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.

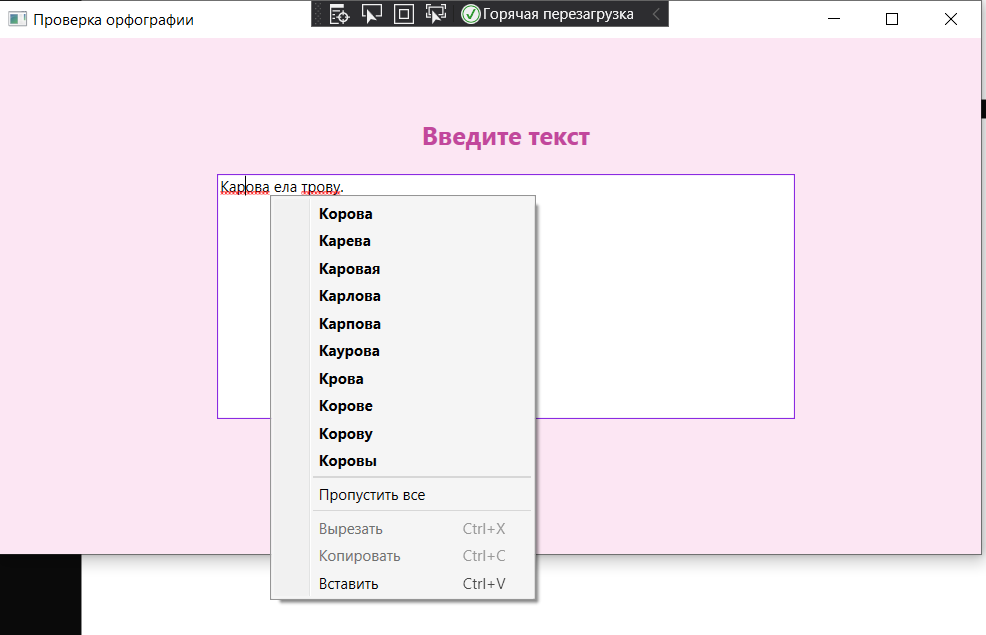
Группа: 3ПКС-420

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.

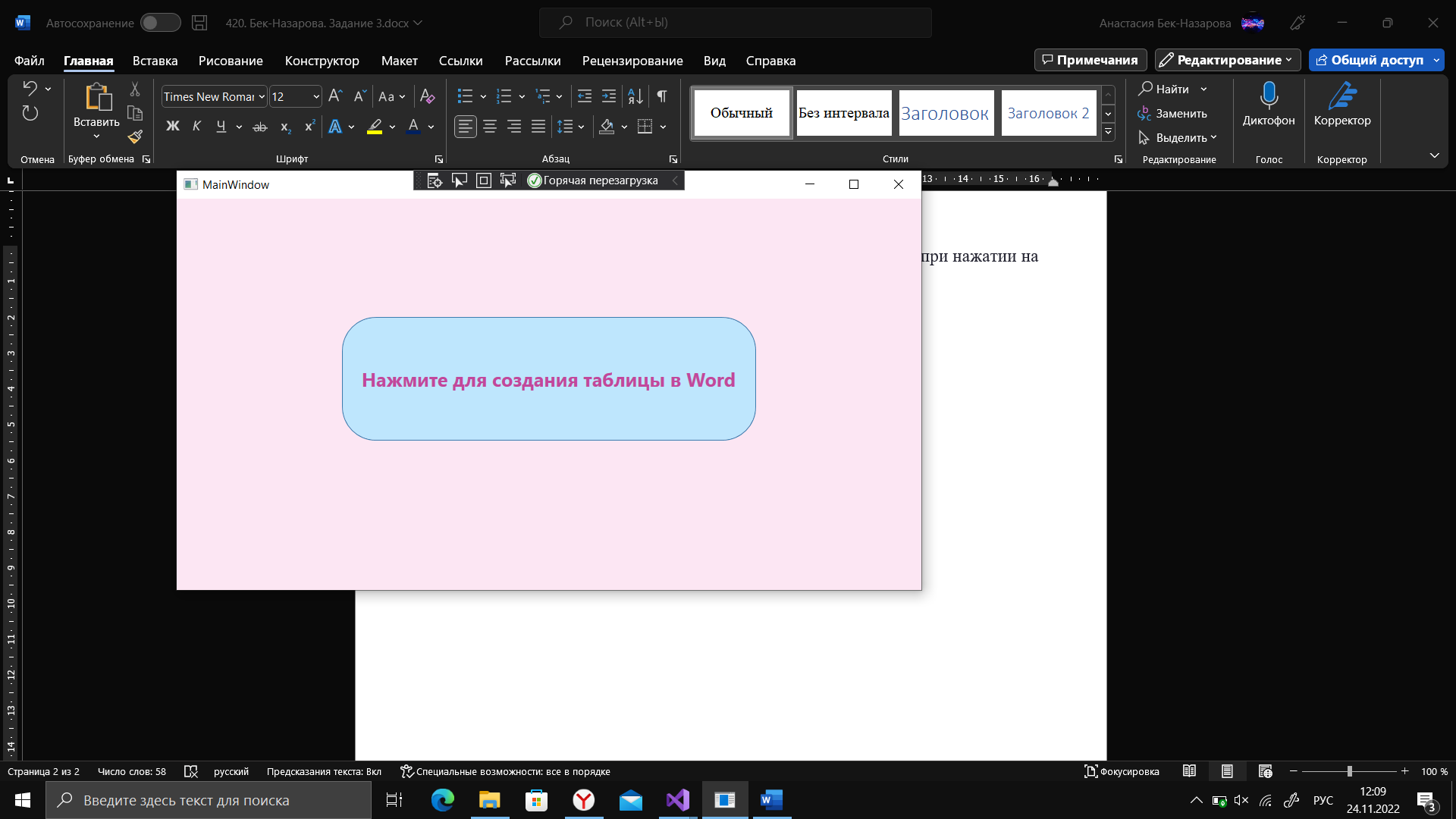
Группа: 3ПКС-420

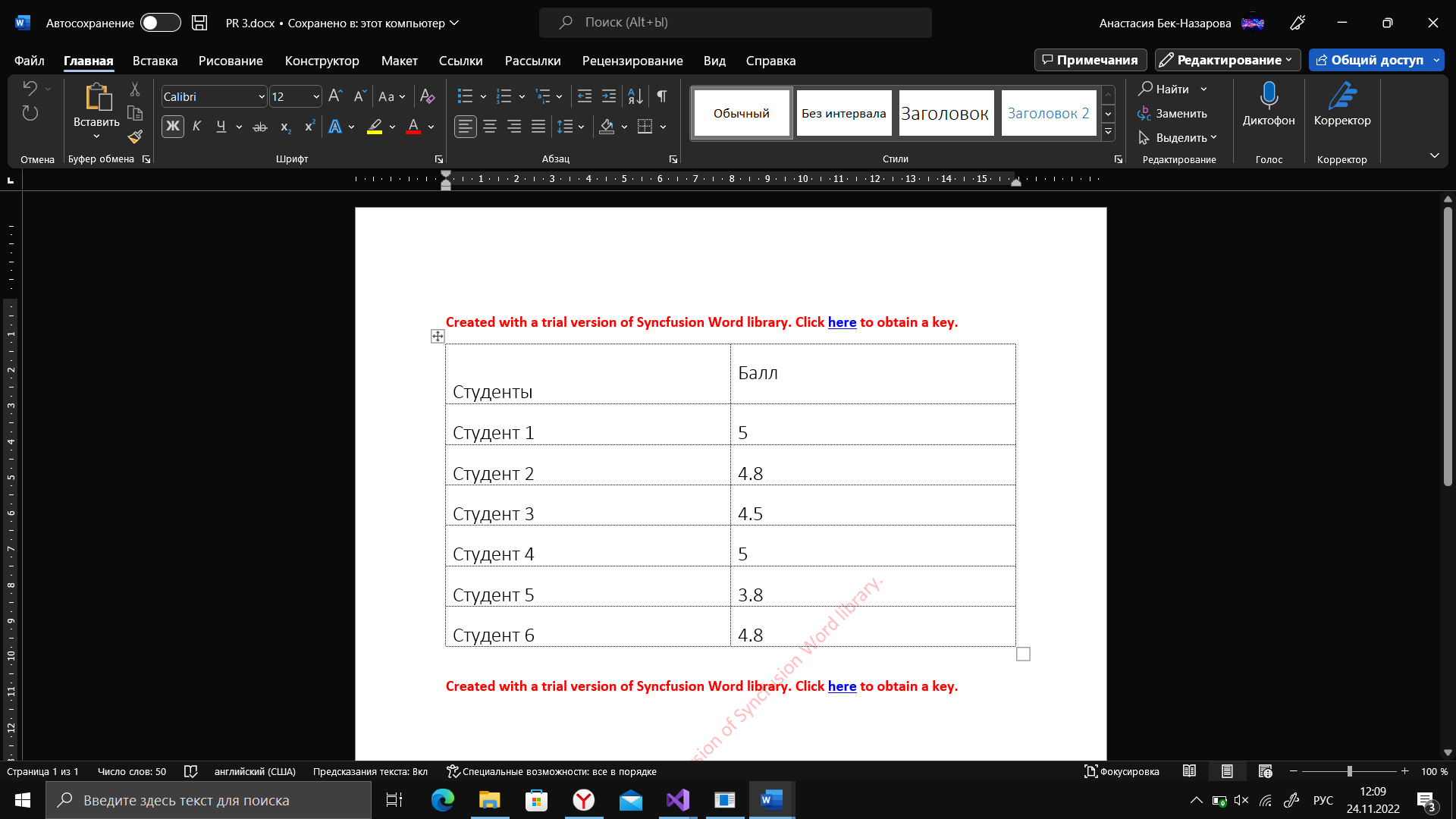
Преподаватель: Сибирев И. В.

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.

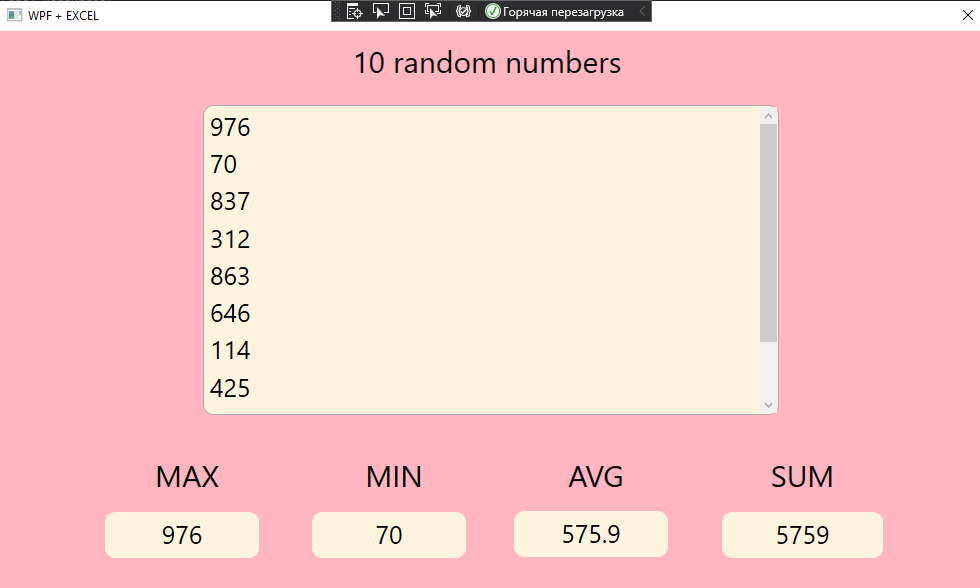
Группа: 3ПКС-420

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.