "Сеансы";
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                              catch (Exception ex)
                                                                                 MessageBox.Show(ex.Message, "Ошибка",
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
    public void MoviePageOpen()
```

```
}
                                                                                  resourceStream.CopyTo(fileStream);
    private void Exit_Click(object sender, RoutedEventArgs e)
                                                                               Process.Start(new ProcessStartInfo(tempFileName) {
      try
                                                                      UseShellExecute = true });
         exitMode = true:
                                                                             catch (Exception ex)
        TransmittedData.idSelectedCashierMovie = 0;
         TransmittedData.idSelectedCashierSession = 0;
                                                                               MessageBox.Show(ex.Message, "Ошибка",
                                                                      MessageBoxButton.OK, MessageBoxImage.Error);
         Authorization authorization = new Authorization();
         authorization.Show();
                                                                           }
         this.Close();
       catch (Exception ex)
                                                                      DirectorsPanel.xaml
                                                                      <Window x:Class="Cinema.DirectorsPanel"
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                                                                           xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    }
                                                                           xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
                                                                           xmlns:mc="http://schemas.openxmlformats.org/markup-
    private void Window_Closing(object sender,
                                                                      compatibility/2006"
System.ComponentModel.CancelEventArgs e)
                                                                           xmlns:local="clr-namespace:Cinema"
                                                                           mc:Ignorable="d"
                                                                           Title="Панель директора" Height="550" Width="1050"
       try
                                                                      MinHeight="550" MinWidth="1050" Loaded="Window_Loaded"
                                                                      Closing="Window_Closing" WindowStartupLocation="CenterScreen">
         if (!exitMode)
           Application.Current.Shutdown();
                                                                        <Grid Style="{DynamicResource MainColorStyle}">
                                                                           <Grid VerticalAlignment="Top">
                                                                             <WrapPanel Style="{DynamicResource
       catch (Exception ex)
                                                                      SecondColorWrapPanelStyle}">
                                                                               <Button x:Name="EmployeePage" Content="Сотрудники"
         MessageBox.Show(ex.Message, "Ошибка",
                                                                      Style="{DynamicResource TabOffButtonStyle}"
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                      Click="SelectedPage_Click"/>
       }
                                                                               <Button x:Name="TicketPage" Content="Продажи"
                                                                      Style="{DynamicResource TabOffButtonStyle}"
    }
                                                                      Click="SelectedPage_Click"/>
                                                                             </WrapPanel>
    private void Help_Click(object sender, RoutedEventArgs e)
                                                                             <WrapPanel HorizontalAlignment="Right">
                                                                               <Button x:Name="Help" Content="Справка" Width="100"
       try
                                                                      Style="{DynamicResource TabHelpButtonStyle}"
         Uri resourceUri = new
                                                                      Click="Help_Click"/>
Uri("pack://application:,,,/Resource/HelpDocument.pdf",
                                                                               <Button x:Name="Exit" Content="Выйти" Width="100"
UriKind.Absolute);
                                                                      Style="{DynamicResource TabExitButtonStyle}" Click="Exit_Click"/>
                                                                             </WrapPanel>
         Stream resourceStream =
                                                                           </Grid>
Application. Get Resource Stream (resource Uri)?. Stream;\\
         string tempFileName = Path.GetTempFileName() + ".pdf";
                                                                           <Frame x:Name="PageManager" Content="MainFrame"</pre>
                                                                      Margin="0,25,0,0" NavigationUIVisibility="Hidden"
         using (FileStream fileStream = new
FileStream(tempFileName, FileMode.Create, FileAccess.Write))
                                                                      Background="White"/>
```

```
</Grid>
</Window>
                                                                                    currentSelectedButton = sender as Button;
DirectorsPanel.xaml.cs
using System;
                                                                                   current Selected Button. Style = \\
using System.Diagnostics;
                                                                          (Style) Application. Current. Resources ["TabOnButtonStyle"];\\
using System.IO;
using System.Windows;
                                                                                    switch (currentSelectedButton.Name)
using System. Windows. Controls;
                                                                                    {
using System.Windows.Media;
                                                                                      case "EmployeePage":
using static Cinema. Authorization;
                                                                                        PageManager.Navigate(new EmployeeControls());
                                                                                         break;
namespace Cinema
{
                                                                                      case "TicketPage":
  /// <summary>
                                                                                        TicketAnalysisOpen();
  /// Логика взаимодействия для DirectorsPanel.xaml
                                                                                        break;
  /// </summary>
  public partial class DirectorsPanel: Window
                                                                                 catch (Exception ex)
     public DirectorsPanel()
                                                                                    MessageBox.Show(ex.Message, "Ошибка",
                                                                          MessageBoxButton.OK, MessageBoxImage.Error);
       InitializeComponent();
                                                                               }
     public bool exitMode;
     public Button currentSelectedButton;
                                                                               private void Exit_Click(object sender, RoutedEventArgs e)
                                                                               {
     private void Window_Loaded(object sender, RoutedEventArgs e)
                                                                                 try
                                                                                   exitMode = true;
       try
         PageManager.Navigate(new EmployeeControls());
                                                                                   TransmittedData.idEmployee = 0;
                                                                                    TransmittedData.idSelectedMovie = 0;
         current Selected Button = Employee Page; \\
                                                                                    TransmittedData.idSelectedGenre = 0;
         current Selected Button. Style = \\
                                                                                    Transmitted Data.id Selected Actor = 0;\\
(Style) Application. Current. Resources ["TabOnButtonStyle"];\\
                                                                                    TransmittedData.idSelectedSession = 0;
                                                                                    TransmittedData.idSelectedEmployee = 0;
       catch (Exception ex)
                                                                                    Authorization authorization = new Authorization();
          MessageBox.Show(ex.Message, "Ошибка",
                                                                                    authorization.Show();
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                    this.Close();
     }
                                                                                 catch (Exception ex)
    private void SelectedPage_Click(object sender, RoutedEventArgs
e)
                                                                                    MessageBox.Show(ex.Message, "Ошибка",
                                                                          MessageBoxButton.OK, MessageBoxImage.Error);
     {
       try
         if (currentSelectedButton != null)
                                                                               public void TicketAnalysisOpen()
            current Selected Button. Style = \\
(Style) Application. Current. Resources ["TabOffButtonStyle"]; \\
                                                                                 try
```

```
PageManager.Navigate(new TicketAnalysis());
                                                                                  resourceStream.CopyTo(fileStream);
       catch (Exception ex)
                                                                                Process.Start(new ProcessStartInfo(tempFileName) {
                                                                       UseShellExecute = true });
         MessageBox.Show(ex.Message, "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                              catch (Exception ex)
     }
                                                                                MessageBox.Show(ex.Message, "Ошибка",
                                                                       MessageBoxButton.OK, MessageBoxImage.Error);
    public void MovieAnalysisOpen()
       try
                                                                           }
         PageManager.Navigate(new MovieAnalysis());
       catch (Exception ex)
                                                                       Light.xaml
                                                                       < Resource Dictionary
         MessageBox.Show(ex.Message, "Ошибка",
                                                                       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                                  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
     }
                                                                         <Style x:Key="MainColorStyle" TargetType="Grid">
                                                                            <Setter Property="Background" Value="White"/>
    private void Window_Closing(object sender,
                                                                         </Style>
System.ComponentModel.CancelEventArgs e)
                                                                          <Style x:Key="SecondColorStyle" TargetType="Grid">
                                                                           <Setter Property="Background" Value="Aqua"/>
                                                                           <Setter Property="Height" Value="35"/>
       try
                                                                          </Style>
                                                                          <Style x:Key="SecondColorRectangleStyle"
         if (!exitMode)
                                                                       TargetType="Rectangle">
                                                                           <Setter Property="Fill" Value="Aqua"/>
            Application.Current.Shutdown();
                                                                           <Setter Property="Height" Value="35"/>
                                                                         </Style>
       catch (Exception ex)
                                                                          <Style x:Key="SecondColorRectanglePlaceStyle"
                                                                       TargetType="Rectangle">
         MessageBox.Show(ex.Message, "Ошибка",
                                                                           <Setter Property="Fill" Value="Aqua"/>
MessageBoxButton.OK, MessageBoxImage.Error);
                                                                           <Setter Property="Opacity" Value="0.5"/>
       }
                                                                          </Style>
                                                                         <Style x:Key="SecondColorSeparatorsRectangleStyle"
     }
                                                                       TargetType="Rectangle">
                                                                           <Setter Property="Fill" Value="Black"/>
    private void Help_Click(object sender, RoutedEventArgs e)
                                                                           <Setter Property="Height" Value="25"/>
                                                                           <Setter Property="Width" Value="2"/>
       try
                                                                           <Setter Property="Opacity" Value="0.2"/>
         Uri resourceUri = new
                                                                          </Style>
Uri("pack://application:,,,/Resource/HelpDocument.pdf",
                                                                          <Style x:Key="SecondColorWrapPanelStyle"
UriKind.Absolute);
                                                                       TargetType="WrapPanel">
                                                                            <Setter Property="Background" Value="#00dbdb"/>
         Stream resourceStream =
                                                                            <Setter Property="Height" Value="25"/>
Application.GetResourceStream(resourceUri)?.Stream;
                                                                          </Style>
         string tempFileName = Path.GetTempFileName() + ".pdf";
         using (FileStream fileStream = new
                                                                         <Style x:Key="TabOnButtonStyle" TargetType="Button">
FileStream(tempFileName, FileMode.Create, FileAccess.Write))
                                                                            <Setter Property="Focusable" Value="False"/>
```

```
<Setter Property="IsTabStop" Value="False"/>
                                                                          </Style>
    <Setter Property="Width" Value="150"/>
                                                                          <Style x:Key="TabExitButtonStyle" TargetType="Button">
    <Setter Property="Height" Value="25"/>
                                                                            <Setter Property="Focusable" Value="False"/>
    <Setter Property="FontSize" Value="16"/>
                                                                            <Setter Property="IsTabStop" Value="False"/>
    <Setter Property="Foreground" Value="Black"/>
                                                                            <Setter Property="Width" Value="150"/>
    <Setter Property="Template">
                                                                            <Setter Property="Height" Value="25"/>
       <Setter.Value>
                                                                            <Setter Property="FontSize" Value="16"/>
         <ControlTemplate TargetType="{x:Type Button}">
                                                                            <Setter Property="Foreground" Value="Black"/>
           <Border Background="Aqua" CornerRadius="5,5,0,0">
                                                                            <Setter Property="Template">
              <ContentPresenter HorizontalAlignment="Center"</p>
                                                                               <Setter.Value>
VerticalAlignment="Center"/>
                                                                                 <ControlTemplate TargetType="{x:Type Button}">
            </Border>
                                                                                   <Border Background="Aqua" CornerRadius="0,5,0,0"
         </ControlTemplate>
                                                                       BorderThickness="1,0,0,0" BorderBrush="#00dbdb">
       </Setter.Value>
                                                                                     <ContentPresenter HorizontalAlignment="Center"</p>
    </Setter>
                                                                        VerticalAlignment="Center"/>
  </Style>
                                                                                   </Border>
  <Style x:Key="TabOffButtonStyle" TargetType="Button">
                                                                                 </ControlTemplate>
    <Setter Property="Focusable" Value="False"/>
                                                                               </Setter.Value>
    <Setter Property="IsTabStop" Value="False"/>
                                                                            </Setter>
    <Setter Property="Width" Value="150"/>
                                                                          </Style>
    <Setter Property="Height" Value="25"/>
                                                                          <Style x:Key="TabControlButtonStyle" TargetType="Button">
    <Setter Property="FontSize" Value="16"/>
                                                                            <Setter Property="FontSize" Value="16"/>
    <Setter Property="Foreground" Value="Black"/>
                                                                            <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Template">
                                                                            <Setter Property="Template">
       <Setter.Value>
                                                                               <Setter.Value>
         <ControlTemplate TargetType="{x:Type Button}">
                                                                                 <ControlTemplate TargetType="{x:Type Button}">
                                                                                   <Border Background="White" CornerRadius="2"
           <Border Background="Transparent"
CornerRadius="5,5,0,0" Opacity="0.6">
                                                                       BorderThickness="0" BorderBrush="Black">
                                                                                     < ContentPresenter HorizontalAlignment="Center"
              < ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                                                                       VerticalAlignment="Center"/>
           </Border>
                                                                                   </Border>
                                                                                 </ControlTemplate>
         </ControlTemplate>
       </Setter.Value>
                                                                               </Setter.Value>
    </Setter>
                                                                            </Setter>
  </Style>
                                                                          </Style>
  <Style x:Key="TabHelpButtonStyle" TargetType="Button">
                                                                          <Style x:Key="MainButtonStyle" TargetType="Button">
    <Setter Property="Focusable" Value="False"/>
                                                                            <Setter Property="FontSize" Value="16"/>
    <Setter Property="IsTabStop" Value="False"/>
                                                                            <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Width" Value="150"/>
                                                                            <Setter Property="Template">
    <Setter Property="Height" Value="25"/>
                                                                               <Setter Value>
    <Setter Property="FontSize" Value="16"/>
                                                                                 <ControlTemplate TargetType="{x:Type Button}">
                                                                                   <Border Background="White" CornerRadius="2"
    <Setter Property="Foreground" Value="Black"/>
                                                                       BorderThickness="1" BorderBrush="Black">
    <Setter Property="Template">
                                                                                     < ContentPresenter HorizontalAlignment="Center"
       <Setter.Value>
         <ControlTemplate TargetType="{x:Type Button}">
                                                                       VerticalAlignment="Center"/>
           <Border Background="Aqua" CornerRadius="5,0,0,0"
                                                                                   </Border>
BorderThickness="0,0,1,0" BorderBrush="#00dbdb">
                                                                                 </ControlTemplate>
              <ContentPresenter HorizontalAlignment="Center"</p>
                                                                              </Setter.Value>
VerticalAlignment="Center"/>
                                                                            </Setter>
           </Border>
                                                                          </Style>
         </ControlTemplate>
                                                                          <Style x:Key="MainClearButtonStyle" TargetType="Button"
       </Setter.Value>
                                                                       BasedOn="{StaticResource MainButtonStyle}">
                                                                            <Setter Property="Template">
    </Setter>
```

```
<Setter.Value>
                                                                                        <ControlTemplate>
         <ControlTemplate TargetType="{x:Type Button}">
                                                                                          <Grid>
           <Border Background="White">
                                                                                            <Grid.ColumnDefinitions>
                                                                                              <ColumnDefinition Width="5*"/>
             <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                                                                                              <ColumnDefinition Width="*"/>
           </Border>
                                                                                            </Grid.ColumnDefinitions>
         </ControlTemplate>
                                                                                            <Border x:Name="Border"
       </Setter.Value>
                                                                      Grid.ColumnSpan="2" CornerRadius="0" Background="Transparent"
    </Setter>
                                                                      BorderBrush="Black" BorderThickness="0,0,0,1" />
  </Style>
                                                                                            <Path x:Name="Arrow" Grid.Column="1"
                                                                     Fill="Black" HorizontalAlignment="Center"
  <Style x:Key="TitleTextBlockStyle" TargetType="TextBlock">
                                                                      VerticalAlignment="Center" Data="M 0 0 L 4 4 L 8 0 Z"/>
    <Setter Property="VerticalAlignment" Value="Center"/>
                                                                                          </Grid>
    <Setter Property="HorizontalAlignment" Value="Center"/>
                                                                                          <ControlTemplate.Triggers>
    <Setter Property="FontSize" Value="20"/>
                                                                                            <Trigger
    <Setter Property="FontWeight" Value="Bold"/>
                                                                     Property="ToggleButton.IsMouseOver" Value="true">
    <Setter Property="Foreground" Value="Black"/>
                                                                                              <Setter TargetName="Border"
                                                                     Property="Background" Value="Aqua" />
    <Setter Property="Background" Value="Transparent"/>
    <Setter Property="TextAlignment" Value="Center"/>
                                                                                            </Trigger>
                                                                                            <Trigger Property="ToggleButton.IsChecked"
  </Style>
  <Style x:Key="MainTextBlockStyle" TargetType="TextBlock">
                                                                      Value="true">
    <Setter Property="FontSize" Value="16"/>
                                                                                              <Setter TargetName="Border"
                                                                     Property="Background" Value="Transparent" />
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="Transparent"/>
                                                                                            </Trigger>
  </Style>
                                                                                          </ControlTemplate.Triggers>
  <Style x:Key="MainLableStyle" TargetType="Label">
                                                                                        </ControlTemplate>
    <Setter Property="FontSize" Value="16"/>
                                                                                     </ToggleButton.Template>
    <Setter Property="Foreground" Value="Black"/>
                                                                                   </ToggleButton>
                                                                                   <ContentPresenter Name="ContentSite"
  </Style>
                                                                     IsHitTestVisible="False" Content="{TemplateBinding
  <Style x:Key="MainTextBoxStyle" TargetType="TextBox">
                                                                      SelectionBoxItem}" ContentTemplate="{TemplateBinding
    <Setter Property="FontSize" Value="16"/>
                                                                      SelectionBoxItemTemplate}"
    <Setter Property="Foreground" Value="Black"/>
                                                                      ContentTemplateSelector="{TemplateBinding ItemTemplateSelector}"
    <Setter Property="Background" Value="Transparent"/>
                                                                      Margin="3" />
    <Setter Property="BorderThickness" Value="0,0,0,1"/>
                                                                                   <TextBox x:Name="PART_EditableTextBox"
    <Setter Property="BorderBrush" Value="Black"/>
                                                                      Visibility="Hidden" IsReadOnly="{TemplateBinding IsReadOnly}"/>
  </Style>
                                                                                   <Popup Name="Popup" Placement="Bottom"
                                                                      IsOpen="{TemplateBinding IsDropDownOpen}"
                                                                      AllowsTransparency="True" Focusable="False"
  <Style x:Key="MainComboBoxStyle" TargetType="ComboBox">
    <Setter Property="FontSize" Value="16"/>
                                                                      PopupAnimation="Slide">
    <Setter Property="Foreground" Value="Black"/>
                                                                                     <Grid Name="DropDown"
    <Setter Property="SelectedIndex" Value="0"/>
                                                                      SnapsToDevicePixels="True" MinWidth="{TemplateBinding
    <Setter Property="BorderThickness" Value="0,0,0,1"/>
                                                                      ActualWidth}" MaxHeight="{TemplateBinding
    <Setter Property="BorderBrush" Value="Black"/>
                                                                     MaxDropDownHeight}">
    <Setter Property="Template">
                                                                                        <Border x:Name="DropDownBorder"
       <Setter.Value>
                                                                      Background="White" BorderBrush="Black" BorderThickness="1"/>
         <ControlTemplate TargetType="ComboBox">
                                                                                        <ScrollViewer Margin="1"
                                                                      SnapsToDevicePixels="True">
             <ToggleButton Grid.Column="2" Focusable="false"
                                                                                          <StackPanel IsItemsHost="True" />
IsChecked="{Binding
                                                                                        </ScrollViewer>
Path=IsDropDownOpen,Mode=TwoWay,RelativeSource={RelativeSou
                                                                                     </Grid>
rce TemplatedParent}}" >
                                                                                   </Popup>
                <ToggleButton.Template>
                                                                                 </Grid>
```

```
</ControlTemplate>
                                                                            <Setter Property="IsIndeterminate" Value="True"/>
       </Setter.Value>
                                                                           <Setter Property="Foreground" Value="Aqua"/>
    </Setter>
                                                                            <Setter Property="BorderBrush" Value="Transparent"/>
  </Style>
                                                                         </Style>
  <Style x:Key="MainSelectedComboBoxStyle" TargetType="{x:Type
ComboBoxItem}">
                                                                         <Style x:Key="MainCheckBoxStyle" TargetType="CheckBox">
    <Setter Property="Template">
                                                                         </Style>
       <Setter.Value>
                                                                       </ResourceDictionary>
         <ControlTemplate TargetType="{x:Type ComboBoxItem}">
           <Border x:Name="myBorder">
                                                                       App.xaml
              <ContentPresenter />
                                                                       <Application x:Class="Cinema.App"
           </Border>
           <ControlTemplate.Triggers>
                                                                       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
              <Trigger Property="IsSelected" Value="True">
                                                                              xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                <Setter TargetName="myBorder"
                                                                              xmlns:local="clr-namespace:Cinema"
Property="Background" Value="Aqua" />
                                                                              StartupUri="Windows/Authorization.xaml">
              </Trigger>
                                                                         <Application.Resources>
              <Trigger Property="IsMouseOver" Value="True">
                                                                           <ResourceDictionary>
                <Setter TargetName="myBorder"
                                                                              <ResourceDictionary.MergedDictionaries>
Property="Background" Value="#adffff" />
                                                                                <ResourceDictionary Source="Style/Light.xaml"/>
              </Trigger>
                                                                              </ResourceDictionary.MergedDictionaries>
           </ControlTemplate.Triggers>
                                                                           </ResourceDictionary>
         </ControlTemplate>
                                                                         </Application.Resources>
       </Setter.Value>
                                                                       </Application>
    </Setter>
  </Style>
  <Style x:Key="MainDataGridStyle" TargetType="DataGrid">
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="Transparent"/>
    <Setter Property="BorderBrush" Value="Transparent"/>
    <Setter Property="BorderThickness" Value="0"/>
    <Setter Property="ColumnWidth" Value="*"/>
  </Style>
  <Style x:Key="MainDatePickerStyle" TargetType="DatePicker">
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="Transparent"/>
    <Setter Property="BorderThickness" Value="0,0,0,1"/>
    <Setter Property="BorderBrush" Value="Black"/>
  </Style>
  <Style x:Key="SecondDatePickerStyle" TargetType="DatePicker">
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="Transparent"/>
    <Setter Property="BorderBrush" Value="Transparent"/>
    <Setter Property="IsEnabled" Value="False"/>
  </Style>
  <Style x:Key="MainProgressBarStyle" TargetType="ProgressBar">
    <Setter Property="Height" Value="5"/>
```

## Приложение 2

## РУКОВОДСТВО ОПЕРАТОРА

Функциональным назначением программы является автоматизация продажи билетов в кинотеатре.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- Управление сотрудниками;
- Управление фильмами;
- Управление сеансами;
- Настройка конфигурации зала;
- Резервное копирование базы данных;
- Продажа билетов на выбранный сеанс фильма;
- Расчет количества проданных билетов на фильмы;
- Расчет количества проданных билетов на сеансы выбранного фильма;
- Сохранение отчета по количеству проданных билетов на фильмы в файл формата pdf;
- Сохранение отчета по количеству проданных билетов на сеансы выбранного фильма в файл формата pdf;
- Открытие pdf файла проданного билета для его печати или сохранения;

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий себя:

• процессор с тактовой частотой, 1,5 ГГц, не менее;

- оперативную память объемом 1 гб, не менее;
- жесткий диск со свободным местом 500 Мб, не менее;
- монитор, с разрешением экрана 1280\*720, не менее;
- компьютерная мышь;
- клавиатура;

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 7/8/10/11.

Все пользователи должны обладать навыками работы с графическим пользовательским интерфейсом операционной системы.

Для запуска программного продукта запустить исполняемый файл Cinema.exe.

После запуска программа откроется окно авторизации (Рисунок 1). Оно представлено именем базы данных (именем сервера), логином и паролем.

Авторизация	1	×
АВТОРИЗАЦИЯ		
Имя БД Логин Пароль	DESKTOP-LOL\SQLEXPRESS	
v1.0		

Рисунок 1 – Окно авторизации

Пользователь может авторизоваться по четырьмя ролями: администратор, кассир, букер(менеджер), директор.

После авторизации под ролью администратора мы видим панель администратора (Рисунок 2). Она открывается на окне "Сотрудники" здесь мы можем найти и просмотреть информацию о сотрудниках, а также добавить удалить или редактировать сотрудников.

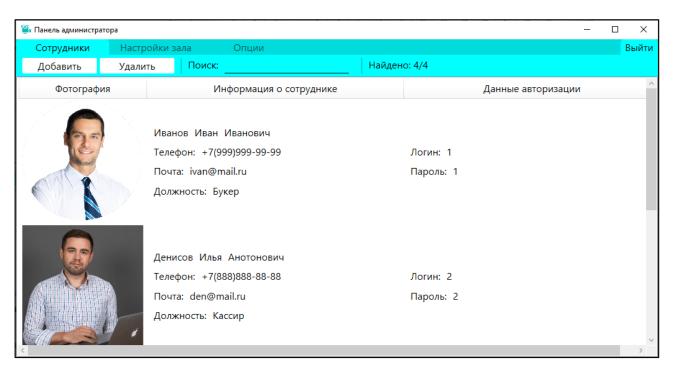


Рисунок 2 – Панель администратора с открытым окном сотрудники

Для того чтобы добавить сотрудника необходимо нажать на кнопку "Добавить" после чего откроется оно добавления сотрудник (Рисунок 3).

👺 Сотрудник		×		
сотрудник				
Фамилия:				
Имя:				
Отчество:				
Телефон:				
Эл. Почта:				
Должность:	Букер 🔻			
Логин:				
Пароль:				
Фотография:				
	Загрузить			
	Сохранить			

Рисунок 3 – Окно добавления сотрудника

После добавления данных о сотруднике приложение отобразит сообщение о успешном сохранение (Рисунок 4).

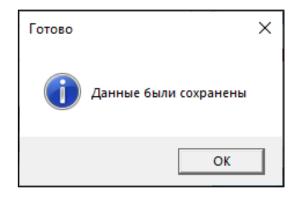


Рисунок 4 – Сообщение о успешном сохранение данных

Для редактирования данных сотрудника необходимо дважды нажать на редактируемого сотрудника, после чего откроется окно редактирования сотрудника (Рисунок 3).

Для удаления сотрудника необходимо выбрать сотрудника и нажать на кнопку удалить после чего отроиться сообщение-подтверждение удаления (Рисунок 4).

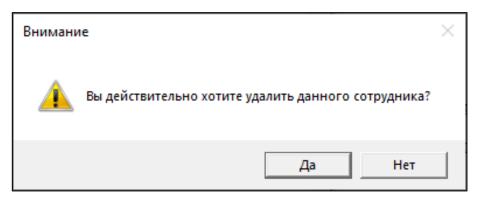


Рисунок 4 – Сообщение подтверждение удаления сотрудника

Также панель администратора обладает настройкой конфигурацией зала (Рисунок 5). В данном окне пользователь может настроить размерность зала выбрав количество рядов и мест, а также убрать лишние места в зале. После настройки зала следует сохранить конфигурацию при помощи кнопки "Сохранить".

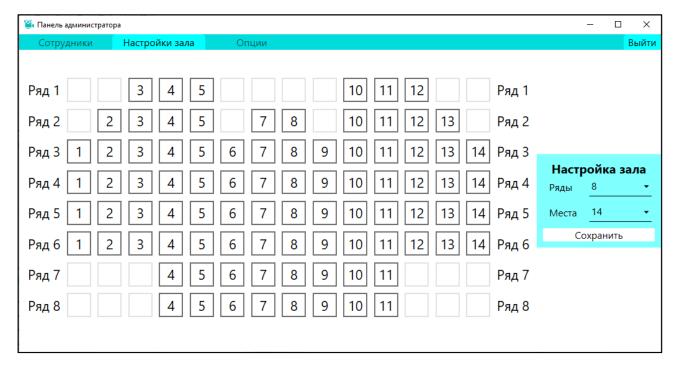


Рисунок 5 – Панель администратора с открытым окном настройки зала

После сохранения конфигурации откроется сообщение о успешном сохранение (Рисунок 6).

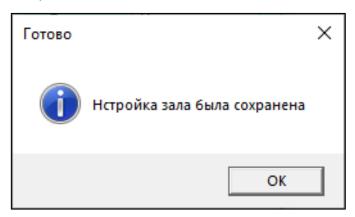


Рисунок 6 – Сообщение о успешном сохранение конфигурации

Панель администратора обладает возможностью резервного копирования, которая находиться во вкладке "Опции" (Рисунок 7). В данном окне есть 2 панели для создания полной резервной копии и восстановление из файла резервной копии.

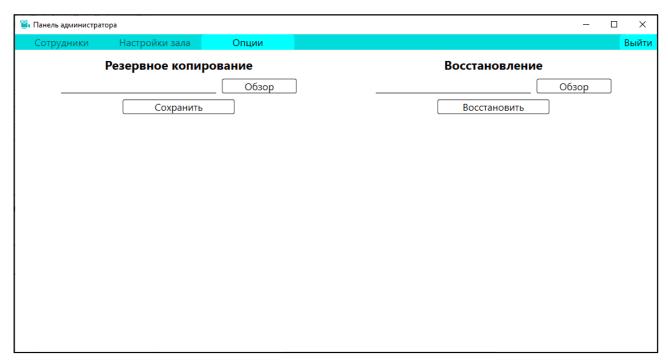


Рисунок 7 – Панель администратора с открытым окном опции

После авторизации под ролью букера (менеджера) мы видим панель менеджера (Рисунок 8). Она открывается на окне "Фильмы" здесь мы можем найти и просмотреть информацию о фильмах, а также добавить удалить или редактировать фильмы.

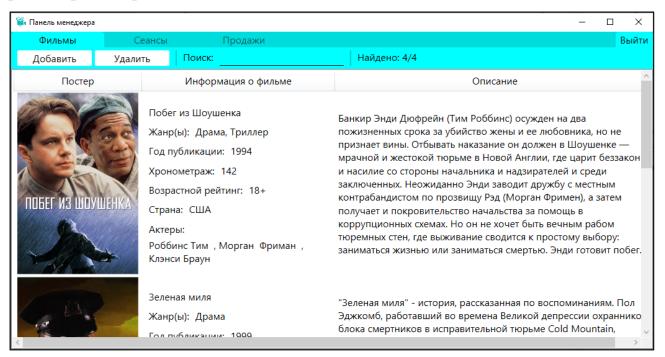


Рисунок 8 – Панель менеджера с открытым окном фильмы

Окно добавления фильма поддерживает добавление, изменение и удаление жанров для этого нажать соответствующую кнопку, а в случае редактирования и удаления необходимо предварительно выбрать жанр из списка (Рисунок 9). Для выбора жанра необходимо дважды нажать на нужный жанр.



Рисунок 9 – Окно добавления жанра

Окно добавления фильма также поддерживает добавление, изменение и удаление актеров для этого нажать соответствующую кнопку, а в случае редактирования и удаления необходимо предварительно выбрать актера из списка (Рисунок 10).

<b>АКТЕР</b> Фамилия:					
Фамилия:	AKTEP				
- CONTROL - CONT					
Имя:					
Отчество:					
Прозвище:					
Фотография:					
Загрузить					
Сохранить					

Рисунок 10 – Окно добавления жанра

Для того чтобы добавить фильм необходимо нажать на кнопку "Добавить" после чего откроется оно добавления фильма (Рисунок 11).

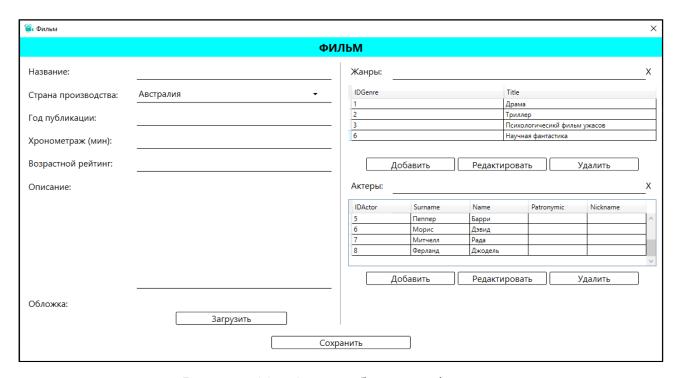


Рисунок 11 – Окно добавления фильма

После добавления данных о фильме приложение отобразит сообщение о успешном сохранение (Рисунок 12).

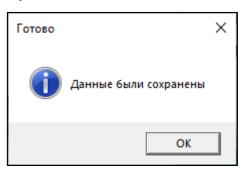


Рисунок 12 – Сообщение о успешном сохранение данных

Для редактирования данных фильма необходимо дважды нажать на редактируемый фильм, после чего откроется окно редактирования фильма (Рисунок 11).

Для удаления фильма необходимо выбрать фильм и нажать на кнопку удалить после чего отроиться сообщение-подтверждение удаления (Рисунок 13).

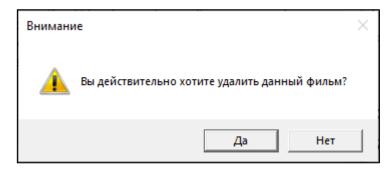


Рисунок 13 – Сообщение подтверждение удаления фильма

Также панель менеджера содержит окно сеансов (Рисунок 14). Здесь мы можем найти и просмотреть информацию о сеансах, а также добавить удалить или редактировать сеанс.

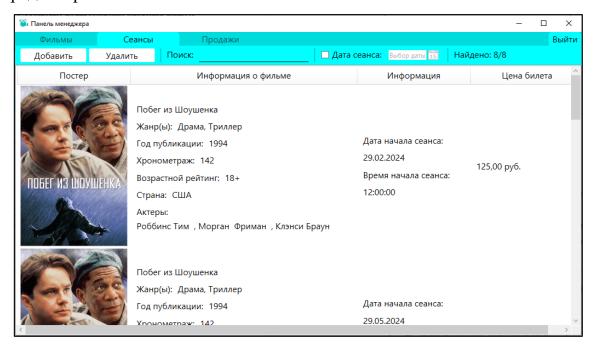


Рисунок 14 – Панель менеджера с открытым окном сеансы

Для того чтобы добавить сеанс необходимо нажать на кнопку "Добавить" после чего откроется оно добавления сеанса (Рисунок 15).



Рисунок 15 – Окно добавления сеанса

После добавления данных о сеансе приложение отобразит сообщение о успешном сохранение (Рисунок 16).

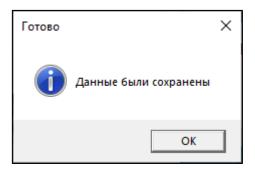


Рисунок 16 – Сообщение о успешном сохранение данных

Для редактирования данных сеанса необходимо дважды нажать на редактируемый сеанс, после чего откроется окно редактирования сеанса (Рисунок 15).

Для удаления сеанса необходимо выбрать сеанс и нажать на кнопку удалить после чего отроиться сообщение-подтверждение удаления (Рисунок 17).

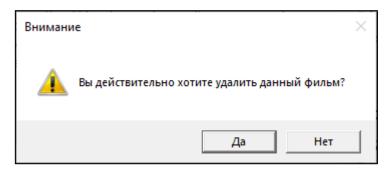


Рисунок 17 – Сообщение подтверждение удаления фильма

Также панель менеджера содержит окно продаж для просмотра и анализирования данных по проданным билетам (Рисунок 18). На данном окне можно найти и просмотреть информацию о проданных билетах на фильмы. А также пользователь может сохранить информацию в файл формата PDF.



Рисунок 18 – Панель менеджера с открытым окном продажи

Для более подробного просмотра проданных билетов пользователь может нажать на интересующий его фильм и перед ним откроется таблица продажи билетов по сеансам (Рисунок 19). На этом окне также можно найти интересующий вас сеанс, а также сохранить информацию в файл формата PDF.

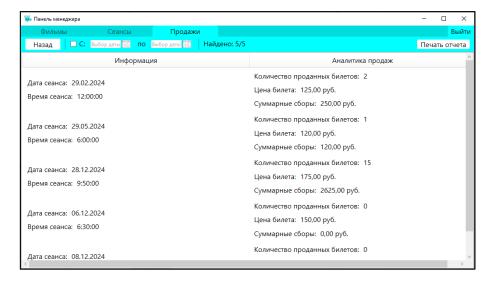


Рисунок 19 — Панель менеджера с открытым окном подробного просмотра продажи билетов

После авторизации под ролью кассира мы видим панель кассира (Рисунок 20). Она открывается на окне "Главная страница" здесь мы можем найти и просмотреть информацию о фильмах.

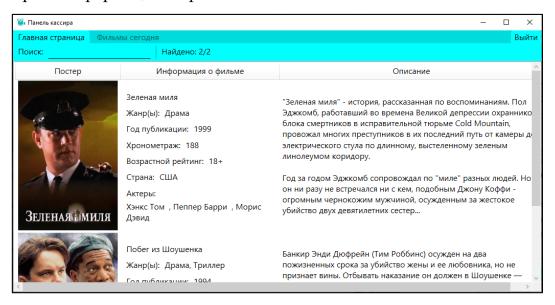


Рисунок 20 – Панель кассира с открытым окном главной страницы

На данном окне пользователь может выбрать фильм, на который нужно забронировать билет. После выбора фильма откроется окно выбора сеансов на выбранный фильм (Рисунок 21). На открытом окне пользователь может выбрать актуальные сеансы.

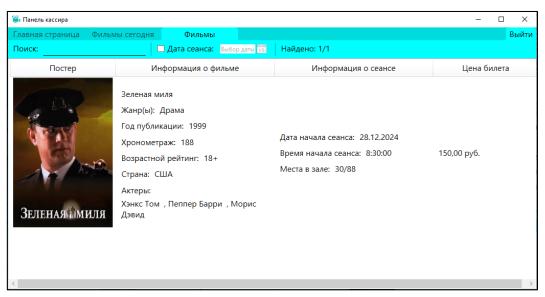


Рисунок 21 – Панель кассира с открытым окном сеансов

После выбора сеанса откроется окно для выбора места в зале (Рисунок 22). На данном окне пользователь может выбрать место в зале.

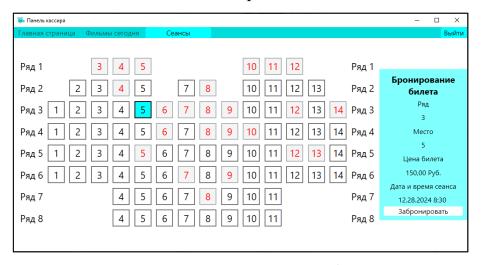


Рисунок 22 – Панель кассира с открытым окном бронирования места в зале

После подтверждения сообщения о бронирование билета (Рисунок 23), он автоматически откроется в PDF редакторе (Рисунок 24).

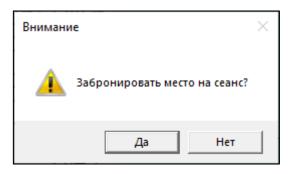


Рисунок 23 – Окно подтверждения бронирования билета



Рисунок 24 – Билет на бронируемый фильм

Для более удобной работы панель кассира обладает возможностью просмотра фильмов, которые буду проходить сегодня (Рисунок 25).

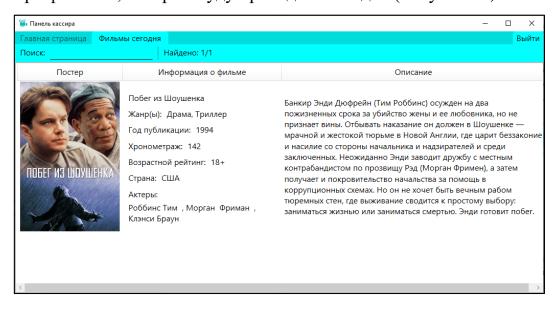


Рисунок 25 – Панель кассира с открытым окном фильмы сегодня

После авторизации под ролью директора мы видим панель директора (Рисунок 26). Она открывается на окне "Сотрудники" здесь мы можем найти и просмотреть информацию о сотрудниках, а также добавить удалить или редактировать сотрудников.

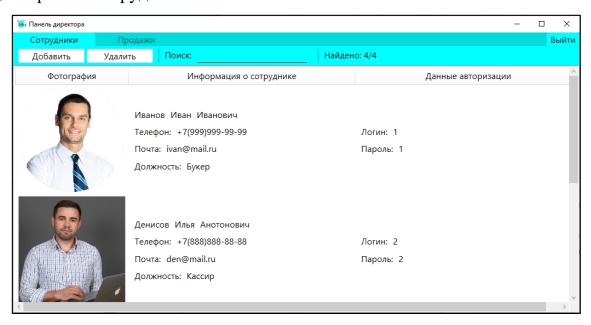


Рисунок 26 – Панель директора с открытым окном сотрудники

Для того чтобы добавить сотрудника необходимо нажать на кнопку "Добавить" после чего откроется оно добавления сотрудник (Рисунок 27).



Рисунок 27 – Окно добавления сотрудника

После добавления данных о сотруднике приложение отобразит сообщение о успешном сохранение (Рисунок 28).

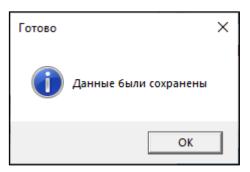


Рисунок 28 – Сообщение о успешном сохранение данных

Для редактирования данных сотрудника необходимо дважды нажать на редактируемого сотрудника, после чего откроется окно редактирования сотрудника (Рисунок 27).

Для удаления сотрудника необходимо выбрать сотрудника и нажать на кнопку удалить после чего отроиться сообщение-подтверждение удаления (Рисунок 29).

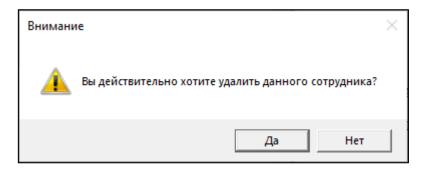


Рисунок 29 – Сообщение подтверждение удаления сотрудника

Также панель директора содержит окно продаж для просмотра и анализирования данных по проданным билетам (Рисунок 30). На данном окне можно найти и просмотреть информацию о проданных билетах на фильмы. А также пользователь может сохранить информацию в файл формата PDF.

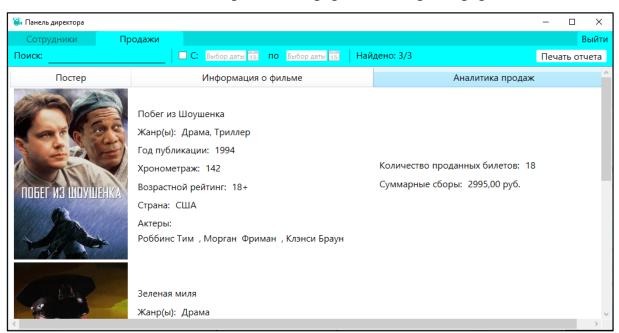


Рисунок 30 – Панель директора с открытым окном продажи

Для более подробного просмотра проданных билетов пользователь может нажать на интересующий его фильм и перед ним откроется таблица продажи билетов по сеансам (Рисунок 31). На этом окне также можно найти интересующий вас сеанс, а также сохранить информацию в файл формата PDF.

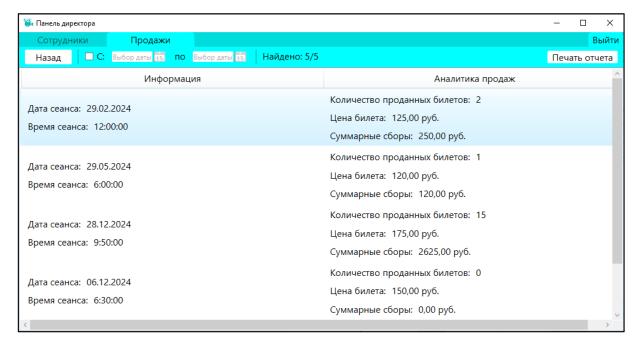


Рисунок 31 — Панель менеджера с открытым окном подробного просмотра продажи билетов

## Приложение 3

Программный продукт поставляется на USB флеш накопителе.

Ha USB накопителя находиться:

- Установочник программы
- Проект программного продукта
- Скрипт базы данных без тестовых данных
- Скрипт базы данных с тестовыми данными
- Файл курсового проекта в формате Microsoft Word (docx)
- Файл с презентацией курсового проекта

Также программный продукт можно получить, перейдя по ссылке:

- Облако Mail:
- GitHub: https://github.com/ALTERRA4546/Cinema