|  |  |  |  |
| --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН  СЕВЕРО-КАЗАХСТАНСКИЙ УНИВЕРСИТЕТ ИМ. М. КОЗЫБАЕВА  ФАКУЛЬТЕТ ИНЖЕНЕРИИ И ЦИФРОВЫХ ТЕХНОЛОГИЙ  КАФЕДРА «ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ» | | | |
|  | | | |
|  | | |  |
| Объектно-ориентированное программирование  МОДУЛЬ 4  Вариант 2 | | | |
| Выполнил студент  группы ВТиПО-22 |  | Белокопытов Я.Ю. | |
| старший преподаватель, магистр информационных систем |  | Долматова Л.В. | |
|  |  |  | |

Петропавловск, 2024

ЛАБОРАТОРНАЯ РАБОТА №6 ( Упр1,2,3 )

Листинг кода:

using System;

using System.Drawing;

using System.Drawing.Printing;

using System.Windows.Forms;

namespace WindowsFormsApp4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string s = ""; // Строка для хранения всего текста из файла

string[] strings = new string[0]; // Инициализация пустого массива строк

int ArrayCounter = 0; // Счётчик для отслеживания текущей строки в массиве

// Метод для обработки события печати страницы

private void printDocument1\_PrintPage(object sender, PrintPageEventArgs e)

{

if (strings == null || strings.Length == 0)

{

MessageBox.Show("Файл не загружен или пуст.");

e.Cancel = true; // Отменяем печать, если массив строк пуст

return;

}

float LeftMargin = e.MarginBounds.Left;

float TopMargin = e.MarginBounds.Top;

float MyLines = e.MarginBounds.Height / this.Font.GetHeight(e.Graphics);

float YPosition;

int Counter = 0;

string CurrentLine;

while (Counter < MyLines && ArrayCounter < strings.Length)

{

CurrentLine = strings[ArrayCounter];

YPosition = TopMargin + Counter \* this.Font.GetHeight(e.Graphics);

e.Graphics.DrawString(CurrentLine, this.Font, Brushes.Black, LeftMargin, YPosition, new StringFormat());

Counter++;

ArrayCounter++;

}

e.HasMorePages = ArrayCounter < strings.Length;

}

// Кнопка настройки страницы

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

{

pageSetupDialog1.ShowDialog();

}

// Кнопка печати с диалогом печати

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

{

if (printDialog1.ShowDialog() == DialogResult.OK)

printDocument1.Print();

}

// Кнопка предварительного просмотра

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

{

Form2 aForm = new Form2();

// Связываем printDocument1 с PrintPreviewControl на Form2

aForm.printPreviewControl1.Document = printDocument1;

DialogResult aResult = aForm.ShowDialog();

if (aResult == DialogResult.OK)

{

printDocument1.Print();

}

}

// Кнопка для загрузки файла

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

{

var aResult = openFileDialog1.ShowDialog();

if (aResult == DialogResult.OK)

{

System.IO.StreamReader aReader = new System.IO.StreamReader(openFileDialog1.FileName);

s = aReader.ReadToEnd();

aReader.Close();

strings = s.Split(new[] { '\n' }, StringSplitOptions.RemoveEmptyEntries);

ArrayCounter = 0; // Сбрасываем счётчик строк

}

}

private void printPreviewDialog1\_Load(object sender, EventArgs e)

{

printPreviewDialog1.Document = printDocument1;

}

}

}

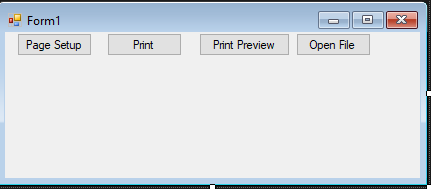


Рисунок 1 Результаты

Лабораторная работа 7 ( Упр1,2,3 )

Листинг кода:

using System;

using System.ComponentModel;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Laba7\_Module4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

backgroundWorker1.WorkerReportsProgress = true;

backgroundWorker1.WorkerSupportsCancellation = true;

backgroundWorker1.DoWork += backgroundWorker1\_DoWork;

backgroundWorker1.ProgressChanged += backgroundWorker1\_ProgressChanged;

backgroundWorker1.RunWorkerCompleted += backgroundWorker1\_RunWorkerCompleted;

}

// Упражнение 1: BackgroundWorker

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

{

if (int.TryParse(textBoxSeconds.Text, out int seconds))

{

progressBar1.Value = 0;

backgroundWorker1.RunWorkerAsync(seconds);

}

else

{

MessageBox.Show("Введите корректное число секунд.");

}

}

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

{

if (backgroundWorker1.IsBusy)

backgroundWorker1.CancelAsync();

}

private void backgroundWorker1\_DoWork(object sender, DoWorkEventArgs e)

{

int seconds = (int)e.Argument;

for (int i = 1; i <= seconds; i++)

{

if (backgroundWorker1.CancellationPending)

{

e.Cancel = true;

return;

}

Thread.Sleep(1000); // Имитация работы

backgroundWorker1.ReportProgress(i \* 100 / seconds);

}

}

private void backgroundWorker1\_ProgressChanged(object sender, ProgressChangedEventArgs e)

{

progressBar1.Value = e.ProgressPercentage;

}

private void backgroundWorker1\_RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)

{

if (e.Cancelled)

{

MessageBox.Show("Процесс отменен.");

}

else

{

MessageBox.Show("Процесс завершен.");

}

}

// Упражнение 2: Async Delegate

private delegate int CalculateDelegate(int a, int b);

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

{

if (int.TryParse(textBoxA.Text, out int a) && int.TryParse(textBoxB.Text, out int b))

{

CalculateDelegate calculate = (x, y) => x + y;

calculate.BeginInvoke(a, b, AsyncDelegateCallback, calculate);

}

else

{

MessageBox.Show("Введите корректные значения для A и B.");

}

}

private void AsyncDelegateCallback(IAsyncResult ar)

{

var calculate = (CalculateDelegate)ar.AsyncState;

int result = calculate.EndInvoke(ar);

Invoke(new Action(() => MessageBox.Show("Результат: " + result)));

}

// Упражнение 3: Task

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

{

RunTask();

}

private async void RunTask()

{

int result = await Task.Run(() => LongRunningTask());

labelTaskResult.Text = "Результат: " + result;

}

private int LongRunningTask()

{

Thread.Sleep(2000); // Имитация долгой операции

return new Random().Next(100); // Рандомный результат

}

}

}

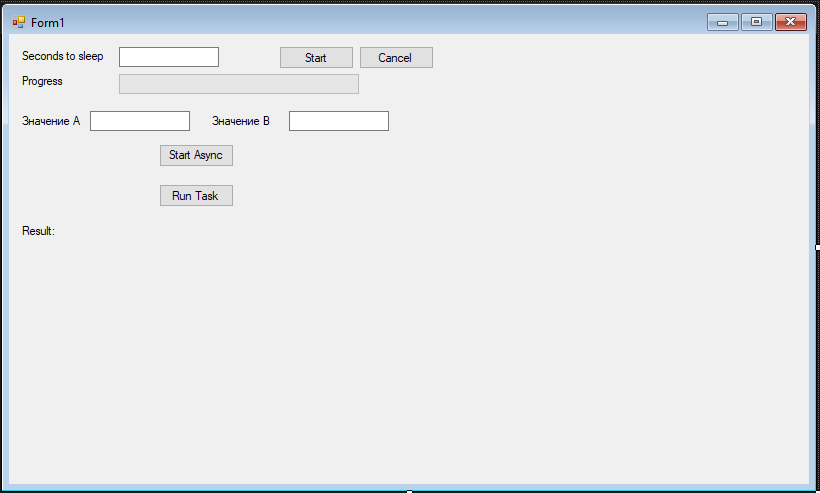


Рисунок 2 Результат

# Лабораторная работа доп

Листинг кода:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace LabaLast\_M4

{

public partial class Form1 : Form

{

private List<Film> films;

private List<Session> sessions;

public Form1()

{

InitializeComponent();

InitializeData();

}

private void InitializeData()

{

films = new List<Film>

{

new Film { Title = "Интерстеллар", Genre = "Фантастика, Драма", Duration = 169, Price = 18.50m },

new Film { Title = "Титаник", Genre = "Драма, Мелодрама", Duration = 195, Price = 20.00m },

new Film { Title = "Начало", Genre = "Боевик, Приключения, Фантастика", Duration = 148, Price = 19.00m },

new Film { Title = "Матрица", Genre = "Боевик, Фантастика", Duration = 136, Price = 18.50m },

new Film { Title = "Побег из Шоушенка", Genre = "Драма", Duration = 142, Price = 15.50m },

new Film { Title = "Темный рыцарь", Genre = "Боевик, Криминал, Драма", Duration = 152, Price = 20.00m },

new Film { Title = "Гладиатор", Genre = "Боевик, Драма, История", Duration = 155, Price = 17.50m },

new Film { Title = "Форрест Гамп", Genre = "Драма, Мелодрама", Duration = 142, Price = 17.00m },

new Film { Title = "Властелин колец: Возвращение короля", Genre = "Боевик, Фэнтези, Драма", Duration = 201, Price = 22.00m },

new Film { Title = "Крёстный отец", Genre = "Криминал, Драма", Duration = 175, Price = 18.00m }

};

sessions = new List<Session>

{

new Session { Film = films[0], StartTime = DateTime.Now.AddHours(1), AvailableSeats = 30 },

new Session { Film = films[1], StartTime = DateTime.Now.AddHours(2), AvailableSeats = 25 },

new Session { Film = films[2], StartTime = DateTime.Now.AddHours(3), AvailableSeats = 20 },

new Session { Film = films[3], StartTime = DateTime.Now.AddHours(4), AvailableSeats = 15 }

};

cmbFilms.DataSource = films;

cmbFilms.DisplayMember = "Title";

dgvFilms.DataSource = films;

}

private void FilterFilmsByGenre()

{

var filteredFilms = films.Where(f => f.Genre.IndexOf(txtFilterGenre.Text, StringComparison.OrdinalIgnoreCase) >= 0).ToList();

dgvFilms.DataSource = filteredFilms;

}

private void ShowSessions()

{

var selectedFilm = cmbFilms.SelectedItem as Film;

if (selectedFilm != null)

{

var filmSessions = sessions.Where(s => s.Film == selectedFilm).ToList();

lstSessions.Items.Clear();

foreach (var session in filmSessions)

{

lstSessions.Items.Add($"{session.StartTime}: {session.AvailableSeats} seats available");

}

}

}

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

{

FilterFilmsByGenre();

}

private void cmbFilms\_SelectedIndexChanged(object sender, EventArgs e)

{

ShowSessions();

}

}

}

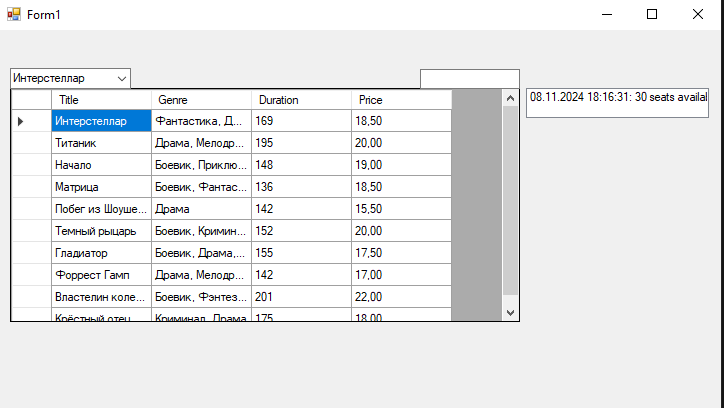


Рисунок 3 Результат