Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 «Программирование в компьютерных системах»

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

*(наименование профессионального модуля)*

Выполнил:

обучающийся учебной группы № 420

Бек-Назарова Анастасия Николаевна

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

И.В. Сибирев

*(И.О. Фамилия)*

Е.Л. Альшакова

*(И.О. Фамилия)*

**Москва**

**20­­­­22**

**Перечень заданий/работ, выполненных в ходе учебной практики**

Специальность 09.02.03 «Программирование в компьютерных системах»

*(наименование специальности)*

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

*(наименование профессионального модуля)*

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Темы учебной практики** | **Выполненные задания/работы** |
| 1 | Тема 1. Программирование разветвляющегося процесса | Разработать программу сравнения двух чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 2 | Тема 2. Программирование разветвляющихся алгоритмов с помощью команд передачи управления языка Assembler | Реализовать алгоритм ветвления на языке Assembler соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 3 | Тема 3. Работа с флагами регистра флагов с использованием команд языка Assembler | Разработать программу с использованием регистра FLAGS соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 4 | Тема 4. Программирование алгоритмов с использованием арифметических команд и флага переноса языка Assembler | Разработать программу решения квадратного уравнения соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 5 | Тема 5. Выполнение операций сдвига и логических операций над числами с помощью команд языка Assembler | Разработать программу с логических преобразований чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 6 | Тема 6. Умножение знаковых чисел | Разработать программу вычисления произведений произвольных чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 7 | Тема 7. Работа с флагами CF, OF. Деление знаковых чисел | Разработать программу вычисления деления произвольных чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 8 | Тема 8. Циклы | Разработать программу вычисления значений функций разложением в ряд Тейлора соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 9 | Тема 9. Работа с сегментами. Подпрограммы | Разработать программу с использованием подпрограмм соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 10 | Тема 10. Обработка одномерных массивов с использованием команд языка Assembler | Разработать программу обработки одномерного массива соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 11 | Тема 11. Работа со стеком | Разработать программу с использованием стека соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 12 | Тема 12. Работа с файлами | Разработать программу работы с файлом соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 13 | Тема 13. Ассемблерная вставка в С++ | Разработать программу на языке соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 14 | Тема 14. Разработка программы с использованием Windows Forms | Разработать программу с использованием Windows Forms соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 15 | Тема 15. Написание программы для произведения стереометрических расчетов | Разработать программу для произведения стереометрических расчетов соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 16 | Тема 16. Написание консольного приложения для вывода числового массива | Разработать консольное приложение для вывода числового массива соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 17 | Тема 17. Написание приложения для работы с символьным массивом | Разработать приложение для работы с символьным массивом соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 18 | Тема 18. Создание приложения с использованием компонента ListBox | Разработать приложение с использованием компонента ListBox соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 19 | Тема 19. Создание приложения с использованием компонента DataGridView и многострочным textBox | Разработать приложение с использованием компонента DataGridView и многострочным textBox соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 20 | Тема 20. Создание приложения с для произведения планиметрических расчетов с использованием классов | Разработать приложение с для произведения планиметрических расчетов с использованием классов соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 21 | Тема 21. Рефакторинг и оптимизация ранее разработанного приложения для обработки числового массива с использованием классов | Произвести разработку приложения для обработки числового массива с использованием классов соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 22 | Тема 22. Создание приложения с использованием свойства и методов класса String | Разработать приложение с использованием свойств и методов класса String соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 23 | Тема 23. Создание приложения с использованием делегатов (часть 1) | Разработать приложение с использованием делегатов соблюдая следующие этапы: - определить спецификации;  - создать алгоритм. |
| 24 | Тема 24. Создание приложения с использованием делегатов (часть 2) | Разработать приложение с использованием делегатов соблюдая следующие этапы:  - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 25 | Тема 25. Создание приложения с использованием событий | Разработать приложение с использованием событий соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 26 | Тема 26. Создание приложения «Калькулятор» с использованием классов | Разработать приложение «Калькулятор» с использованием событий соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 27 | Тема 27. Создание приложения для проверки орфографии | Разработать приложение для проверки орфографии соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 28 | Тема 28. Создание приложения для организации взаимодействия с MS Word | Разработать приложение для организации взаимодействия с MS Word соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 29 | Тема 29. Создание приложения с использованием функций MS Excel | Разработать приложение с использованием функций MS Excel соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 30 | Тема 30. Создание поточного приложения | Разработать поточное приложение соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 31 | Тема 31. Защита отчета по практике | Защита сформированного отчета по практике. |

**СОДЕРЖАНИЕ**

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1**

**Тема: «События C#»**

Студент: Бек-Назарова А. Н.

Группа: 3ПКС-420

Преподаватель: Сибирев И. В.

Преподаватель: Альшакова Е. Л.

Дата: 24.11.2022

**Цель работы**: разработать калькулятор на языке С#.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, монитор, снимок экрана, серебряный

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

**Вывод**: разработан калькулятор на языке С#.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 2**

**Тема: «Проверка орфографии»**

Студент: Бек-Назарова А. Н.

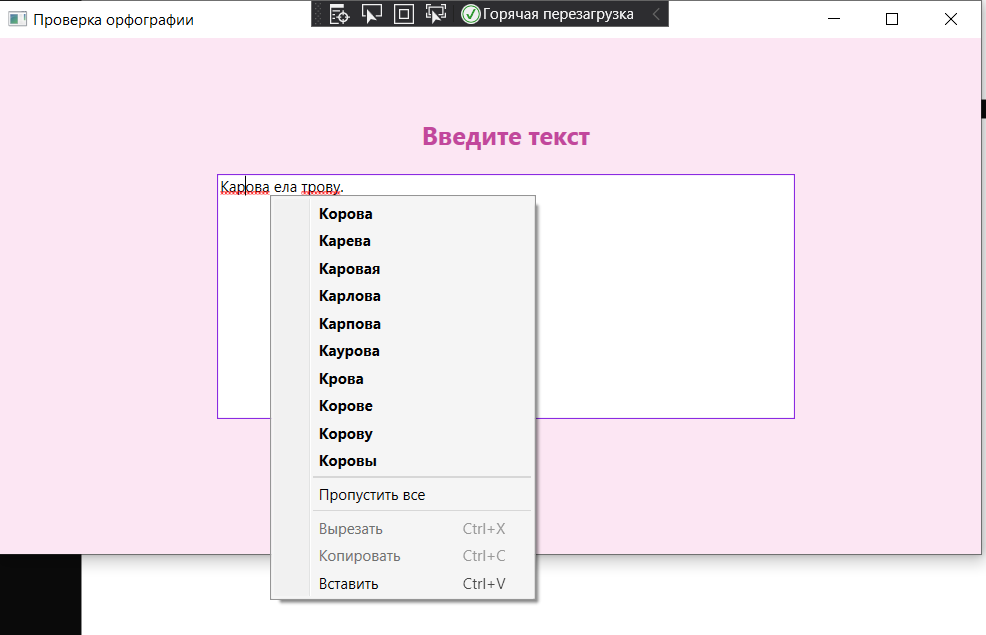
Группа: 3ПКС-420

Преподаватель: Сибирев И. В.

Преподаватель: Альшакова Е. Л.

Дата: 24.11.2022

**Цель работы**: разработать программу на языке С#, которая предлагает пользователю ввести какие-либо слова, предложения в текстовое поле и проверяет их на орфографию.



<Window x:Class="FormatText.MainWindow"

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-compatibility/2006"

xmlns:local="clr-namespace:FormatText"

mc:Ignorable="d"

Title="Проверка орфографии" Height="450" Width="800" Background="#fce6f3">

<Grid>

<TextBox SpellCheck.IsEnabled="True" HorizontalAlignment="Left" Height="196" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="463" Margin="174,109,0,0" BorderBrush="BlueViolet"/>

<TextBlock HorizontalAlignment="Left" TextWrapping="Wrap" Text="Введите текст" VerticalAlignment="Top" Margin="174,64,0,0" Width="463" FontSize="20" FontWeight="Bold" TextAlignment="Center" Height="27" Foreground="#c2479c"/>

</Grid>

</Window>

**Вывод**: разработана программа на языке С#, которая предлагает пользователю ввести какие-либо слова, предложения в текстовое поле и проверяет их на орфографию.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 3**

**Тема: «Создание таблицы в MS Word»**

Студент: Бек-Назарова А. Н.

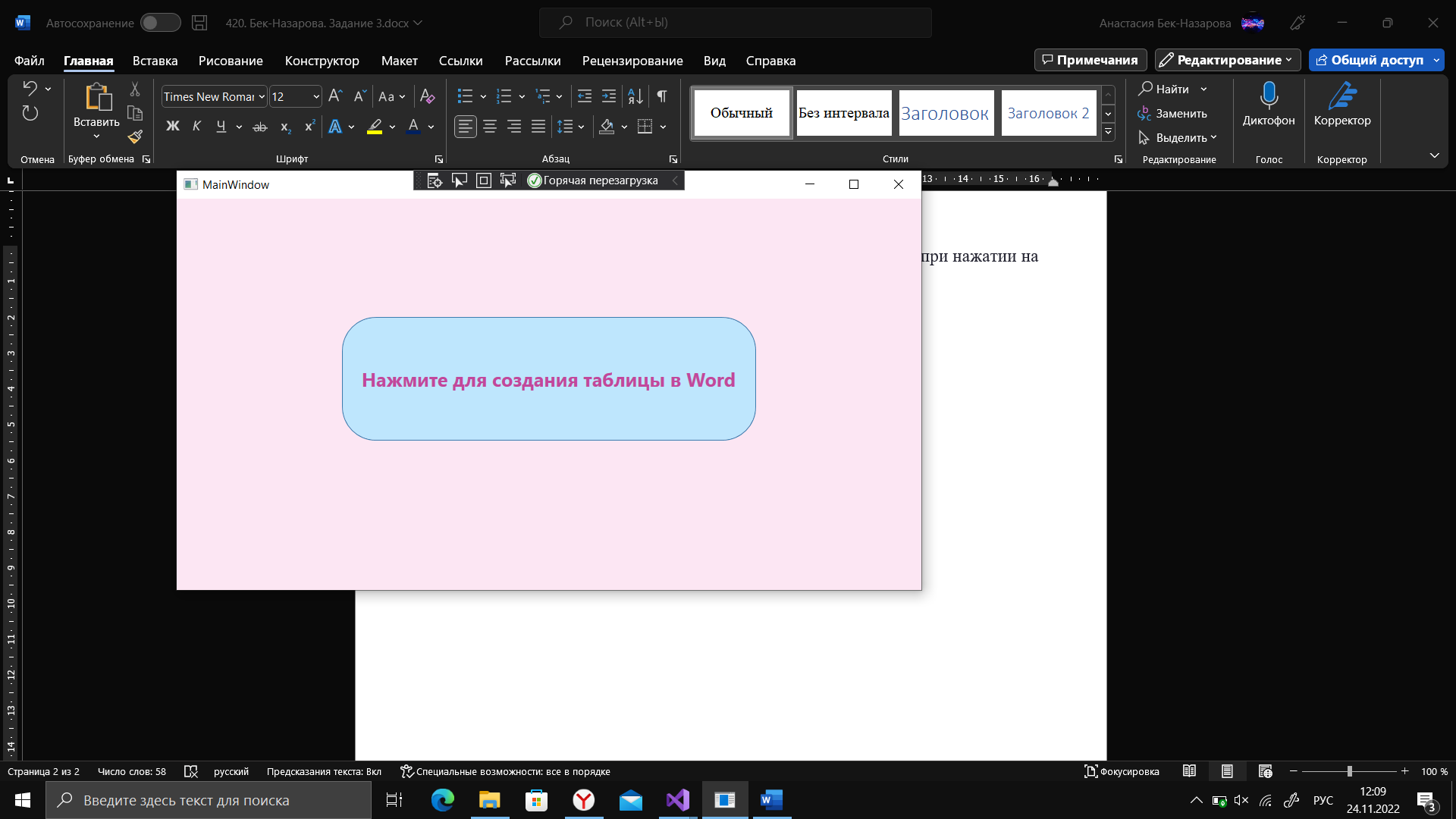
Группа: 3ПКС-420

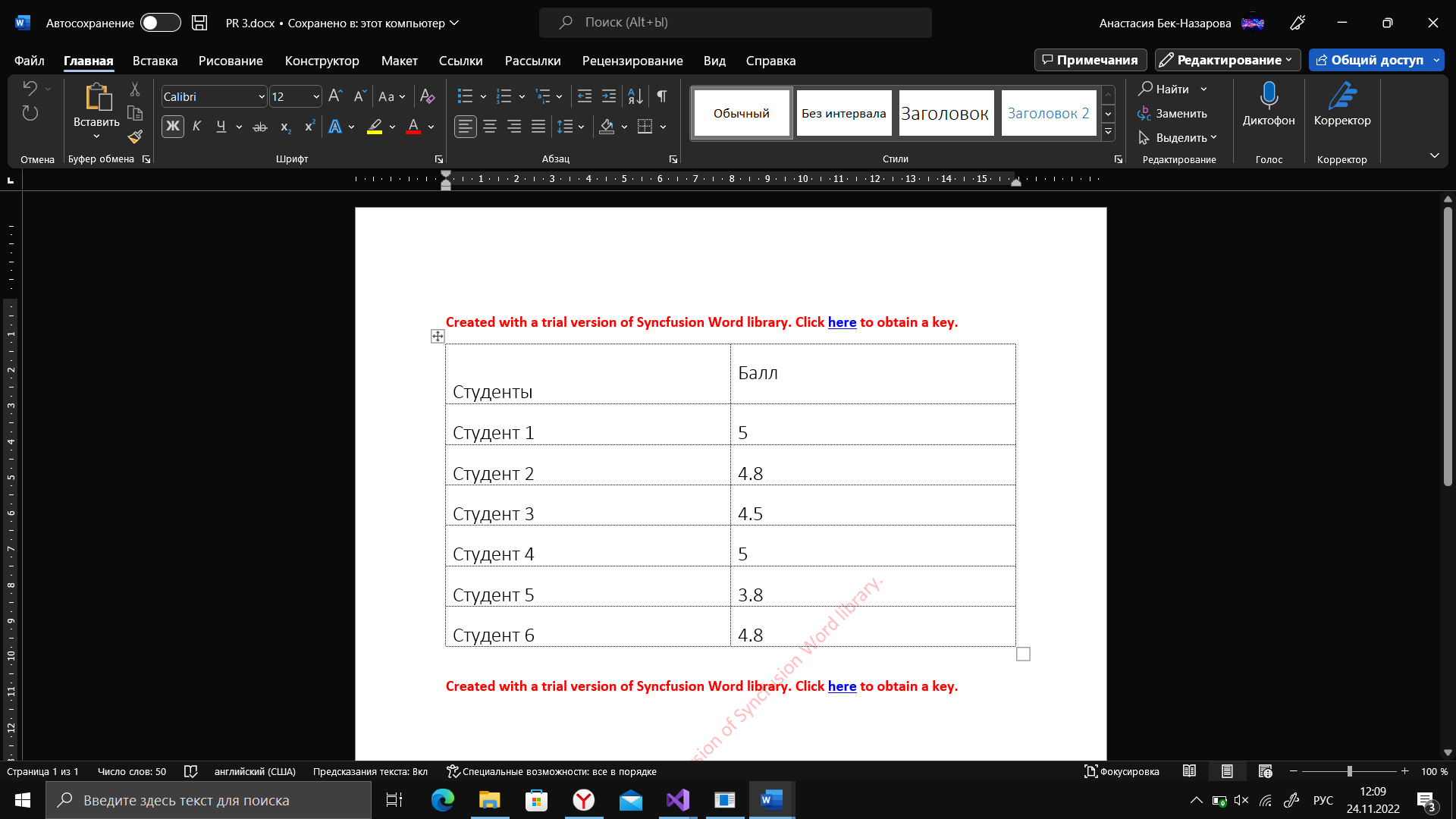
Преподаватель: Сибирев И. В.

Преподаватель: Альшакова Е. Л.

Дата: 24.11.2022

**Цель работы**: необходимо разработать программу, которая при нажатии на кнопку создает таблицу в MS Word.





using Syncfusion.DocIO;

using Syncfusion.DocIO.DLS;

using System;

using System.ComponentModel;

using System.Diagnostics;

using System.Drawing;

using System.Windows;

namespace zadanie3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

}

private void Button\_Click(object sender, RoutedEventArgs e)

{// Creating a new document.

WordDocument document = new WordDocument();

//Adding a new section to the document.

WSection section = document.AddSection() as WSection;

//Set Margin of the section

section.PageSetup.Margins.All = 72;

//Set page size of the section

//Create Paragraph styles

WParagraphStyle style = document.AddParagraphStyle("Normal") as WParagraphStyle;

style.CharacterFormat.FontName = "Calibri";

style.CharacterFormat.FontSize = 11f;

style.ParagraphFormat.BeforeSpacing = 0;

style.ParagraphFormat.AfterSpacing = 8;

style.ParagraphFormat.LineSpacing = 13.8f;

style = document.AddParagraphStyle("Heading 1") as WParagraphStyle;

style.ApplyBaseStyle("Normal");

style.CharacterFormat.FontName = "Calibri Light";

style.CharacterFormat.FontSize = 16f;

style.ParagraphFormat.BeforeSpacing = 12;

style.ParagraphFormat.AfterSpacing = 0;

style.ParagraphFormat.Keep = true;

style.ParagraphFormat.KeepFollow = true;

style.ParagraphFormat.OutlineLevel = OutlineLevel.Level1;

IWParagraph paragraph = section.HeadersFooters.Header.AddParagraph();

// Gets the image stream.

//Appends table.

IWTable table = section.AddTable();

table.ResetCells(7, 2);

table.TableFormat.Borders.BorderType = BorderStyle.Dot;

table.TableFormat.IsAutoResized = true;

//Appends paragraph.

paragraph = table[0, 0].AddParagraph();

paragraph.ParagraphFormat.AfterSpacing = 0;

//Appends picture to the paragraph.

paragraph = table[0, 0].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Студенты");

//Appends paragraph.

paragraph = table[0, 1].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Балл");

paragraph = table[1, 0].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Студент 1");

//Appends paragraph.

paragraph = table[1, 1].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("5");

paragraph = table[2, 0].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Студент 2");

//Appends paragraph.

paragraph = table[2, 1].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("4.8");

paragraph = table[3, 0].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Студент 3");

//Appends paragraph.

paragraph = table[3, 1].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("4.5");

paragraph = table[4, 0].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Студент 4");

//Appends paragraph.

paragraph = table[4, 1].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("5");

paragraph = table[5, 0].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Студент 5");

//Appends paragraph.

paragraph = table[5, 1].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("3.8");

paragraph = table[6, 0].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("Студент 6");

//Appends paragraph.

paragraph = table[6, 1].AddParagraph();

paragraph.ApplyStyle("Heading 1");

paragraph.ParagraphFormat.AfterSpacing = 0;

paragraph.ParagraphFormat.LineSpacing = 12f;

paragraph.AppendText("4.8");

//Appends paragraph.

section.AddParagraph();

//Saves the Word document

document.Save("C:\\PR 3.docx");

Process wordProcess = new Process();

wordProcess.StartInfo.FileName = "C:\\PR 3.docx";

wordProcess.StartInfo.UseShellExecute = true;

wordProcess.Start();

}

}

}

**Вывод**: разработана программа, которая при нажатии на кнопку создает таблицу в MS Word.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 4**

**Тема: «Использование функций MS Excel»**

Студент: Бек-Назарова А. Н.

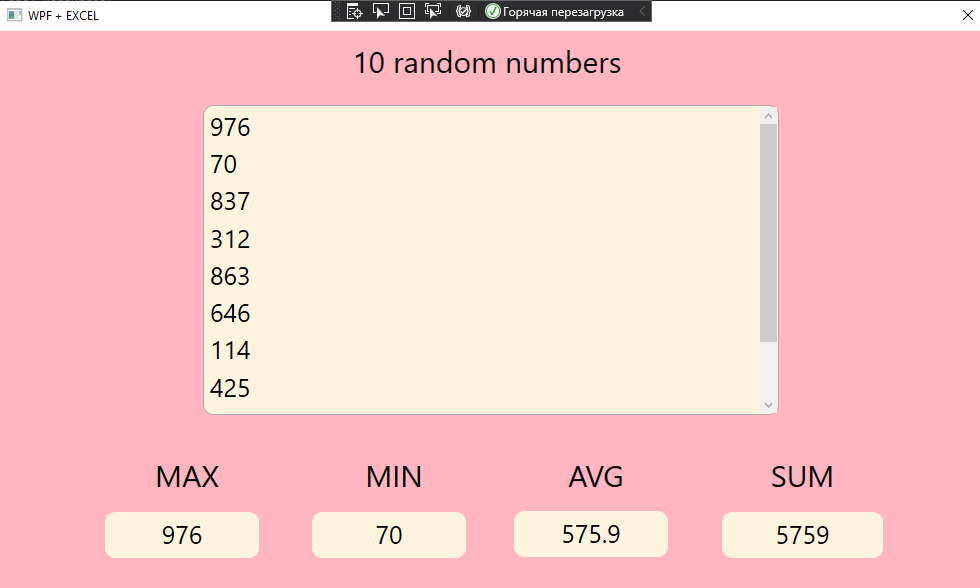
Группа: 3ПКС-420

Преподаватель: Сибирев И. В.

Преподаватель: Альшакова Е. Л.

Дата: 04.12.2022

**Цель работы**: необходимо разработать программу с использованием функций MS Excel не используя функцию нахождения числа Pi.



Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Вывод**: разработана программа с использованием функций MS Excel не используя функцию нахождения числа Pi.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 5**

**Тема: «Работа с файлами в различных кодировках»**

Студент: Бек-Назарова А. Н.

Группа: 3ПКС-420

Преподаватель: Сибирев И. В.

Преподаватель: Альшакова Е. Л.

Дата: 04.12.2022

**Цель работы**: написать программу на языке C#, которая реализует работу с файлами в различных кодировках.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Вывод**: написана программа на языке C#, которая реализует работу с файлами в различных кодировках.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 Программирование в компьютерных системах

**ОТЧЕТ**

по дисциплине ПМ.01.01 Учебная практика

*(указать вид (этап) практики)*

Профессиональный модуль ПМ.01. Разработка модулей программного обеспечения для компьютерных систем

Выполнил:

обучающийся учебной группы № 2ПКС-420

А. Н. Бек-Назарова

*(подпись) (И.О. Фамилия)*

Проверил:

Руководитель практики от колледжа:

Е. Л. Альшакова

*(подпись) (И.О. Фамилия)*

*(оценка) (подпись)*

**Москва – 2022**

**Перечень заданий/работ, выполненных в ходе учебной практики**

Специальность Программирование в компьютерных системах

*(наименование специальности)*

Профессиональный модуль ПМ.01.Разработка модулей программного обеспечения для компьютерных систем

*(наименование профессионального модуля)*

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Темы учебной практики** | **Выполненные задания/работы** |
| **1** | **Локальные данные процедур** | Дано описание: X DD 100 DUP(?) ; числа со знаком. Обнулить все отрицательные элементы массива X. |
| **2** | **Процедуры** | Ввести одномерный массив l = {7, 6, 15, 17, 12, -12, 4, 0, -10, -22}. Заменить в массиве нулями те элементы, модуль которых при делении на 5 дает в остатке 2. В полученном массиве найти максимальный элемент и его номер. Программу разработать на основе пользовательских функций. |
| **3** | **Рекурсивные процедуры** | Рекурсивно описать процедуру RevPrint(N), которая печатает в обратном порядке цифры десятичной записи целого неотрицательного числа N. Например, RevPrint(12345) должна вывести текст 54321. |
| **4** | **Обработка двумерных массивов** | Разместить в памяти компьютера матрицу 4\*3 целых чисел со знаком. Написать программу, позволяющую найти минимальный элемент среди элементов массива по каждой строке. Исходную матрицу и полученные значения вывести на экран. |
| **5** | **Обработка одномерных массивов** | Написать программу определения заданной характеристики последовательности чисел C1, C2, ..., Cn. Найти номер минимального числа. |
| **6** | **Рекурсивные процедуры** | Описать рекурсивную функцию degree5(N), определяющую, является ли натуральное число N степенью числа 5. Если N – степень пятёрки, то функция возвращает показатель этой степени, иначе функция выдаёт ответ -1. Например, degree5(50)= -1, degree5(125)=3, degree5(5)=1, degree5(1)=0. |
| **7** | **Работа со стеком** | Дана целочисленная прямоугольная матрица.  1) Определить количество столбцов, не содержащих ни одного нулевого элемента.  2) Найти сумму положительных четных элементов строки целочисленной матрицы. |

**Задание 1.**

Дано описание: X DD 100 DUP(?) ; числа со знаком. Обнулить все отрицательные элементы массива X.

Процедура обнуления отрицательных элементов:

proc zamena

mov esi,X

mov ecx,N

mov eax,0

mov ebx,0

metka:

cmp ebx,[esi]

jg metka1

jmp met

metka1:

mov [esi],ebx

met:

add esi,4

loop metka

ret

endp

Макросы для ввода и вывода массива:

macro prntn massiv, N

{

local metka

mov esi,massiv

mov ecx,[N]

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

}

macro input N, A

{

local metka

mov esi, A

mov ecx, [N]

cycl:

push ecx

cinvoke printf,text3

cinvoke scanf, fmat, K

pop ecx

mov ebx, [K]

mov [esi], ebx

add esi, 4

loop cycl

}

Основной код:

cinvoke printf,text1

cinvoke scanf, fmat, N

input N, X

cinvoke printf,text2

prntn X, N

call zamena

cinvoke printf,text2

prntn X, N

Примеры работы кода

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

**Задание 2.**

Ввести одномерный массив l = {7, 6, 15, 17, 12, -12, 4, 0, -10, -22}. Заменить в массиве нулями те элементы, модуль которых при делении на 5 дает в остатке 2. В полученном массиве найти максимальный элемент и его номер. Программу разработать на основе пользовательских функций.

Процедура поиска максимального числа и его номера:

proc maxim

lea ebx,[X]

mov ecx,[N]

mov edx,[N]

inc edx

mov eax,0

mov [ID],ecx

cycl:

cmp eax,[ebx]

jg cycl1

mov [ID],edx

sub [ID],ecx

mov eax,[ebx]

cycl1:

add ebx,4

loop cycl

ret

endp

Процедура замены в массиве нулями тех элементов, модуль которых при делении на 5 дает в остатке 2:

proc zamena

mov esi,X

mov ecx,[N]

mov edx,0

metka:

mov eax,[esi]

getabs:

neg eax ;находим модуль числа

js getabs

mov ebx,5

idiv ebx

mov ebx, 2

cmp ebx,edx

je metka1

jmp met

metka1:

mov eax, 0

mov [esi],eax

met:

mov edx,0

add esi,4

loop metka

ret

endp

Макрос для вывода массива:

macro prntn massiv, N

{

local metka

mov esi,massiv

mov ecx,[N]

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

}

Основной код:

cinvoke printf,text2

prntn X, N

call zamena

cinvoke printf,text2

prntn X, N

call maxim

cinvoke printf,text1,eax

cinvoke printf,text3,[ID]

Пример работы кода:

Изображение выглядит как текст

Автоматически созданное описание

**Задание 3.**

Рекурсивно описать процедуру RevPrint(N), которая печатает в обратном порядке цифры десятичной записи целого неотрицательного числа N. Например, RevPrint(12345) должна вывести текст 54321.

Процедура, которая печатает в обратном порядке цифры десятичной записи целого неотрицательного числа N:

proc fib

mov eax, [n]

mov ecx, 10

idiv ecx

mov [n], eax

push eax

cinvoke printf,text2,edx

pop eax

cmp eax,0

ja f1

ret

f1:

call fib

ret

endp

Основной код:

cinvoke printf,text1

cinvoke scanf, fmat, n

mov edx, 0

call fib

Примеры входных и выходных данных:

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

**Задание 4.**

Разместить в памяти компьютера матрицу 4\*3 целых чисел со знаком. Написать программу, позволяющую найти минимальный элемент среди элементов массива по каждой строке. Исходную матрицу и полученные значения вывести на экран.

Процедура на нахождение минимума в каждой строке:

proc minim

mov ecx,[lines]

m:

push ecx

lea ebx, [mas]

mov ecx,[columns]

cycl:

cmp ebx,[esi]

jl metka

mov ebx,[esi]

metka:

mov [min], ebx

add esi,4

loop cycl

mov eax,[min]

mov [edi], eax

add edi, 4

inc [counter]

pop ecx

loop m

ret

endp

Макрос на вывод двумерного массива:

macro prntn massiv,lines,columns

{

local metka

local met

mov esi,massiv

mov ecx,[lines]

met:

push ecx

mov ecx,[columns]

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

cinvoke printf,text3

pop ecx

loop met

}

cinvoke printf,text1,[lines],[columns]

prntnd mas,lines,columns

cinvoke printf,text4

mov esi,mas

mov edi,ismas

call minim

prntn ismas,lines

Пример входных и исходный данных:

Изображение выглядит как текст

Автоматически созданное описание

**Задание 5.**

Написать программу определения заданной характеристики последовательности чисел C1, C2, ..., Cn. Найти номер минимального числа.

Процедура, которая находит минимальный элемент в последовательности и индекс этого элемента:

proc ind

lea ebx,[mas]

mov ecx,[N]

lea eax,[mas]

cycl:

mov edx, [N]

cmp eax,[ebx]

jl metka

mov eax,[ebx]

sub edx, ecx

mov [index], edx

metka:

mov [Z], eax

add ebx,4

loop cycl

ret

endp

Основной код:

cinvoke printf,text1

cinvoke scanf, fmat, N

prntn mas, N

call ind

cinvoke printf,text2,[Z]

cinvoke printf,text3,[index]

Макрос, выводящий последовательность в консоль:

macro prntn massiv, Nom

{

local metka

mov esi,massiv

mov ecx,[Nom]

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

}

Примеры входных и выходных данных:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Задание 6.**

Описать рекурсивную функцию degree5(N), определяющую, является ли натуральное число N степенью числа 5.

Если N – степень пятёрки, то функция возвращает показатель этой степени, иначе функция выдаёт ответ -1. Например, degree5(50)= -1, degree5(125)=3, degree5(5)=1, degree5(1)=0.

Рекурсивная функция, определяющая, является ли натуральное число N степенью числа 5:

proc degree5

mov ebx, 5

cmp eax, 5

jge f2

cmp edx, 0

je f3

mov [answ],-1

f3:

ret

f2:

mov edx, 0

idiv ebx

inc [answ]

call degree5

ret

endp

Основной код:

cinvoke printf,text1

cinvoke scanf, fmat, N

mov eax, [N]

cmp eax, 1

je cd

call degree5

cd:

cinvoke printf,text4,[N],[answ]

Примеры входных и выходных данных:

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст, ударение

Автоматически созданное описание

**Задание 7.**

Дана целочисленная прямоугольная матрица.

1) Определить количество столбцов, не содержащих ни одного нулевого элемента.

2) Найти сумму положительных четных элементов строки целочисленной матрицы.

Процедура на нахождение количества столбцов, не содержащих ни одного нулевого элемента:

proc nul

mov ecx,[columns]

m:

push ecx

mov eax, 0

mov ebx, 0

mov ecx,[lines]

cycl:

cmp [esi],eax

je metka

inc ebx

metka:

mov [min], ebx

add esi,4

loop cycl

cmp ebx,[lines]

jne dob

inc [ind]

dob:

pop ecx

loop m

cinvoke printf,text2,[ind]

ret

endp

Процедура на нахождение суммы положительных четных элементов:

proc summ

mov ecx,[lines]

su:

push ecx

mov ecx,[columns]

mov eax, 0

mov [s], 0

su1:

mov edx, 0

cmp [esi],edx

jle sum

mov ebx,2

mov eax, [esi]

idiv ebx

cmp edx,0

jne sum

mov eax, [s]

add eax, [esi]

mov [s], eax

sum:

add esi,4

loop su1

cinvoke printf,text4,eax

pop ecx

loop su

ret

endp

Основной код:

cinvoke printf,text1,[lines],[columns]

prntnd mas,lines,columns

mov esi,mas

call nul

cinvoke printf,text4

mov edi,ismas

mov esi,mas

call summ

prntn ismas,lines

Макрос на вывод двумерного массива:

macro prntn massiv,lines,columns

{

local metka

local met

mov esi,massiv

mov ecx,[lines]

met:

push ecx

mov ecx,[columns]

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

cinvoke printf,text3

pop ecx

loop met

}

Входные и выходные данные:

Изображение выглядит как текст

Автоматически созданное описание

Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 Программирование в компьютерных системах

**ОТЧЕТ**

по дисциплине ПМ.01.01 Учебная практика

*(указать вид (этап) практики)*

Профессиональный модуль ПМ.01. Разработка модулей программного обеспечения для компьютерных систем

Выполнил:

обучающийся учебной группы № 2ПКС-420

А. Н. Бек-Назарова

*(подпись) (И.О. Фамилия)*

Проверил:

Руководитель практики от колледжа:

Г.И. Киреева

*(подпис ) (И.О. Фамилия)*

*(оценка) (подпись)*

**Москва – 2022**

**Задание 1.** Требуется в консольном проекте создать двумерный числовой массив, заполнить его случайными целыми числами в диапазоне от 10 до 99.

Вывести массив в консоль, установив разный цвет текста для следующих зон массива:

* Зоны главной и побочной диагоналей массива
* Зона выше главной и выше побочной диагоналей (верхний треугольник)
* Зона ниже главной и ниже побочной диагоналей (нижний треугольник)
* Зона выше главной и ниже побочной диагоналей (правый треугольник)
* Зона ниже главной и выше побочной диагоналей (левый треугольник).

Для каждой из зон вычислить следующие значения: sum, min, max, average.

Вывести в консоль эти значения с комментариями, используя для каждого из них такой же цвет текста, как и у соответствующей зоны массива.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

using System;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите размер матрицы: ");

int N = Convert.ToInt32(Console.ReadLine());

int[][] massiv = new int[N][];

int[] max = new int[6] { 0, 0, 0, 0, 0, 0 };

int[] min = new int[6] { 100, 100, 100, 100, 100, 100 };

int[] sum = new int[6] { 0, 0, 0, 0, 0, 0 };

int[] arr = new int[6] { N, N, 0, 0, 0, 0 };

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

{

massiv[i] = new int[N];

}

Console.CursorTop = 5;

Console.CursorLeft = 50;

Console.WriteLine("Разноцветный массив\n");

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

{

Console.CursorLeft = 45;

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

{

Random rnd = new Random();

int element = rnd.Next(10, 100);

if (i == j)

{

Console.ForegroundColor = ConsoleColor.Red; //главная диагональ

if (element > max[0]) max[0] = element;

if (element < min[0]) min[0] = element;

sum[0] += element;

}

else if ((i + j) == (N - 1))

{

Console.ForegroundColor = ConsoleColor.Yellow; // побочная диагональ

if (element > max[1]) max[1] = element;

if (element < min[1]) min[1] = element;

sum[1] += element;

}

else if ((i < j) && ((i + j) < (N - 1)))

{

Console.ForegroundColor = ConsoleColor.Green;

if (element > max[2]) max[2] = element;

if (element < min[2]) min[2] = element;

sum[2] += element;

arr[2]++;

}

else if ((i < j) && ((i + j) > (N - 1)))

{

Console.ForegroundColor = ConsoleColor.Blue;

if (element > max[3]) max[3] = element;

if (element < min[3]) min[3] = element;

sum[3] += element;

arr[3]++;

}

else if ((i > j) && ((i + j) > (N - 1)))

{

Console.ForegroundColor = ConsoleColor.Cyan;

if (element > max[4]) max[4] = element;

if (element < min[4]) min[4] = element;

sum[4] += element;

arr[4]++;

}

else if ((i > j) && ((i + j) < (N - 1)))

{

Console.ForegroundColor = ConsoleColor.White;

if (element > max[5]) max[5] = element;

if (element < min[5]) min[5] = element;

sum[5] += element;

arr[5]++;

}

Console.Write(element + " ");

massiv[i][j] = element;

}

Console.WriteLine();

}

Console.WriteLine("\n");

for (int i = 0; i < 6; i++)

{

Console.CursorLeft = 45;

switch (i)

{

case 0:

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("\tГлавная диагональ: ");

break;

case 1:

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("\tПобочная диагональ: ");

break;

case 2:

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("\tВерхняя зона: ");

break;

case 3:

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("\tПравая зона: ");

break;

case 4:

Console.ForegroundColor = ConsoleColor.Cyan;

Console.WriteLine("\tНижняя зона: ");

break;

case 5:

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("\tЛевая зона: ");

break;

}

Console.CursorLeft = 42;

Console.WriteLine("Максимальный элемент: " + max[i]);

Console.CursorLeft = 42;

Console.WriteLine("Минимальный элемент: " + min[i]);

Console.CursorLeft = 42;

Console.WriteLine("Сумма элементов: " + sum[i]);

arr[i] = sum[i] / arr[i];

Console.CursorLeft = 42;

Console.WriteLine("Среднее арифметическое элементов: " + arr[i] + "\n");

}

Console.ForegroundColor = ConsoleColor.White;

}

}

}

**Задание 2.**

Создать форму для вычисления функции, под ней поле с функцией   
Изображение выглядит как текст

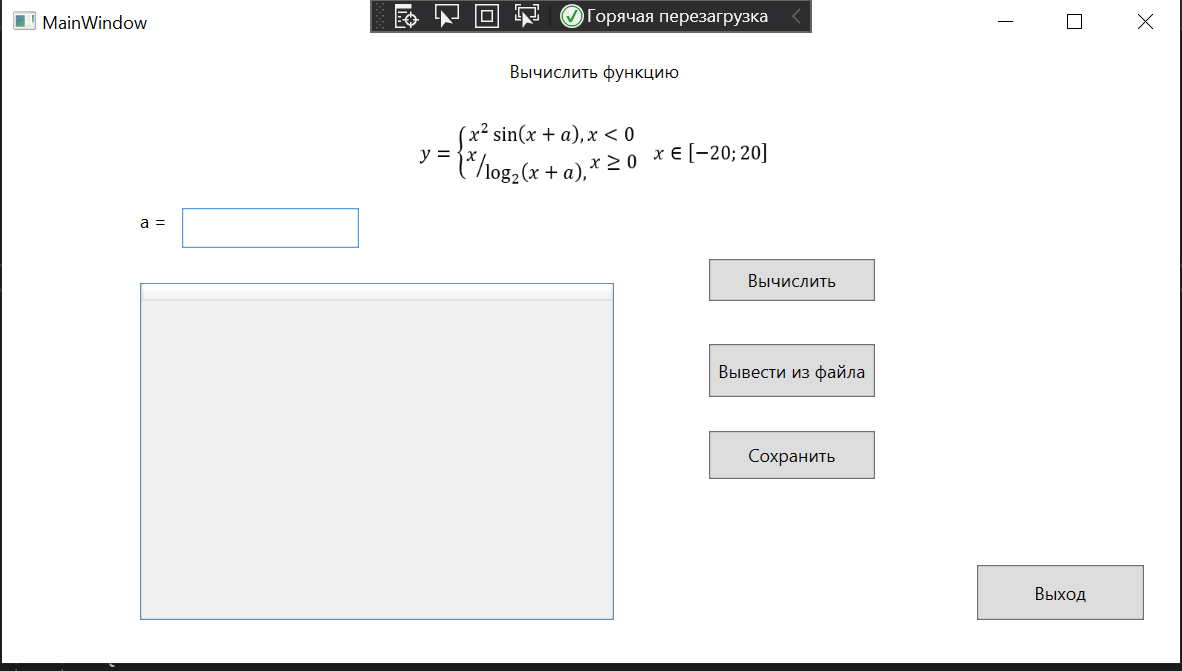
Автоматически созданное описание

«а» вводится пользователем, при нажатии на кнопку вычислить – вычисляется функция.

Сетка Grid под этим, в которой столбцы x | y со значениями

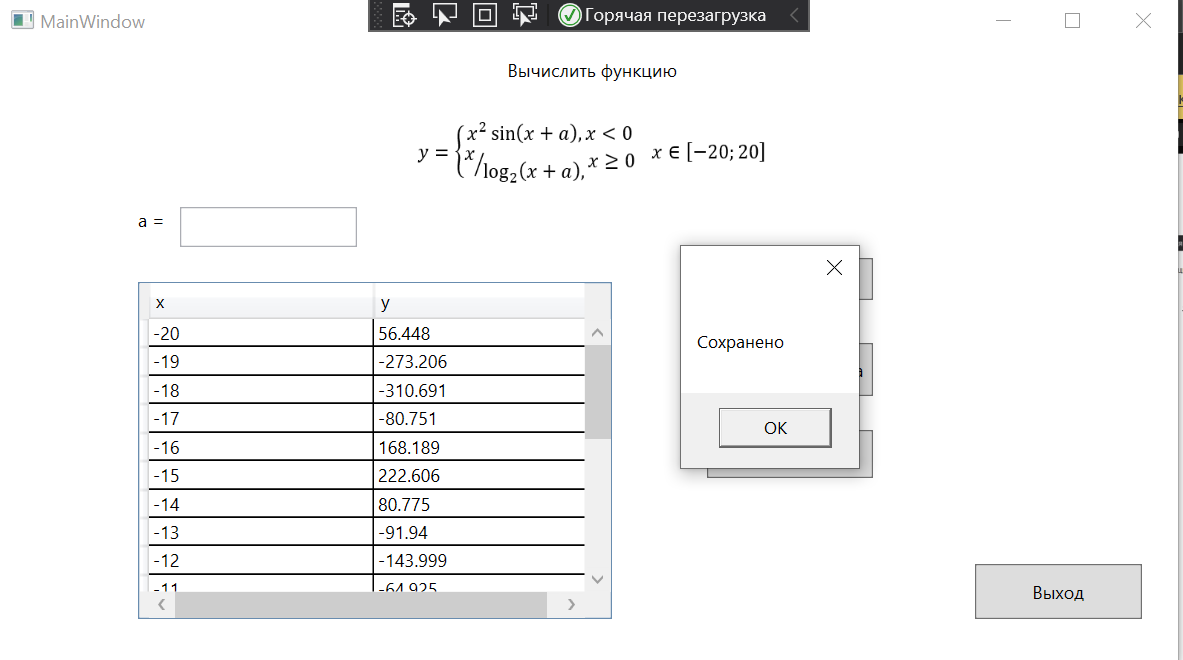
под кнопкой вычислить кнопка записать, она записывает в файл всю таблицу Grid под кнопкой записать кнопка считать, она считывает из файла,

кнопка выхода.



Изображение выглядит как стол

Автоматически созданное описание



Изображение выглядит как стол

Автоматически созданное описание

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApp2

{

public class vichis

{

public int x { get; set; }

public double y { get; set; }

}

public partial class MainWindow : Window

{

int a;

double kaknibyd = 0;

List<vichis> listvich = new List<vichis>();

List<vichis> listvivod = new List<vichis>();

string patch = "C:\\Users\\bekna\\source\\repos\\WpfApp2\\data.txt";

public MainWindow()

{

InitializeComponent();

inputA.Focus();

}

private void BringOut(object sender, RoutedEventArgs e)

{

using (StreamReader reader = new StreamReader(patch))

{

string line;

string xx;

string yy;

while ((line = reader.ReadLine()) != null)

{

xx = line.Split().First();

yy = line.Split().Last();

listvivod.Add(new vichis { x = Convert.ToInt32(xx), y = Convert.ToDouble(yy) });

}

DataGrid.ColumnWidth = 150;

DataGrid.ItemsSource = listvivod;

}

}

private void Save(object sender, RoutedEventArgs e)

{

using (StreamWriter write = new StreamWriter(patch, false)) {

for(int i = 0; i < listvich.Count; i++)

{

write.WriteLineAsync(listvich[i].x.ToString() + " " + listvich[i].y.ToString());

}

MessageBox.Show("Сохранено");

write.Close();

}

DataGrid.ColumnWidth = 150;

DataGrid.DataContext = null;

DataGrid.Columns.Clear();

}

private void Calculate(object sender, RoutedEventArgs e)

{

if (inputA.Text == "")

{

MessageBox.Show("Значение 'а' не введено, по умолчанию 'a' = 0");

a = 0;

}

else

{

a = Convert.ToInt32(inputA.Text);

}

inputA.Clear();

for(int i = -20; i<0; i++)

{

kaknibyd = Math.Round(i\*i\*(Math.Sin(i+a)),3);

listvich.Add(new vichis { x = i , y = kaknibyd }) ;

}

for (int i = 0; i <= 20; i++)

{

if ((i - a) >= 0)

{

if (Math.Log(i + a) != 0)

{

kaknibyd = Math.Round(i / Math.Log(i + a),3);

listvich.Add(new vichis { x = i, y = kaknibyd });

}

}

}

DataGrid.ColumnWidth = 150;

DataGrid.ItemsSource = listvich;

}

private void Exit(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

<Window x:Class="WpfApp2.MainWindow"

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-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<TextBox x:Name="inputA" HorizontalAlignment="Left" Height="27" Margin="120,109,0,0" TextWrapping="Wrap" Text = "" VerticalAlignment="Top" Width="118"/>

<TextBlock HorizontalAlignment="Left" Height="27" Margin="92,110,0,0" TextWrapping="Wrap" Text="а = " VerticalAlignment="Top" Width="45"/>

<Button Content="Вычислить" HorizontalAlignment="Left" Height="28" Margin="471,143,0,0" VerticalAlignment="Top" Width="111" Click="Calculate"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Height="32" Margin="471,258,0,0" VerticalAlignment="Top" Width="111" Click="Save"/>

<Button Content="Выход" HorizontalAlignment="Left" Height="37" Margin="650,347,0,0" VerticalAlignment="Top" Width="111" Click="Exit"/>

<TextBlock HorizontalAlignment="Left" Height="25" Margin="10,10,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="773" TextAlignment="Center"> <Run Text="Вычислить функцию"/><Run Text=" "/><LineBreak/></TextBlock>

<Button Content="Вывести из файла" HorizontalAlignment="Left" Height="35" Margin="471,200,0,0" VerticalAlignment="Top" Width="111" Click="BringOut"/>

<DataGrid x:Name="DataGrid" AutoGenerateColumns="True" HorizontalAlignment="Left" Height="225" Margin="92,159,0,0" VerticalAlignment="Top" Width="316"/>

<Image Margin="256,35,259.333,306.667" Source="Image1.png" Stretch="Fill"/>

</Grid>

</Window>

**Задание 3.**

- добавить окно ввода игры, обработать исключения(пробелы и т.д.)

- кнопка добавить, она добавляет в ListBox введённое предложение

- кнопка изменить, выбрать страну в ListBox, при нажатии на кнопку открывается изменение этого слова с мигающим курсором

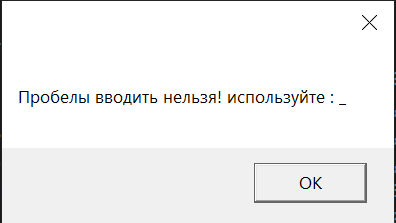
- кнопка удалить, также нужно выбрать и нажать кнопку, появится окно подтверждения с да/нет

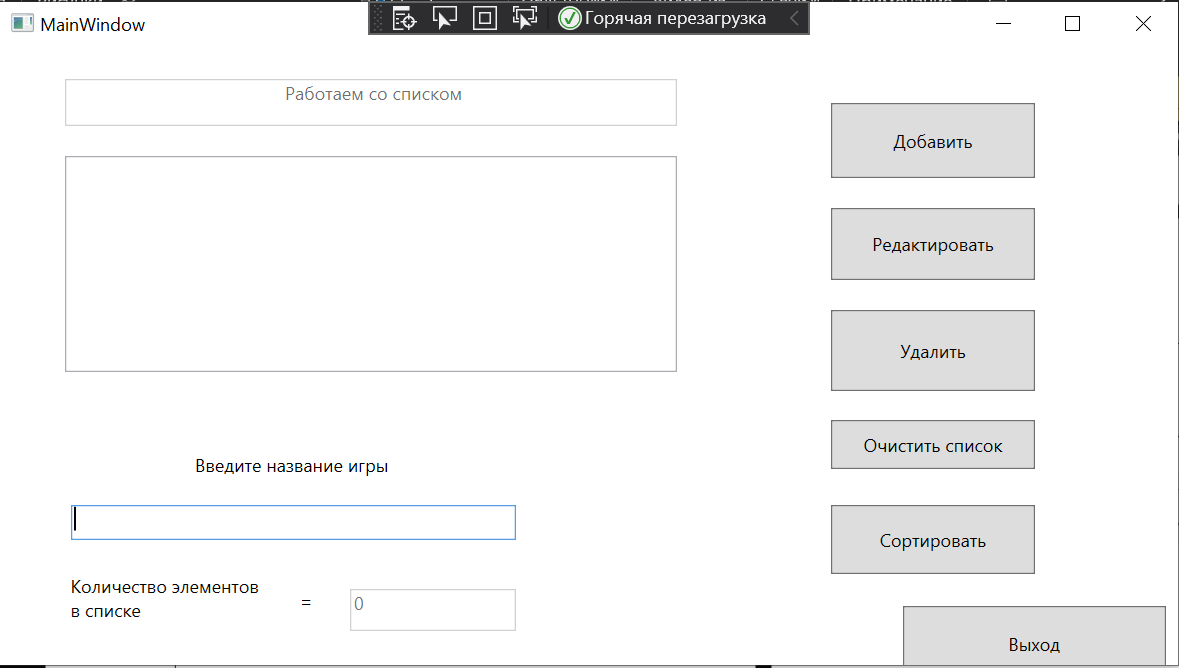
- очистить список, то же самое, что и удаление, но предупреждение с восклицательным знаком

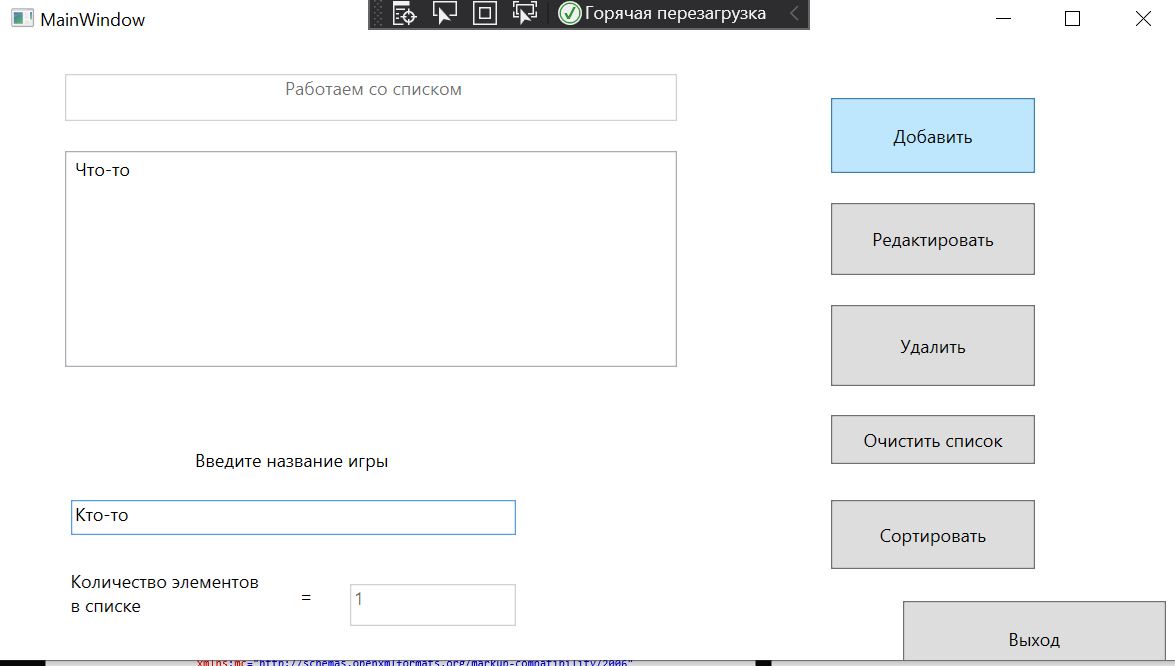
- сортировку она не показала, но думаю просто по алфавиту

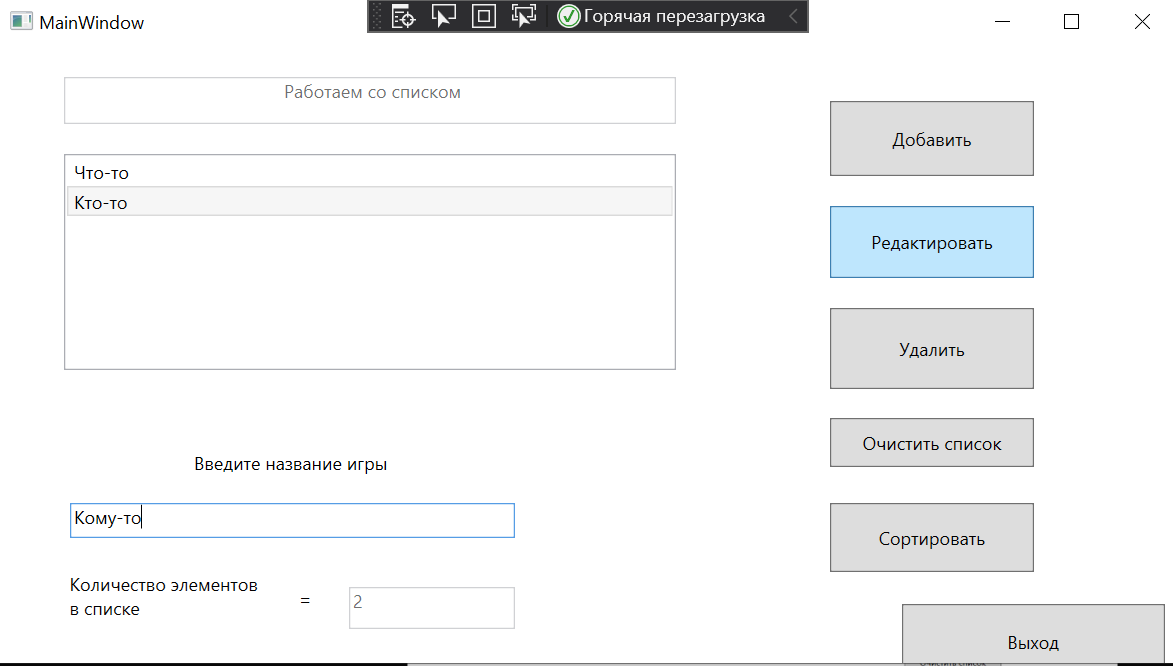
- счетчик записей под окном ввода

- кнопка выхода



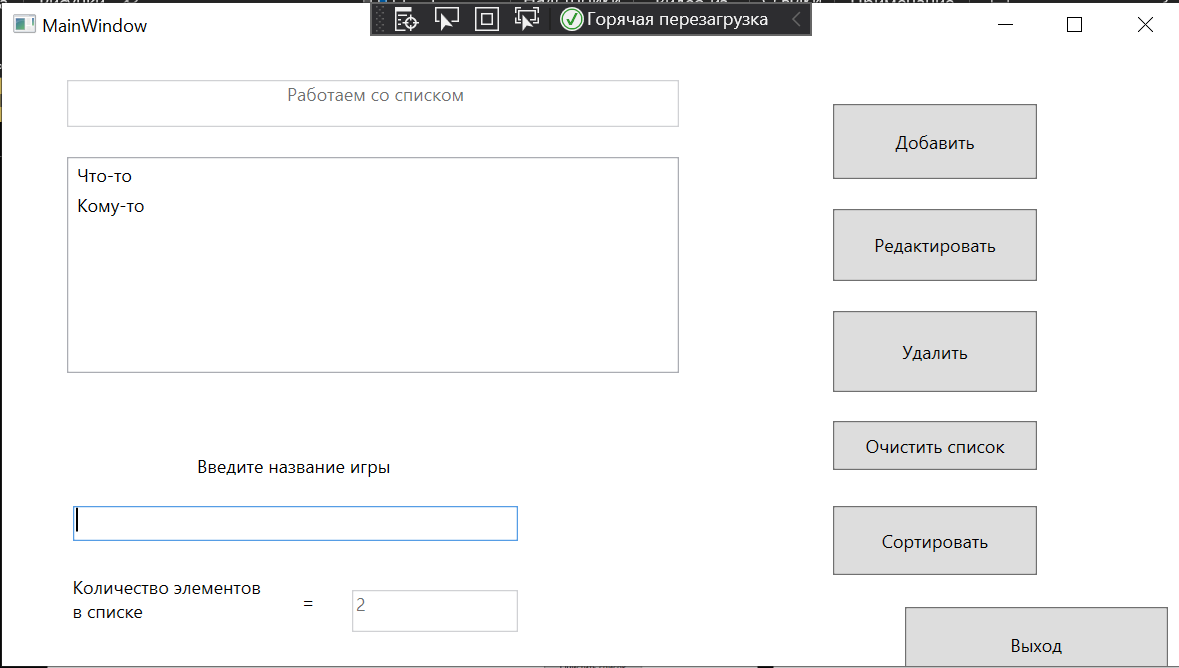


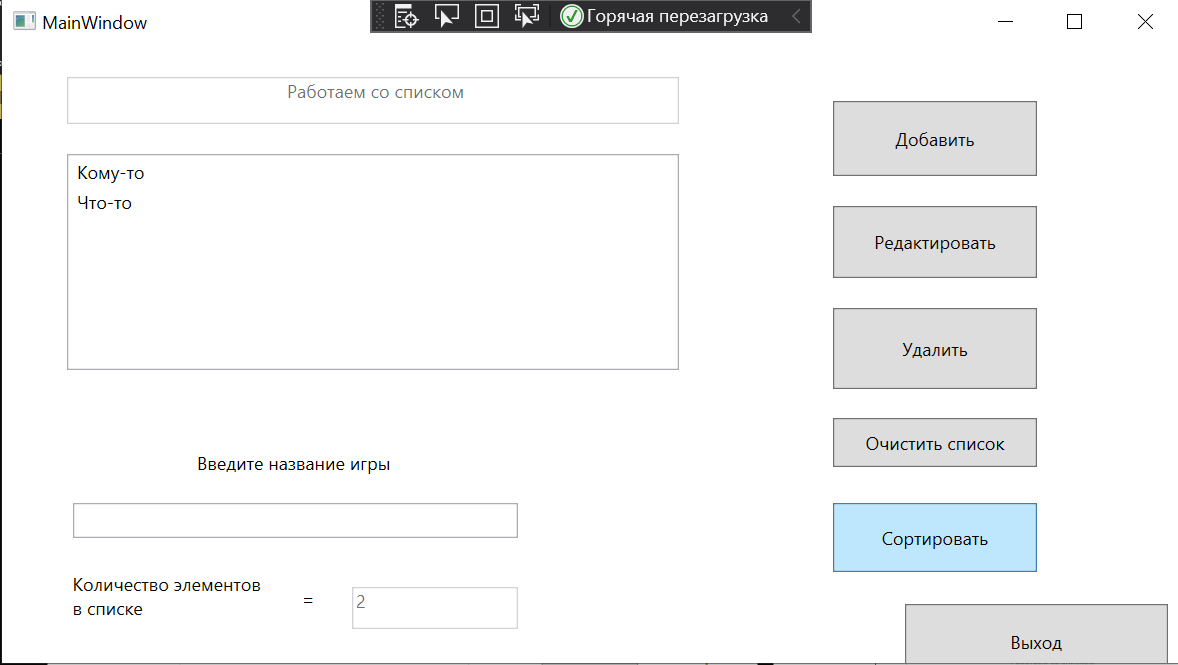


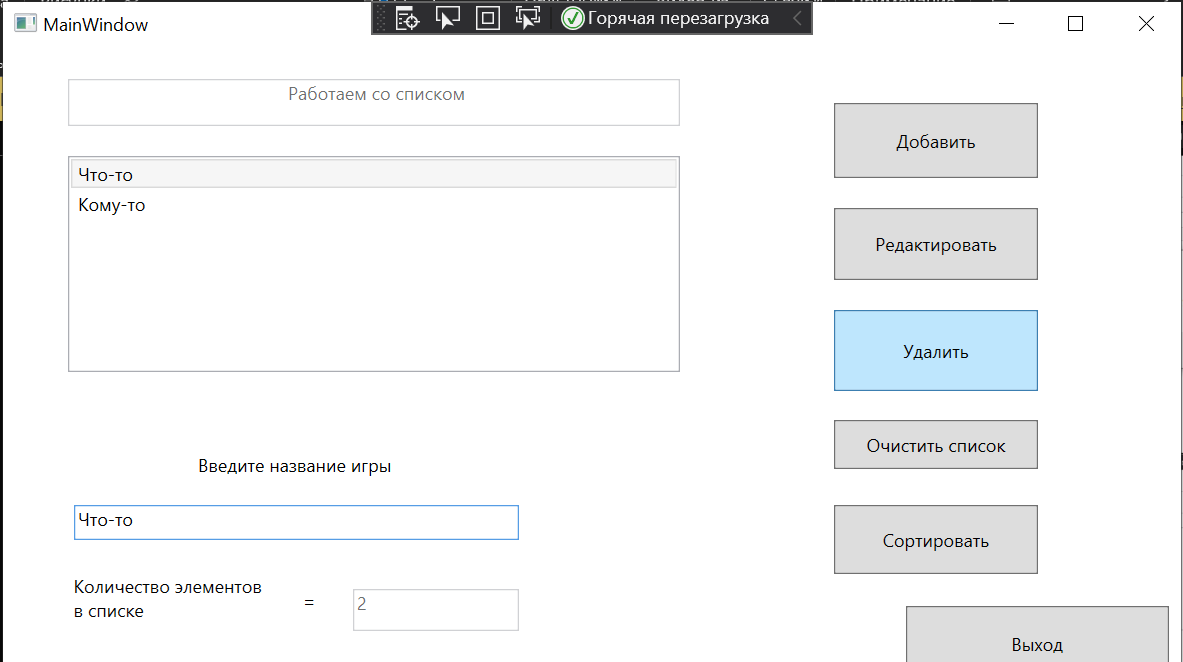


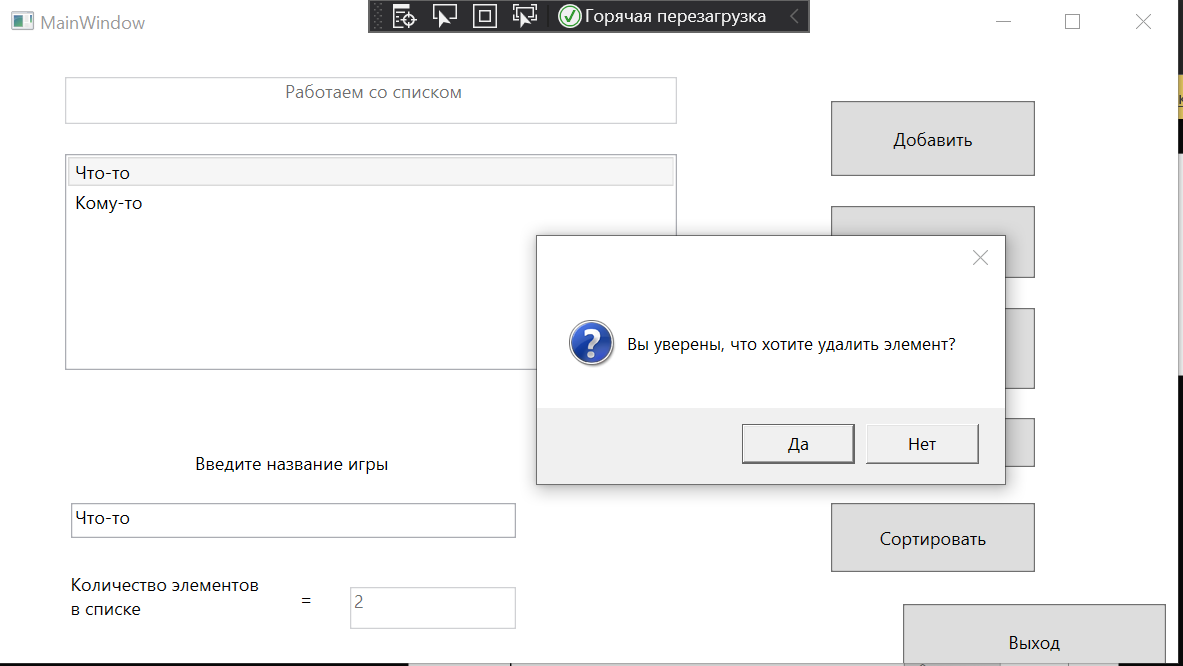
Изображение выглядит как текст

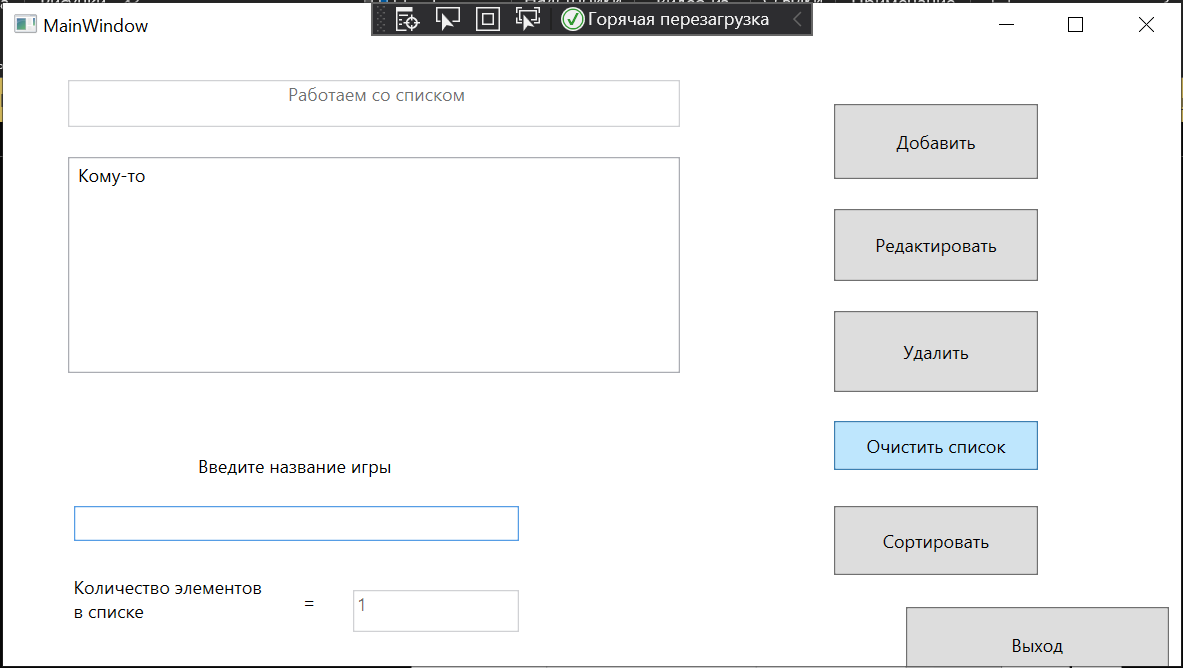
Автоматически созданное описание











using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Text.RegularExpressions;

namespace WpfApp4

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

input\_Info.Focus();

Kolvo.IsEnabled = false;

nazvanie.IsEnabled = false;

MessageBoxResult res = MessageBox.Show("Пробелы вводить нельзя! используйте : \_");

}

private void ListBox\_SelectionChanged(object sender, EventArgs e)

{

try

{

input\_Info.Text = List\_Info.SelectedItem.ToString();

input\_Info.Focus();

input\_Info.SelectionStart = input\_Info.Text.Length;

}

catch { }

}

private void Add\_unit(object sender, RoutedEventArgs e)

{

if (input\_Info.Text != "")

{

if (input\_Info.SelectionStart == 0) { e.Handled = true; }

string b = input\_Info.Text.TrimStart();

List\_Info.Items.Add(b);

Kolvo.Text = List\_Info.Items.Count.ToString();

}

input\_Info.Clear();

input\_Info.Focus();

}

private void Change\_Info(object sender, RoutedEventArgs e)

{

var ind = List\_Info.SelectedIndex;

if (ind != -1)

{

MessageBoxButton buttons = MessageBoxButton.YesNo;

MessageBoxResult res = MessageBox.Show("Вы уверены, что хотите изменить элемент?", "", buttons, MessageBoxImage.Question);

switch (res)

{

case MessageBoxResult.Yes:

List\_Info.Items.RemoveAt(ind);

if (input\_Info.Text != "")

{

List\_Info.Items.Insert(ind, input\_Info.Text);

}

Kolvo.Text = List\_Info.Items.Count.ToString();

input\_Info.Clear();

break;

case MessageBoxResult.No:

input\_Info.Clear();

break;

}

}

else

{

MessageBoxResult res = MessageBox.Show("Выберите элемент, который хотите редактировать!", null, MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void Delete\_Info(object sender, RoutedEventArgs e)

{

int selectedindx = List\_Info.SelectedIndex;

if (selectedindx != -1)

{

MessageBoxButton buttons = MessageBoxButton.YesNo;

MessageBoxResult res = MessageBox.Show("Вы уверены, что хотите удалить элемент?", "", buttons, MessageBoxImage.Question);

switch (res)

{

case MessageBoxResult.Yes:

List\_Info.Items.RemoveAt(selectedindx);

Kolvo.Text = List\_Info.Items.Count.ToString();

input\_Info.Clear();

break;

case MessageBoxResult.No:

break;

}

}

else

{

MessageBoxResult res = MessageBox.Show("Выберите элемент, который хотите удалить!",null,MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void CLear\_List(object sender, RoutedEventArgs e)

{

MessageBoxButton buttons = MessageBoxButton.YesNo;

MessageBoxResult res = MessageBox.Show("Вы уверены, что хотите очистить список?","",buttons,MessageBoxImage.Question);

switch (res)

{

case MessageBoxResult.Yes:

List\_Info.Items.Clear();

Kolvo.Text = List\_Info.Items.Count.ToString();

break;

case MessageBoxResult.No:

break;

}

}

private void Sort\_List(object sender, RoutedEventArgs e)

{

List\_Info.Items.SortDescriptions.Add(

new System.ComponentModel.SortDescription("",

System.ComponentModel.ListSortDirection.Ascending));

}

private void Exit\_Programm(object sender, RoutedEventArgs e)

{

Close();

}

private void input\_Info\_TextChanged(object sender, TextChangedEventArgs e)

{

(sender as TextBox).Text = Regex.Replace((sender as TextBox).Text, @"\s+", "");

input\_Info.Focus();

input\_Info.SelectionStart = input\_Info.Text.Length;

}

}

}

<Window x:Class="WpfApp4.MainWindow"

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-compatibility/2006"

xmlns:local="clr-namespace:WpfApp4"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<TextBox Name="nazvanie" HorizontalAlignment="Left" Height="31" Margin="43,22,0,0" Text="&#x9;&#x9;&#x9;Работаем со списком" TextWrapping="Wrap" VerticalAlignment="Top" Width="408"/>

<ListBox x:Name="List\_Info" HorizontalAlignment="Left" Height="144" Margin="43,73,0,0" VerticalAlignment="Top" Width="408" SelectionChanged="ListBox\_SelectionChanged"/>

<Button Content="Добавить" HorizontalAlignment="Left" Height="50" Margin="554,38,0,0" VerticalAlignment="Top" Width="136" Click="Add\_unit"/>

<Button Content="Редактировать" HorizontalAlignment="Left" Height="48" Margin="554,108,0,0" VerticalAlignment="Top" Width="136" RenderTransformOrigin="0.48,-0.725" Click="Change\_Info"/>

<Button Content="Удалить" HorizontalAlignment="Left" Height="54" Margin="554,176,0,0" VerticalAlignment="Top" Width="136" Click="Delete\_Info"/>

<Button Content="Очистить список" HorizontalAlignment="Left" Height="33" Margin="554,249,0,0" VerticalAlignment="Top" Width="136" Click="CLear\_List"/>

<Button Content="Cортировать" HorizontalAlignment="Left" Height="46" Margin="554,306,0,0" VerticalAlignment="Top" Width="136" Click="Sort\_List"/>

<Button Content="Выход" HorizontalAlignment="Left" Height="51" Margin="602,373,0,0" VerticalAlignment="Top" Width="175" Click="Exit\_Programm"/>

<TextBox Name="input\_Info" HorizontalAlignment="Left" Height="23" Margin="47,306,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="297" TextChanged="input\_Info\_TextChanged"/>

<Label Content="&#x9;Введите название игры" HorizontalAlignment="Left" Height="30" Margin="77,266,0,0" VerticalAlignment="Top" Width="245"/>

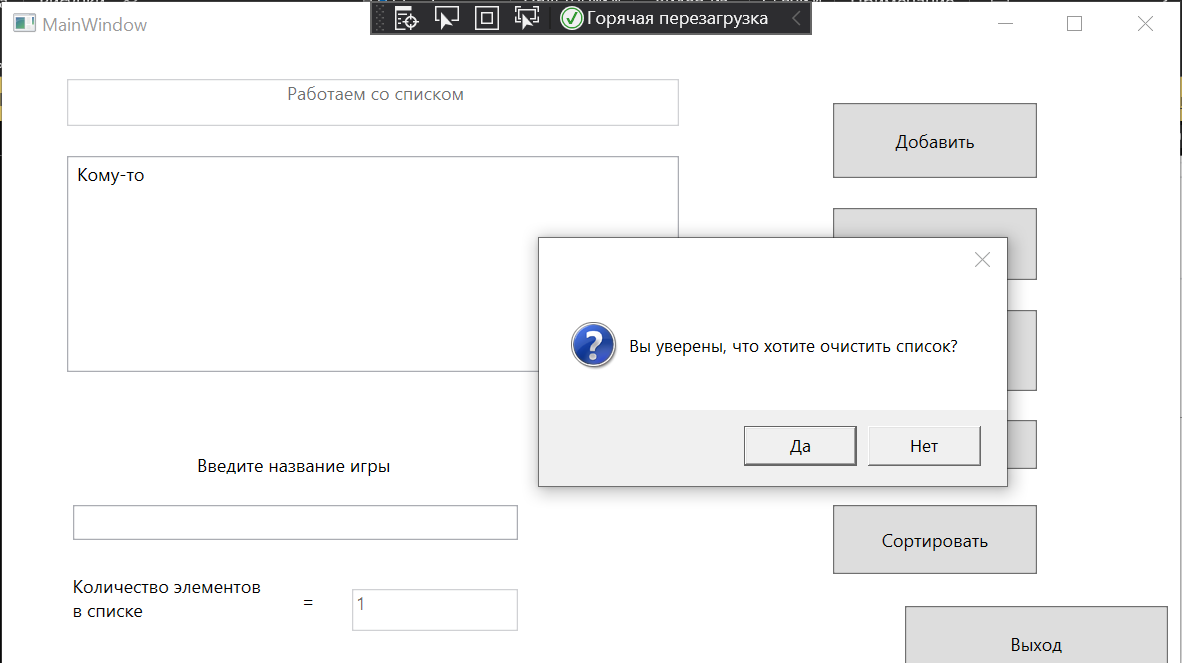
<TextBlock HorizontalAlignment="Left" Height="38" Margin="47,352,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="200"><Run Language="ru-ru" Text="Количество элементов"/><LineBreak/><Run Language="ru-ru" Text="в списке"/></TextBlock>

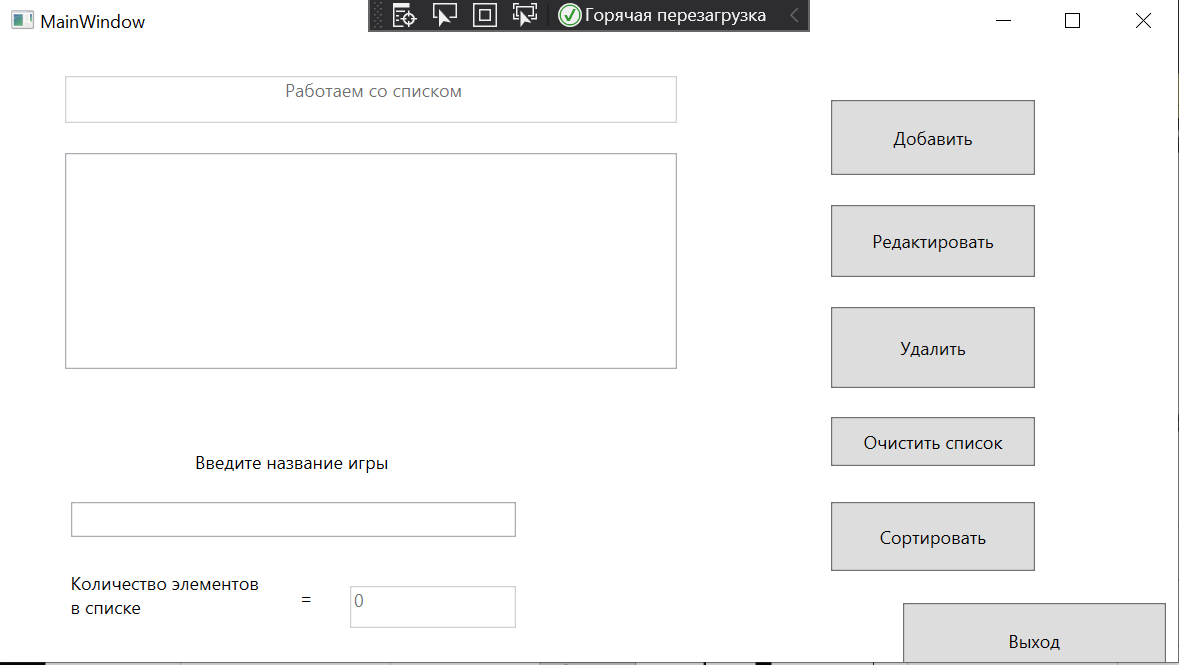
<TextBlock HorizontalAlignment="Left" Height="18" Margin="200,362,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="66"><Run Language="ru-ru" Text="="/></TextBlock>

<TextBox x:Name="Kolvo" HorizontalAlignment="Left" Height="28" Margin="233,362,0,0" Text="0" TextWrapping="Wrap" VerticalAlignment="Top" Width="111" />

</Grid>

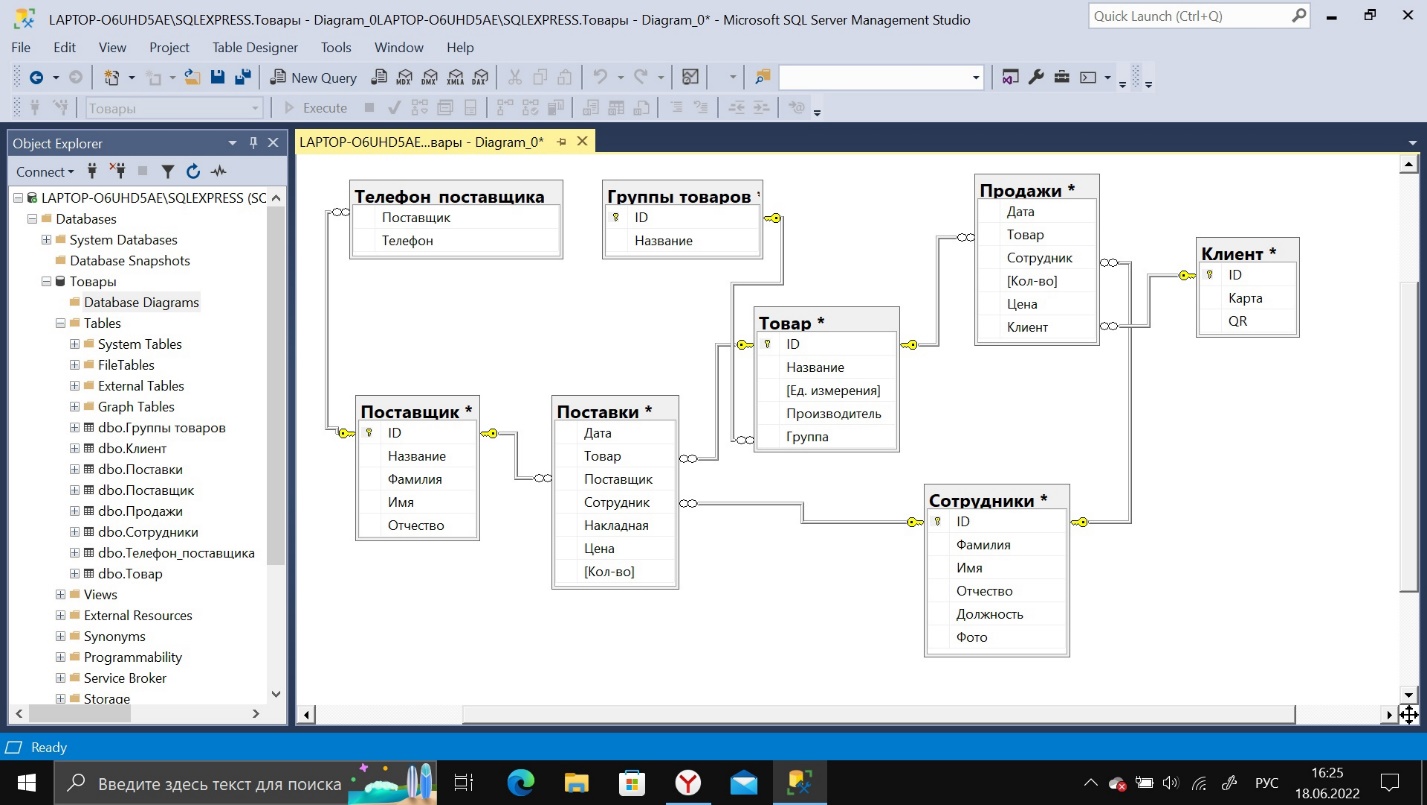
</Window>

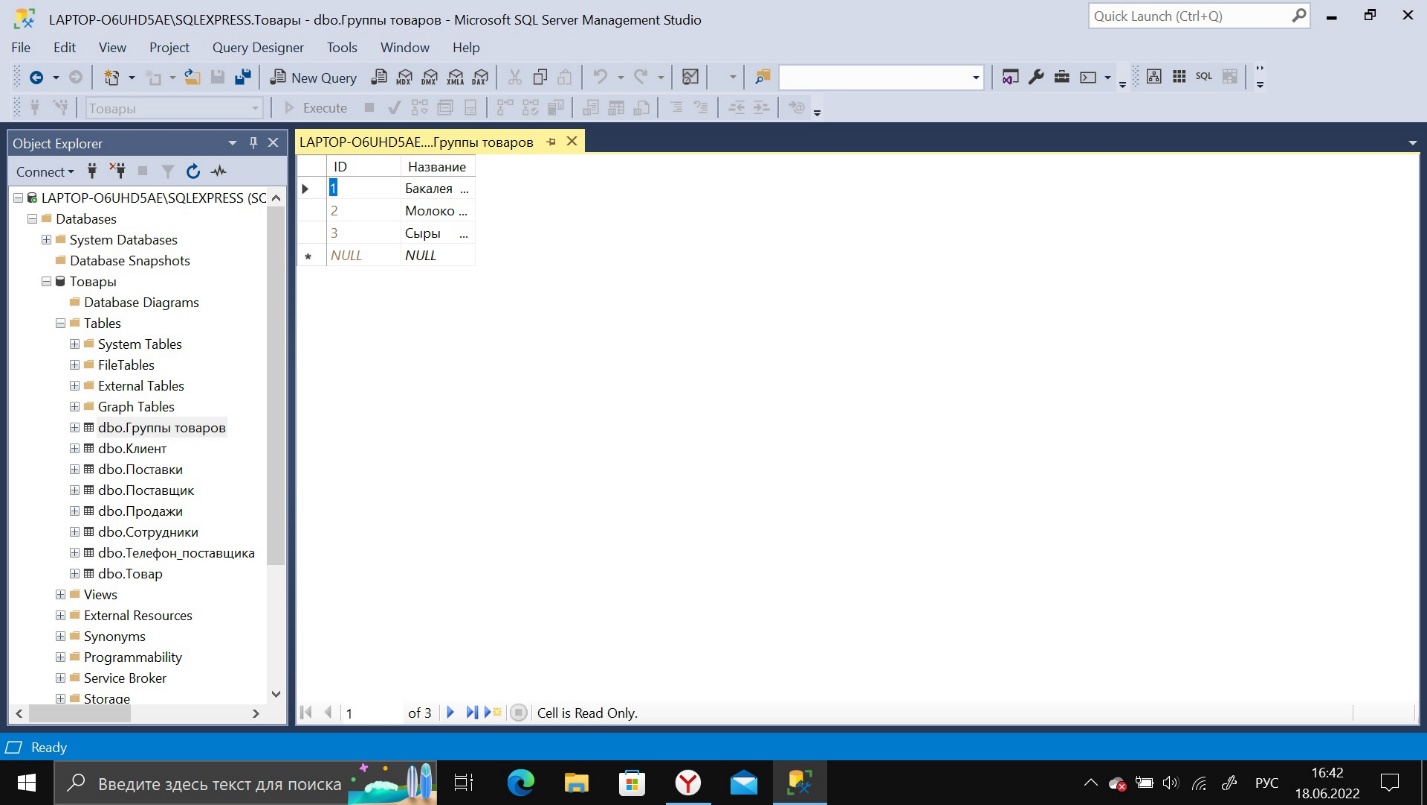


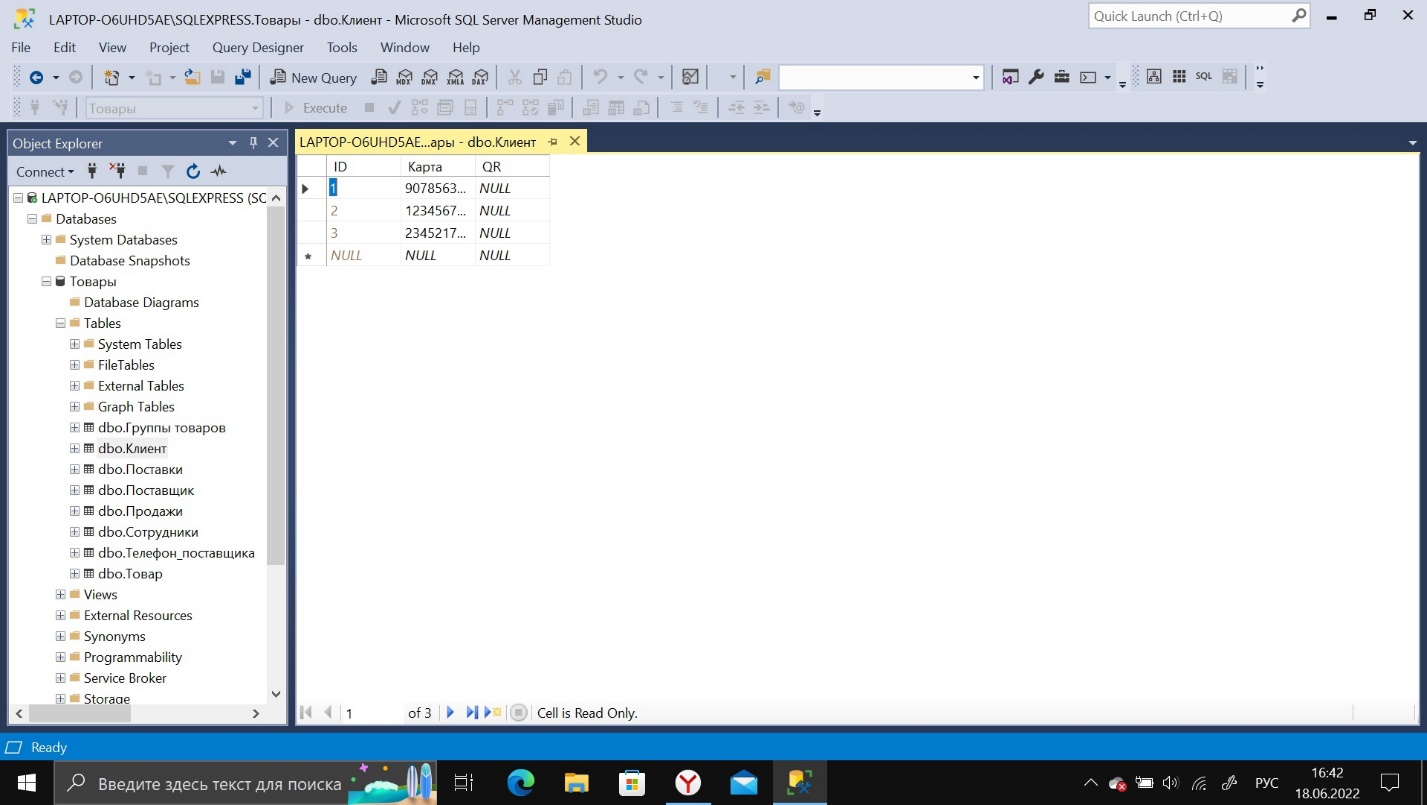


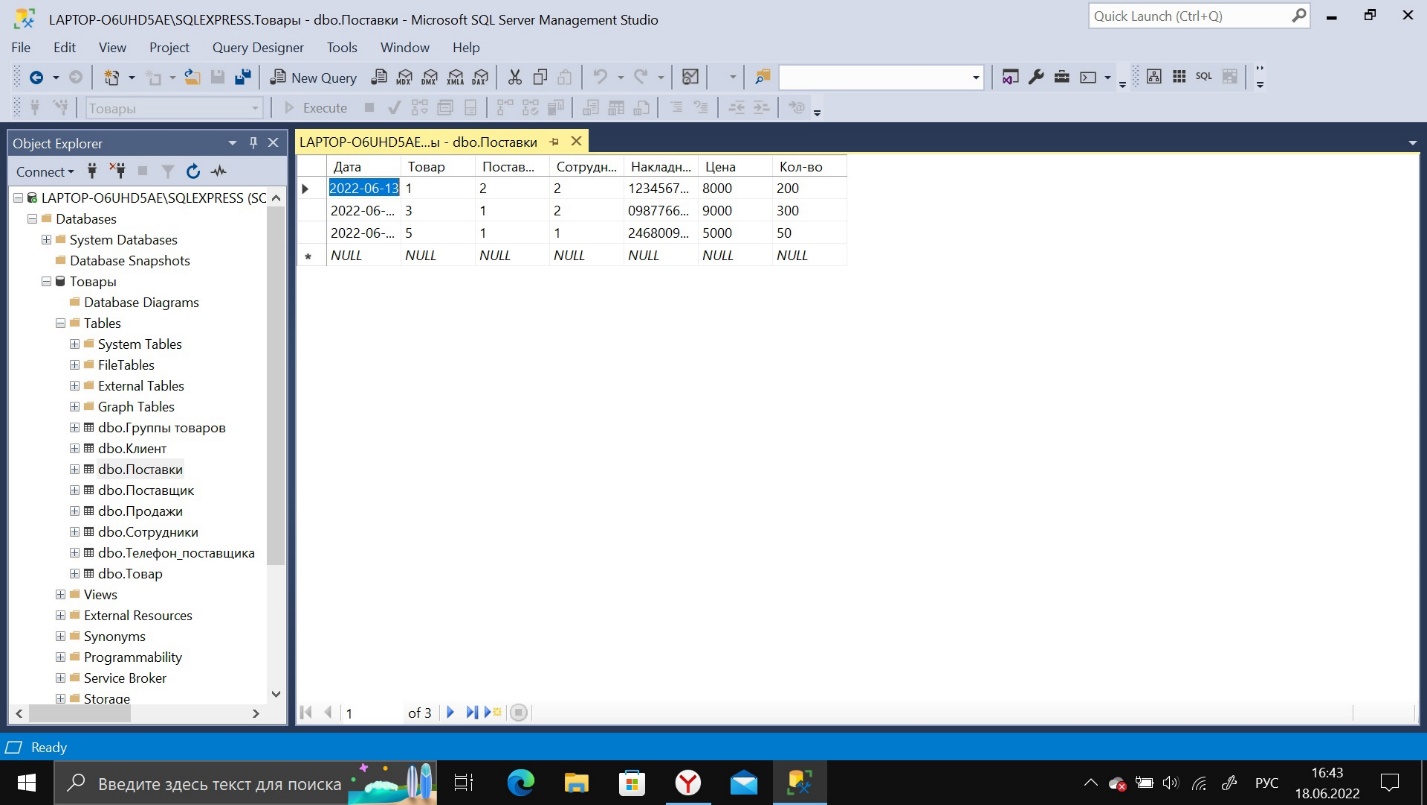
**Задание 4.**

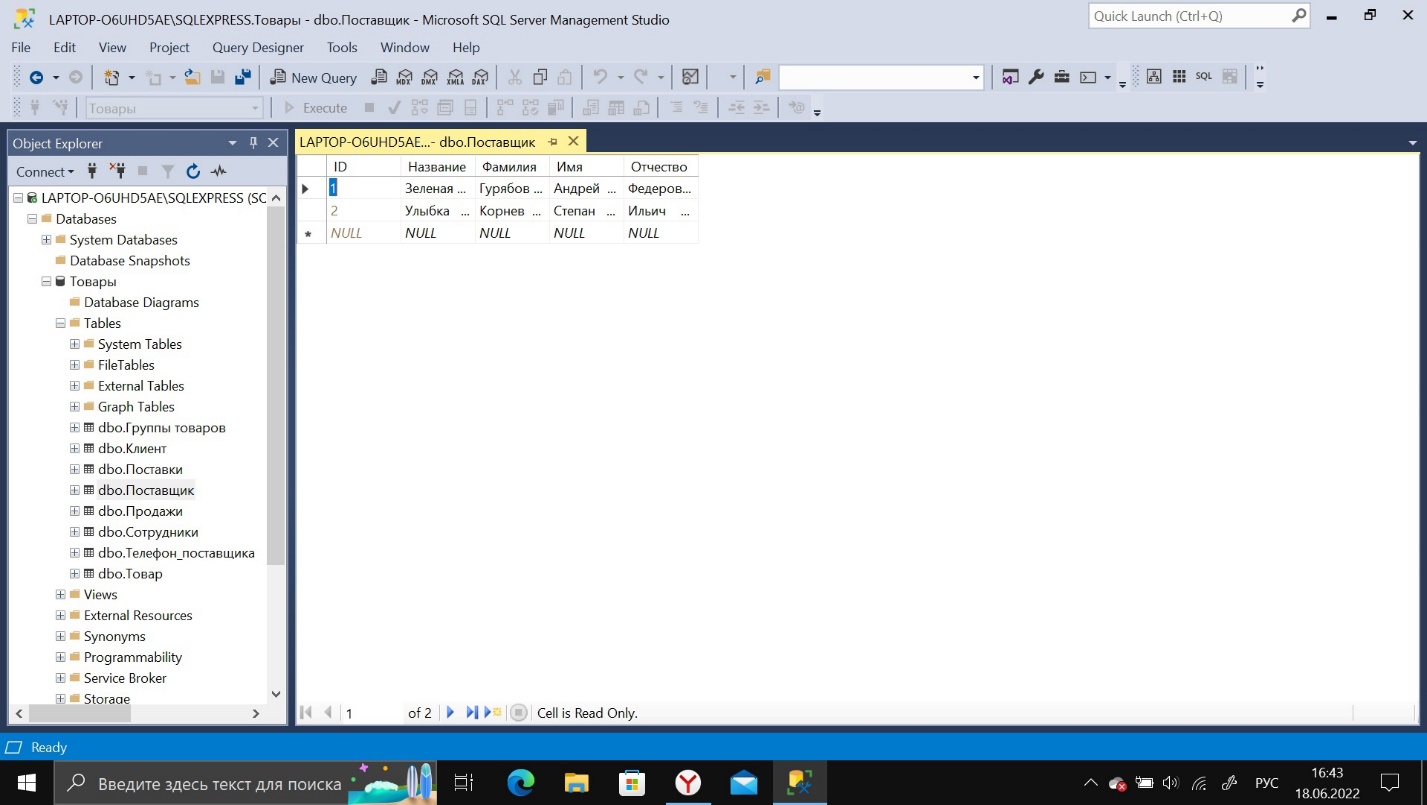
Создать базу данных «Товары». В ней создать таблицы, которые относятся к этой базе. Установить связи между таблицами этой базы, заполнить их некоторыми данными. В отчёт вставить схемы данных и данные из таблиц.











Изображение выглядит как текст

Автоматически созданное описание

