**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**«Чувашский государственный университет имени И.Н. Ульянова»**

**(ФГБОУ ВО «ЧГУ им. И.Н. Ульянова»)**

**Факультет информатики и вычислительной техники**

**Кафедра компьютерных технологий**

ОТЧЕТ

ОБ УЧЕБНОЙ ПРАКТИКЕ

(ОЗНАКОМИТЕЛЬНОЙ ПРАКТИКЕ)

на базе кафедры компьютерных технологий

(наименование профильной организации/ структурного подразделения университета)

|  |  |  |
| --- | --- | --- |
| Обучающийся 1 курса, направление подготовки «Программная инженерия» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Степанов Я.О. |
|  | подпись, дата | ФИО |
|  |  |  |
|  |  |  |
| Руководитель,  доцент кафедры  компьютерных технологий,  к. пед. н., доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, дата | Давыдова О.В. |
| уч. степень, уч. звание | ФИО |
|  |  |  |
|  |  |  |
| Заведующий кафедрой  компьютерных технологий,  д.пед. н., профессор | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись, дата | Лавина Т.А. |
| уч. степень, уч. звание | ФИО |
|  |  |  |

Чебоксары 2025

**Содержание**

[Введение 3](#_Toc471850995)

[1 Проектирование и разработка базы данных в СУБД MS Access……………………………4](#_Toc471850995)

[1.1 Постановка задачи 4](#_Toc471850995)

1.2. Разработка таблиц базы данных…………………………………………………...4

2 Разработка приложения Windows Form информационно-справочной системы 6

2.1 Подключение базы данных Access к приложению Visual Studio ……………….6

2.2 Разаботка интерфейса приложения………………………………………………...7

2.3. Работа с базой данных из DataGridView. Select, Insert, Update, Delete…………… 9

[Заключение 14](#_Toc471850999)

[Список использованных источников 15](#_Toc471850999)

[Приложение 16](#_Toc471851001)

**Введение**

Целями прохождения учебной практики являются:

* получение первичных профессиональных умений и навыков;
* закрепление, расширение и углубление теоретических и практических знаний умений и навыков, полученных обучающимися ранее при изучении дисциплин (модулей) учебного плана.

Задачи учебной практики:

* применение естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности классификация информационных технологий, в том числе отечественного производства, по назначению и характеру использования, по степени охвата задач управления;
* применение информационных технологий, в том числе отечественного производства, при решении теоретических и прикладных задач профессиональной деятельности;
* рациональный поиск информации в соответствии с потребностями в глобальных информационных сетях с использованием современных поисковых систем;
* применение информации, полученной из глобальных информационных сетей, с соблюдением законодательства в области информации, информационных технологий, защиты информации и авторского права;
* применение антивирусных программных средств и других методов защиты информации в профессиональной деятельности;
* формализация и разработка алгоритмов для поставленных задач;
* разработка программного кода с использованием языков программирования;
* оформление программного кода в соответствии с установленными требованиями.
* проверка и отладка программного кода;
* применение технологий программирования.

1. **Проектирование и разработка базы данных в СУБД MS Access**
   1. **Постановка задачи**

Разработать информационно справочную систему для работы с базой данных.

Входные данные:

1)База данных состоит из записей. (Код фильма, Название, Жанр, Длительность, Рейтинг, В отдельном файле - информация о сеансах)

Обеспечить:

- просмотр содержимого базы данных;

- редактирование базы данных:

а) удаление записи;

б) добавление записи;

в) редактирование записи:

г) сохранение данных в базу данных;

- поиск данных о сеансе по залу;

- поиск данных о фильме по жанру;

- сортировка данных;

- получение справочной информации

* 1. **Разработка таблиц базы данных**

Для создания базы данных использовалась СУБД MS Access. Была создана база данных с именем «FilmsBase1», где были созданы две таблицы: «Movie» (см.рисунок 1) и «Session» (см. рисунок 2) для хранения информации о фильмах и сеансах соответственно. Полями таблицы Movie являются: MovieID (формата «счётчик») для записи индивидуального номера фильма, Title (текстового формата) для записи названия фильма, Genre (текстового формата) для записи жанра фильма, Duration (числового формата) для записи продолжительности фильма в минутах и Rating (числового формата с точкой) для записи рейтинга. Полями таблицы Session являются: SessionID (формата «счётчик») для записи индивидуального номера сеанса, MovieID (формата «счетчик») для записи индивидуального номера фильма (связан с таблицей Movie), DateTime (формат «Дата:время») для даты и время сеанса , Hall(текстовый формат) для записи зала в котором проходит сеанс и Price(числовой формат с запятой) для записи цены за сеанс.

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рисунок 1 – Таблица «Movie» в СУБД MS Access

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

Рисунок 2 – Таблица «Session» в СУБД MS Access

Поле «MovieID» в таблице Movie связано с полем «MovieID» в таблице Session по принципу «один-ко-многим» (см. рисунок 4), что обеспечивает непрерывную связь между таблицами Movie и Session (см. рисунок 3). Это означает, что каждый сеанс из таблицы Session должен быть связан с определённым фильмом, информация о котором уже содержится в таблице Movie. Попытка нарушить этот принцип приведёт к возникновению ошибки. Кроме того, поля типа «счётчик» должны иметь уникальные значения, что исключает возможность наличия нескольких сеансов или фильмов с одинаковыми идентификаторами.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 3 – Связь между таблицами «Movie» и «Session» в СУБД MS Access

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 4 – Специфика связи таблиц «Movie» и «Session»

1. **Разработка приложения Windows Form ИС-системы**
   1. **Подключение БД Access к приложению Visual Studio**

Чтобы подключить БД Access к приложению Visual Studio был изначально выбран шаблон для работы с Windows Forms, использующими .NET Framework (см. рисунок 5).

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Рисунок 5 – Шаблон, на основе которого будет создано приложение

В качестве инструмента был использован встроенный функционал Visual Studio — интерфейс «Источники данных», в рамках которого был добавлен новый источник данных, представляющий собой заранее подготовленную базу данных MS Access

(см. рисунок 6).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 6 - интерфейс «Источники данных»

Путь к базе данных был задан относительный для облегчения работы с информационной системой с любого компьютера. Предполагается, что файл базы данных будет находиться в той же директории, что и исполняемый exe-файл (см. рисунок 7).

Изображение выглядит как текст, программное обеспечение, Шрифт, веб-страница

Автоматически созданное описание

Рисунок 7 – Расположение файла базы данных в одной директории с exe-файлом

* 1. **Разработка интерфейса приложения**

Было решено использовать интерфейс типа «Главное меню» (см. рисунок 8) для обеспечения удобства пользования приложением. В главном меню приложения были использованы 3 элемента button (кнопки «Сеансы», «Фильмы», «Выход»),1 label(надпись «Кинотеатр» ) и 1 pictureBox (изображение с котом).

Изображение выглядит как текст, кот, Мелкие и средние кошки, снимок экрана

Автоматически созданное описание

Рисунок 8 – интерфейс Главного меню

При нажатии на кнопку «Сеансы» открывается форма для просмотра и управления сеансами (см. рисунок 9). В ней есть DataGridView, Label, Buttons, TextBoxes и MenuStrip для поиска по залу. Можно добавлять, удалять, изменять сеансы, а также сортировать по цене (по возрастанию и убыванию). Кнопка «Назад» возвращает в главное меню.

При нажатии на кнопку «Фильмы» открывается форма для просмотра и управления фильмами (см. рисунок 10). В ней есть DataGridView, Label, Buttons, TextBoxes и MenuStrip для поиска по жанру. Можно добавлять, удалять, изменять фильмы, а также сортировать по рейтингу (по возрастанию и убыванию). Кнопка «Назад» возвращает в главное меню.

При нажатии на кнопку «Выход» приложение завершает работу и закрывается.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 9 – интерфейс «Сеансы»

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 10 – интерфейс «Фильмы»

* 1. **Работа с базой данных из DataGridView. Select, Insert, Update, Delete**

Метод INSERT INTO позволяет добавлять новые записи в таблицу. Важно отметить что таблицы связаны и не получиться добавить в таблицу «Session» код фильма, которого нет в таблице «Movie» Поэтому вначале стоит добавить новый фильм в таблицу «Movie»(см рисунок 11). Нужно нажать кнопку «Добавить фильм» ,заполнить все поля и нажать кнопку «Добавить» после этого добавиться новый фильм в таблицу «Movie». Затем у нас высветиться сообщение(см рисунок 12),свидетельствующее об успешном добавлении фильма. Также необходимо нажать кнопку «Обновить» для отображения актуальной информации о фильмах. Для обновления записей используем UPDATE

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 11 – интерфейс добавления нового фильма

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 12 – интерфейс сообщения успешного добавления фильма

Теперь добавим информацию о сеансе также при помощи метода INSERT INTO.Нужно нажать кнопку «Добавить сеанс» в форме «Сеансы» и заполнить поля новой формы (cм рисунок 13). Потом нажать кнопку «Добавить» и «Обновить» (Аналогично с добавлением фильмов)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 13 – интерфейс добавления нового сеанса

Для удаления записей используется sql-выражение DELETE. Удаление происходит по коду фильма(см рисунок 13). Также связные данные удаляются и из сеансов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 13 – Удаление данных о фильме

Для поиска записи в бд используем SELECT. В форме «Фильмы» нажимаем «Поиск» затем нажимаем «Фильмы по жанру» ,вводим интересующий наш жанр фильма и нажимаем поиск.

Аналогично можно сделать поиск в форме «Сеансы»(см. рисунок 15).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 14 – интерфейс Поиска по жанру фильма

Изображение выглядит как текст, снимок экрана, программное обеспечение, диаграмма

Автоматически созданное описание

Рисунок 15 – интерфейс поиска сеанса по залу

Для сортировки будем использовать кнопки «возраст» для сортировки по возрастанию цены , «убыв» по убыванию цены.

Также данная возможность реализована для фильмов по возрастанию и убыванию рейтинга.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 16 – сортировка сеансов по возрастанию цены

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 17 – сортировка фильмов по возрастанию рейтинга

**Заключение**

В ходе учебной практики:

1. Приобретены начальные профессиональные умения и навыки.

2. Углублены и расширены теоретические и практические знания, а также навыки,

полученные ранее в ходе изучения дисциплин.

3. Применены информационные технологии при расширении теоретических и

прикладных задач профессиональной деятельности.

4. Эффективно осуществлялся поиск информации в глобальных информационных

сетях с использованием современных поисковых систем, соответствуя потребностям.

5. Полученная из глобальных информационных систем информация использовалась

с соблюдением законодательства в области информации, информационных технологий,

защиты информации и авторского права.

6. Разработан программный код для решения задач с использованием языков

программирования С# и создана база данных в приложении Microsoft Access.

7. Программный код оформлен в соответствии с установленными требованиями.

8. Проверен и отлажен программный код.

**Список использованных источников**

1. Разработка Windows-приложений в среде программирования Visual Studio.Net [Электронный ресурс] : учебно-методическое пособие по дисциплине Информатика и программирование / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 20 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61536.html
2. Казанский, А. А. Программирование на visual c# 2013: учебное пособие для прикладного бакалавриата / А. А. Казанский. — М. : Издательство Юрайт, 2018. — 191 с. [Электронный ресурс]. URL: https:// https://urait.ru/book/programmirovanie-na-visual-c-2013-414194
3. Гаспариан М.С. Информационные системы и технологии [Электронный ресурс] : учебное пособие / М.С. Гаспариан, Г.Н. Лихачева. — Электрон. текстовые данные. — М. : Евразийский открытый институт, 2011. — 370 c. — 978-5-374-00192-1. — Режим доступа: <http://www.iprbookshop.ru/10680.html>
4. Курипта О.В. Основы программирования и алгоритмизации [Электронный ресурс] : практикум / О.В. Курипта, О.В. Минакова, Д.К. Проскурин. – Электрон. текстовые данные. – Воронеж: Воронежский государственный архитектурно-строительный университет, ЭБС АСВ, 2015. – 133 c. – 978-5-89040-575-3. – Режим доступа: http://www.iprbookshop.ru/59123.html
5. ГОСТ 7.32-2017. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления [Электронный ресурс] . URL: <http://protect.gost.ru/document.aspx?control=7&id=218998>
6. Общероссийский классификатор стандартов. Программное обеспечение [Электронный ресурс]. URL: <http://standartgost.ru/0/757-programmnoe_obespechenie>.

**ПРИЛОЖЕНИЕ**

**Main(Главное меню)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Pract

{

public partial class Main : Form

{

public Main()

{

InitializeComponent();

}

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

{

Form1 form1 = new Form1();

form1.Show();

this.Hide();

}

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

{

Form2 form2 = new Form2();

form2.Show();

this.Hide();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

Close();

}

private void Main\_Load(object sender, EventArgs e)

{

}

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

{

}

}

}

**Form1(Сеансы)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Pract

{

public partial class Main : Form

{

public Main()

{

InitializeComponent();

}

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

{

Form1 form1 = new Form1();

form1.Show();

this.Hide();

}

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

{

Form2 form2 = new Form2();

form2.Show();

this.Hide();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

Close();

}

private void Main\_Load(object sender, EventArgs e)

{

}

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

{

}

}

}

**Search(Поиск по залу)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Pract

{

public partial class Search : Form

{

public static string connectString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=FilmsBase1.mdb";

private OleDbConnection myConn;

public Search()

{

InitializeComponent();

}

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

{

}

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

{

myConn = new OleDbConnection(connectString);

myConn.Open();

string num=textBox1.Text;

string query = "SELECT [SessionID],[MovieID],[DateTime],[Hall],[Price] FROM [Session] WHERE [Hall] LIKE '%" + num + "%'";

OleDbDataAdapter comm = new OleDbDataAdapter(query,myConn);

DataTable dt = new DataTable();

comm.Fill(dt);

dataGridView1.DataSource = dt;

myConn.Close();

}

private void Search\_Load(object sender, EventArgs e)

{

}

}

}

**SessionAdd(добавление сеанса)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Pract

{

public partial class SessionAdd : Form

{

public static string connectString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=FilmsBase1.mdb";

private OleDbConnection myConn;

public SessionAdd()

{

InitializeComponent();

myConn = new OleDbConnection(connectString);

myConn.Open();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

int sesid = Convert.ToInt32(textBox4.Text);

int movid = Convert.ToInt32(textBox1.Text);

DateTime data = dateTimePicker1.Value;

string hall = textBox2.Text;

decimal pric = Convert.ToDecimal(textBox3.Text);

string query = "INSERT INTO [Session] ([SessionID],[MovieID],[DateTime],[Hall],[Price]) VALUES (" + sesid + ", " + movid + ", '" + data.ToString("yyyy-MM-dd HH:mm:ss") + "', '" + hall + "', " + pric.ToString(CultureInfo.InvariantCulture) + ")";

OleDbCommand command = new OleDbCommand(query,myConn);

command.ExecuteNonQuery();

MessageBox.Show("Данные о сеансе добавлены");

}

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

{

this.Close();

}

private void SessionAdd\_Load(object sender, EventArgs e)

{

}

}

}

**SessionUpdate(изменение сеансов)**

using System;

using System.Data.OleDb;

using System.Windows.Forms;

namespace Pract

{

public partial class SessionUpdatecs : Form

{

public Form1 MainForm { get; set; }

public int SessionId { get; set; } // Добавляем свойство для хранения ID сеанса

private static string connect = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=FilmsBase1.mdb";

private OleDbConnection dbConnection;

public SessionUpdatecs(int sessionId) // Добавляем параметр в конструктор

{

InitializeComponent();

SessionId = sessionId;

dbConnection = new OleDbConnection(connect);

dbConnection.Open();

LoadSessionData(); // Загружаем данные сеанса при открытии формы

}

private void LoadSessionData()

{

string query = "SELECT \* FROM [Session] WHERE [SessionID] = ?";

OleDbCommand command = new OleDbCommand(query, dbConnection);

command.Parameters.AddWithValue("?", SessionId);

using (OleDbDataReader reader = command.ExecuteReader())

{

if (reader.Read())

{

textBox1.Text = reader["MovieID"].ToString();

dateTimePicker1.Value = Convert.ToDateTime(reader["DateTime"]);

textBox2.Text = reader["Hall"].ToString();

textBox3.Text = reader["Price"].ToString();

}

}

}

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

{

try

{

int movid = Convert.ToInt32(textBox1.Text);

DateTime data = dateTimePicker1.Value;

string hall = textBox2.Text;

decimal price = Convert.ToDecimal(textBox3.Text); // Используем decimal для денег

string query = @"UPDATE [Session] SET

[MovieID] = ?,

[DateTime] = ?,

[Hall] = ?,

[Price] = ?

WHERE [SessionID] = ?";

OleDbCommand command = new OleDbCommand(query, dbConnection);

command.Parameters.AddWithValue("?", movid);

command.Parameters.AddWithValue("?", data);

command.Parameters.AddWithValue("?", hall);

command.Parameters.AddWithValue("?", price);

command.Parameters.AddWithValue("?", SessionId);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Сеанс успешно обновлен!");

MainForm.RefreshData(); // Вызываем метод обновления данных в главной форме

this.Close();

}

else

{

MessageBox.Show("Сеанс не найден!");

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

private void exit\_button\_Click(object sender, EventArgs e)

{

this.Close();

}

private void SessionUpdatecs\_Load(object sender, EventArgs e)

{

// Код для загрузки формы (если нужен)

}

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

{

// Код для обработки нажатия кнопки 2

// Например, закрытие формы:

this.Close();

}

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

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

}

}

}

**Form2(фильмы)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Pract

{

public partial class Form2 : Form

{

public static string connectString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=FilmsBase1.mdb";

private OleDbConnection myConn;

private object sessionTableAdapter;

public Form2()

{

InitializeComponent();

myConn = new OleDbConnection(connectString);

myConn.Open();

}

public void RefreshData()

{

this.movieTableAdapter.Fill(this.filmsBase1DataSet.Movie);

}

private void Form2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "filmsBase1DataSet.Movie". При необходимости она может быть перемещена или удалена.

this.movieTableAdapter.Fill(this.filmsBase1DataSet.Movie);

}

private void Form2\_FormClosing\_1(object sender, FormClosingEventArgs e)

{

myConn.Close();

Main mainForm = Application.OpenForms.OfType<Main>().FirstOrDefault();

if (mainForm != null)

{

mainForm.Show(); // Показываем Main, если он был скрыт

}

else

{

// Если Main не был открыт, создаём новый экземпляр

Main newMain = new Main();

newMain.Show();

}

}

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

{

try

{

int code = Convert.ToInt32(textBox2.Text);

string query = "DELETE FROM [Movie] WHERE [MovieID] = ?";

OleDbCommand command = new OleDbCommand(query, myConn);

command.Parameters.AddWithValue("?", code);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Данные о фильме удалены");

RefreshData();

}

else

{

MessageBox.Show("Сеанс с указанным ID не найден");

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

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

{

if (string.IsNullOrWhiteSpace(textBox3.Text))

{

MessageBox.Show("Введите ID сеанса для редактирования!");

textBox3.Focus();

return;

}

// Проверяем, что введено число

if (!int.TryParse(textBox3.Text, out int movieId))

{

MessageBox.Show("ID фильмв должен быть целым числом!");

textBox3.SelectAll();

textBox3.Focus();

return;

}

// Проверяем существование сеанса в базе данных

try

{

string checkQuery = "SELECT COUNT(\*) FROM [Movie] WHERE [MovieID] = ?";

using (OleDbCommand checkCmd = new OleDbCommand(checkQuery, myConn))

{

checkCmd.Parameters.AddWithValue("?", movieId);

int exists = (int)checkCmd.ExecuteScalar();

if (exists == 0)

{

MessageBox.Show($"Фильм с ID {movieId} не найден!");

textBox3.SelectAll();

textBox3.Focus();

return;

}

}

// Если сеанс существует, открываем форму редактирования

FilmUpdate editForm = new FilmUpdate(movieId);

editForm.MainForm = this;

editForm.ShowDialog();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при проверке фильмы: {ex.Message}");

}

}

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

{

FilmAdd ad = new FilmAdd();

ad.Owner = this;

ad.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

this.movieTableAdapter.Fill(this.filmsBase1DataSet.Movie);

}

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

{

}

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

{

SearchFilm sea = new SearchFilm();

sea.Owner = this;

sea.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

myConn.Close();

Main mainForm = Application.OpenForms.OfType<Main>().FirstOrDefault();

if (mainForm != null)

{

mainForm.Show();

this.Close();// Показываем Main, если он был скрыт

}

else

{

// Если Main не был открыт, создаём новый экземпляр

Main newMain = new Main();

newMain.Show();

this.Close();

}

}

private void LoadData(string sortOrder = "")

{

using (OleDbConnection connection = new OleDbConnection(connectString))

{

string query = "SELECT [MovieID], [Title],[Genre],[Duration],[Rating] FROM [Movie]";

if (!string.IsNullOrEmpty(sortOrder))

{

query += $" ORDER BY [Rating] {sortOrder}";

}

OleDbDataAdapter adapter = new OleDbDataAdapter(query, connection);

DataTable table = new DataTable();

adapter.Fill(table);

dataGridView1.DataSource = table;

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button6\_Click(object sender, EventArgs e)

{

LoadData("ASC");

}

private void button7\_Click(object sender, EventArgs e)

{

LoadData("DESC");

}

}

}

**SearchFilm(поиск по жанру фильма)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Pract

{

public partial class SearchFilm : Form

{

public static string connectString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=FilmsBase1.mdb";

private OleDbConnection myConn;

public SearchFilm()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

myConn = new OleDbConnection(connectString);

myConn.Open();

string genre = textBox1.Text;

string query = "SELECT [MovieID],[Title],[Genre],[Duration],[Rating] FROM [Movie] WHERE [Genre] LIKE '%" + genre + "%'";

OleDbDataAdapter comm = new OleDbDataAdapter(query, myConn);

DataTable dt = new DataTable();

comm.Fill(dt);

dataGridView1.DataSource = dt;

myConn.Close();

}

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

{

}

private void SearchFilm\_Load(object sender, EventArgs e)

{

}

}

}

**FilmAdd(добавление фильма)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Pract

{

public partial class FilmAdd : Form

{

public static string connectString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=FilmsBase1.mdb";

private OleDbConnection myConn;

public FilmAdd()

{

InitializeComponent();

myConn = new OleDbConnection(connectString);

myConn.Open();

}

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

{

int movid = Convert.ToInt32(textBox1.Text);

string title = textBox3.Text;

string genre = textBox2.Text;

string prod = textBox4.Text;

string rating = textBox5.Text;

string query = "INSERT INTO [Movie] ([MovieID], [Title], [Genre], [Duration], [Rating]) " +

"VALUES (@MovieID, @Title, @Genre, @Duration, @Rating)";

// Создаем команду с параметрами

using (OleDbCommand command = new OleDbCommand(query, myConn))

{

// Добавляем параметры с соответствующими значениями

command.Parameters.AddWithValue("@MovieID", movid);

command.Parameters.AddWithValue("@Title", title);

command.Parameters.AddWithValue("@Genre", genre);

command.Parameters.AddWithValue("@Duration", prod);

command.Parameters.AddWithValue("@Rating", rating);

// Выполняем запрос

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Данные о фильме успешно добавлены!");

}

else

{

MessageBox.Show("Не удалось добавить данные о фильме");

}

}

}

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

{

this.Close();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void FilmAdd\_Load(object sender, EventArgs e)

{

}

}

}

**FilmUpdate(изменение фильмов в таблице)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Pract

{

public partial class FilmUpdate : Form

{

public Form2 MainForm { get; set; }

public int MovieId { get; set; } // Добавляем свойство для хранения ID сеанса

private static string connect = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=FilmsBase1.mdb";

private OleDbConnection dbConnection;

public FilmUpdate(int movieId)

{

InitializeComponent();

MovieId = movieId;

dbConnection = new OleDbConnection(connect);

dbConnection.Open();

LoadFilmData();

}

private void LoadFilmData()

{

string query = "SELECT \* FROM [Movie] WHERE [MovieID] = ?";

OleDbCommand command = new OleDbCommand(query, dbConnection);

command.Parameters.AddWithValue("?", MovieId);

using (OleDbDataReader reader = command.ExecuteReader())

{

if (reader.Read())

{

textBox4.Text = reader["Title"].ToString();

textBox2.Text = reader["Genre"].ToString();

textBox3.Text = reader["Duration"].ToString();

textBox5.Text = reader["Rating"].ToString();

}

}

}

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

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

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

{

this.Close();

}

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

{

try

{

string title = textBox4.Text;

string genre = textBox2.Text;

string duration = textBox3.Text;

string rating = textBox5.Text;

string query = @"UPDATE [Movie] SET

[Title] = ?,

[Genre] = ?,

[Duration] = ?,

[Rating] = ?

WHERE [MovieID] = ?";

OleDbCommand command = new OleDbCommand(query, dbConnection);

command.Parameters.AddWithValue("?", title);

command.Parameters.AddWithValue("?", genre);

command.Parameters.AddWithValue("?", duration);

command.Parameters.AddWithValue("?", rating);

command.Parameters.AddWithValue("?", MovieId);

int rowsAffected = command.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Данные фильма успешно обновлены!");

MainForm.RefreshData(); // Вызываем метод обновления данных в главной форме

this.Close();

}

else

{

MessageBox.Show("Фильм не найден!");

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

private void FilmUpdate\_Load(object sender, EventArgs e)

{

}

}

}