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

Я. Р. Гнеушев

*(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**

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Темы учебной практики** | **Выполненные задания/работы** |
| **1** | **Локальные данные процедур** | Дано 50 чисел, среди которых есть по крайней мере одно отрицательное. Найти наибольшее среди отрицательных чисел. |
| **2** | **Процедуры** | Ввести одномерный массив m = {-1, 0, 10, -3, -5, 6, -2, 3, 4}. Вычислить сумму элементов с нечетными номерами. Сформировать и вывести массив n, элементами которого являются индексы положительных элементов массива m. 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 беззнаковых чисел. Написать программу, позволяющую найти сумму всех строк массива. Исходную матрицу и полученные суммы вывести на экран. |
| **5** | **Обработка одномерных массивов** | Написать программу определения заданной характеристики последовательности чисел C1, C2, ..., Cn. Найти количество чисел, равных C1 |

# Задание 1

Дано 50 чисел, среди которых есть по крайней мере одно отрицательное. Найти наибольшее среди отрицательных чисел.

proc func

mov ECX,[N]

lea ESI, [X]

mov EBX,0

mov DL,0

SEARCH:

lodsd

or EAX,EAX

jns NEXT

or DL,DL

jnz COMPARE

mov EBX,EAX

mov DL,1

COMPARE:

cmp EAX,EBX

jle NEXT

mov BX,AX

NEXT: loop SEARCH

cinvoke printf, text1,ebx

invoke sleep, 9000

invoke exit, 0

mov sp,bp

pop bp

ret

endp

call func

invoke sleep, 9000

invoke exit, 0

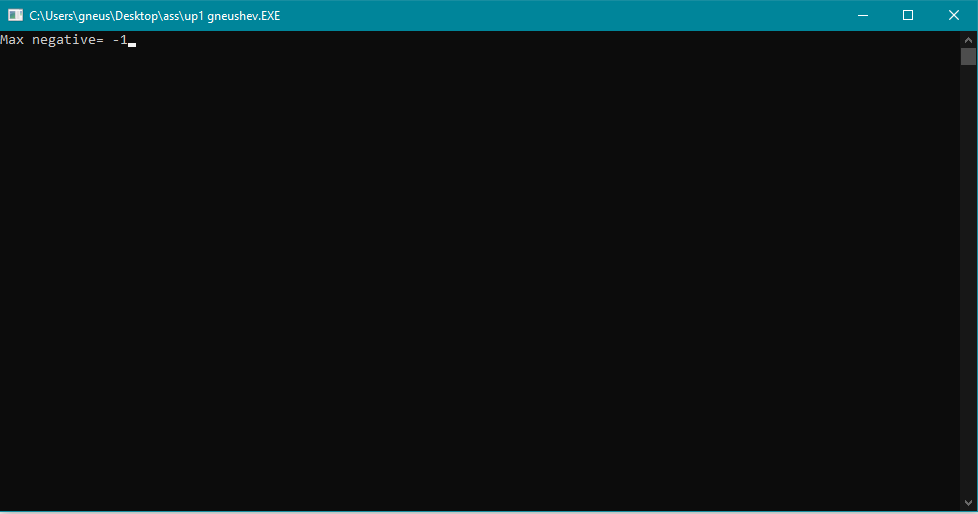
text1 db 'Max negative= %d',0

fpp db '%d',9, 0

N dd 50

X dd -3, 0, 2, 5, 10, -5, 0, 77, 69, -13, -14, 9, 1, 4, 1, 8, 9, 0, -3, 10, 19, -42, 5, 90,71

dd 9, 5, -9, 45, -90, -10, -1, 6, 14, 456, 122, 34, 87, 99, 12, 11, 321, 67, 81, 44, 92, 55, 2, -89, 100



# Задание 2

Ввести одномерный массив m = {-1, 0, 10, -3, -5, 6, -2, 3, 4}. Вычислить сумму элементов с нечетными номерами. Сформировать и вывести массив n, элементами которого являются индексы положительных элементов массива m. Программу разработать на основе пользовательских функций.

macro prnt op1, op2 {

local metka

mov ebx, op1

mov ecx,op2

mov eax, 0

metka :

mov eax,[ebx]

push ecx

cinvoke printf, fpp, eax

pop ecx

add ebx, 4

loop metka

}

prnt m,9

lea ebx, [m]

mov ecx,5

mov eax,0

cycl:

add eax, [ebx]

add ebx,8

loop cycl

cinvoke printf, text1, eax

lea ebx, [m]

mov ecx,9

mov eax,0

mov edx,0

mov esi,n

cycl2:

mov eax,[ebx]

cmp eax,0

jg met

add ebx,4

inc edx

loop cycl2

met:

inc [amount]

mov [esi], edx

add esi, 4

add ebx,4

inc edx

loop cycl2

prnt n,[amount]

Изображение выглядит как текст

Автоматически созданное описание

# Задание 3

proc RevPrint

mov eax, [n]

mov ecx, 10

idiv ecx

mov [n], eax

push eax

cinvoke printf,text2,edx

pop eax

cmp eax,0

ja met

ret

met:

call RevPrint

ret

endp

cinvoke printf,text1

cinvoke scanf, fmat, n

mov edx, 0

call RevPrint

Изображение выглядит как текст

Автоматически созданное описание

# Задание 4

Разместить в памяти компьютера матрицу 4\*3 целых беззнаковых чисел. Написать программу, позволяющую найти сумму всех строк массива. Исходную матрицу и полученные суммы вывести на экран.

macro prnt op1, op2,op3 {

local metka

mov esi,op1

mov ecx,op2

met2:

push ecx

mov ecx,op3

met1:

push ecx

mov eax,[esi]

add [sum],eax

add [sum2],eax

push ecx

cinvoke printf, fpp, eax

pop ecx

add esi, 4

pop ecx

loop met1

cinvoke printf,text3,[sum]

mov [sum],0

pop ecx

loop met2

}

cinvoke printf,text2

prnt m,4,3

cinvoke printf,text3,[sum2]

Изображение выглядит как текст

Автоматически созданное описание

# Задание 5

Написать программу определения заданной характеристики последовательности чисел C1, C2, ..., Cn. Найти количество чисел, равных C1

macro prnt op1, op2 {

local metka

mov ebx, op1

mov ecx,op2

mov eax, 0

metka :

mov eax,[ebx]

push ecx

cinvoke printf, fpp, eax

pop ecx

add ebx, 4

loop metka

}

prnt m,15

lea ebx, [m]

mov ecx,15

mov eax,[ebx]

proc function

cycl:

cmp eax,[ebx]

jz met

jmp ex

met:

inc [count]

ex:

add ebx,4

loop cycl

cinvoke printf,text1,[count]

endp

call function

Изображение выглядит как текст, экран, снимок экрана, закрыть

Автоматически созданное описание

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:**

**Обработка двумерных массивов**

Студент: Гнеушев Я. Р.

Группа: 2ПКС-420

Преподаватель: Киреева Г. И.

Дата: 19. июня. 2022 г

**Цель работы**: 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.

**Ниже на рисунке представлен фрагмент работающего проекта**

Изображение выглядит как текст

Автоматически созданное описание

**Код программы:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

void Matrix(bool answer)

{

Console.Clear();

int N=0;

do

{

Console.ForegroundColor = ConsoleColor.Red;

Console.Write("Введите размер матрицы(NxN) >0: ");

Console.ResetColor();

Console.ResetColor();

try

{

N = int.Parse(Console.ReadLine());

}

catch (FormatException)

{

Console.WriteLine("Неверный формат ввода. Введите число!");

}

} while (N <= 0);

int[,] arr = new int[N,N];

int sumMainDiag = 0; int minMainDiag = 99; int maxMainDiag = 10;

int sumPobochDiag = 0; int minPobochDiag = 99; int maxPobochDiag = 10;

int sumUpTriangle = 0; int minUpTriangle = 99; int maxUpTriangle = 10;

int sumDwTriangle = 0; int minDwTriangle = 99; int maxDwTriangle = 10;

int sumRtTriangle = 0; int minRtTriangle = 99; int maxRtTriangle = 10;

int sumLfTriangle = 0; int minLfTriangle = 99; int maxLfTriangle = 10;

Console.CursorTop = 10;

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

{

Console.WriteLine();

Console.CursorLeft = 15;

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

{

arr[i,j] = new Random().Next(10, 100);

if (i == j) // главная диагональ

{

Console.ForegroundColor = ConsoleColor.DarkRed;

Console.Write(arr[i,j] + " ");

sumMainDiag += arr[i,j];

if (arr[i,j]<minMainDiag) minMainDiag = arr[i,j];

if (arr[i, j] > maxMainDiag) maxMainDiag = arr[i, j];

}

else if (i + j == N-1) // побочная диагональ

{

Console.ForegroundColor = ConsoleColor.DarkBlue;

Console.Write(arr[i,j] + " ");

sumPobochDiag += arr[i,j];

if (arr[i, j] < minPobochDiag) minPobochDiag = arr[i, j];

if (arr[i, j] > maxPobochDiag) maxPobochDiag = arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.White;

Console.Write(arr[i,j] + " ");

sumUpTriangle += arr[i,j];

if (arr[i, j] < minUpTriangle) minUpTriangle = arr[i, j];

if (arr[i, j] > maxUpTriangle) maxUpTriangle = arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

Console.Write(arr[i,j] + " ");

sumRtTriangle += arr[i,j];

if (arr[i, j] < minRtTriangle) minRtTriangle = arr[i, j];

if (arr[i, j] > maxRtTriangle) maxRtTriangle = arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.Green;

Console.Write(arr[i,j] + " ");

sumDwTriangle += arr[i,j];

if (arr[i, j] < minDwTriangle) minDwTriangle = arr[i, j];

if (arr[i, j] > maxDwTriangle) maxDwTriangle = arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.Yellow;

Console.Write(arr[i,j] + " ");

Console.ResetColor();

sumLfTriangle+= arr[i,j];

if (arr[i, j] < minLfTriangle) minLfTriangle = arr[i, j];

if (arr[i, j] > maxLfTriangle) maxLfTriangle = arr[i, j];

}

}

}

Console.WriteLine("\n");

if (answer)

{

Console.CursorLeft = 20;

Console.ForegroundColor = ConsoleColor.DarkRed;

Console.WriteLine("Главная диагональ: \n\t\t\tСумма: " + sumMainDiag + "\n\t\t\tМинимиму: " + minMainDiag + "\n\t\t\tМаксимум: " + maxMainDiag);

Console.CursorLeft = 20;

Console.ForegroundColor = ConsoleColor.DarkBlue;

Console.WriteLine("Побочная диагональ: \n\t\t\tСумма: " + sumPobochDiag + "\n\t\t\tМинимиму: " + minPobochDiag + "\n\t\t\tМаксимум: " + maxPobochDiag);

Console.CursorLeft = 20;

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("Верхний треугольник: \n\t\t\tСумма: " + sumUpTriangle + "\n\t\t\tМинимиму: " + minUpTriangle + "\n\t\t\tМаксимум: " + maxUpTriangle);

Console.CursorLeft = 20;

Console.ForegroundColor = ConsoleColor.DarkGreen;

Console.WriteLine("Правый треугольник: \n\t\t\tСумма: " + sumRtTriangle + "\n\t\t\tМинимиму: " + minRtTriangle + "\n\t\t\tМаксимум: " + maxRtTriangle);

Console.CursorLeft = 20;

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Нижний треугольник: \n\t\t\tСумма: " + sumDwTriangle + "\n\t\t\tМинимиму: " + minDwTriangle + "\n\t\t\tМаксимум: " + maxDwTriangle);

Console.CursorLeft = 20;

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("Левый треугольник: \n\t\t\tСумма: " + sumLfTriangle + "\n\t\t\tМинимиму: " + minLfTriangle + "\n\t\t\tМаксимум: " + maxLfTriangle);

Console.ResetColor();

}

}

string answ="";

Console.WriteLine("\t\t Гнеушев\_420 вывод цветной матрицы");

while (answ != "1" || answ != "2") {

Console.ForegroundColor = ConsoleColor.Yellow;

Console.Write("Вы хотите вывести подсчёты по каждой зоне?\n");

Console.ResetColor();

Console.ForegroundColor = ConsoleColor.DarkGreen;

Console.WriteLine("1 - да");

Console.ForegroundColor = ConsoleColor.DarkRed;

Console.WriteLine("2 - нет");

Console.ResetColor();

Console.Write("Ваш ответ: "); answ = Console.ReadLine();

if (answ == "1" || answ == "2") break;

else

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Введён неверный ответ!");

Console.ResetColor();

}

}

if (answ == "1") { Matrix(true); }

else if (answ == "2") { Matrix(false); }

}

}

}

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:**

**Вычисление ветвящейся функции с использованием C# WPF**

Студент: Гнеушев Я. Р.

Группа: 2ПКС-420

Преподаватель: Киреева Г. И.

Дата: 19. июня. 2022 г

**Цель работы**: вычислить ветвящуюся функцию, представленную на рисунке

в проекте. Функция имеет особые точки, связанные с наличием log (x+a) в

знаменателе.

**Ниже на рисунке представлен фрагмент работающего проекта**

Изображение выглядит как стол

Автоматически созданное описание

**Код программы:**

<Window x:Class="WpfApp1.MainWindow"

WindowStartupLocation="CenterScreen"

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"

ResizeMode="CanMinimize"

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

Title="Гнеушев\_420" MaxHeight="450" MaxWidth="600" MinHeight="450" MinWidth="600">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="569\*"/>

<ColumnDefinition Width="20\*"/>

<ColumnDefinition Width="7\*"/>

<ColumnDefinition Width="4\*"/>

</Grid.ColumnDefinitions>

<Image Source="Resources/formula.png" Margin="144,19,113,301" ></Image>

<TextBlock Name="label" HorizontalAlignment="Left" Margin="184,1,0,0" TextWrapping="Wrap" Text="Вычислить функцию" VerticalAlignment="Top" Height="23" Width="182" FontSize="18"/>

<TextBox Name="input" HorizontalAlignment="Left" Height="23" Margin="76,110,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" FontSize="18"/>

<TextBlock HorizontalAlignment="Left" Margin="55,108,0,0" TextWrapping="Wrap" Text="a=" VerticalAlignment="Top" Height="23" Width="30" FontSize="18"/>

<Button IsDefault="true" Name="Calculate" Content="Вычислить" HorizontalAlignment="Left" Margin="300,106,0,0" VerticalAlignment="Top" Width="87" Height="50" FontSize="16" Click="Calculate\_Click"/>

<Button Name="Write" Content="Записать &#xD;&#xA; в файл" HorizontalAlignment="Left" Margin="413,106,0,0" VerticalAlignment="Top" Width="87" Height="50" FontSize="16" Click="Write\_Click" />

<Button Name="Read" Content="Загрузить &#xA; с файла" HorizontalAlignment="Left" Margin="413,175,0,0" VerticalAlignment="Top" Width="87" Height="50" FontSize="16" Click="Read\_Click" />

<Button Name="Exit" Content="Выход" HorizontalAlignment="Left" Margin="413,312,0,0" VerticalAlignment="Top" Width="87" Height="50" FontSize="16" Click="Exit\_Click" />

<DataGrid Name="Data" HorizontalAlignment="Left" Height="254" Margin="55,141,0,0" VerticalAlignment="Top" Width="220"/>

<Button Name="ClearGridData" Content="Очистить&#xA; таблицу" HorizontalAlignment="Left" Margin="413,243,0,0" VerticalAlignment="Top" Width="87" Height="50" FontSize="16" Click="ClearGridData\_Click" />

</Grid>

</Window>

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.IO;

namespace WpfApp1

{

public class list

{

public int X { get; set; }

public string Y { get; set; }

public list(int x, string y)

{

X = x;

Y = y;

}

}

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

input.Focus();

}

private List<list> result = new List<list>();

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

{

try

{

double.Parse(input.Text);

}

catch (FormatException)

{

MessageBox.Show("Введите число!");

return;

}

result.Clear();

Data.ItemsSource = null;

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

{

if (i >= 0)

{

if (i + double.Parse(input.Text) <= 0 || Math.Log10(i + double.Parse(input.Text)) <= 0)

{

result.Add(new list(i, "Не определён"));

continue;

}

else

{

double var = Math.Round(i / Math.Log10(i + double.Parse(input.Text)), 2);

result.Add(new list(i, var.ToString()));

}

}

else

{

double var = Math.Round(Math.Pow(i, 2) \* Math.Sin(i + double.Parse(input.Text)), 2);

result.Add(new list(i, var.ToString()));

}

}

input.Text = "";

Data.ColumnWidth = 97;

Data.ItemsSource = result;

}

private void Write\_Click(object sender, RoutedEventArgs e)//загрузить в файл

{

if (Data.ItemsSource == null) return;

StreamWriter sw = new StreamWriter("A:/c++/formula wpf/formula wpf/file.txt");

for (int i = 0; i <= 40; i++)

{

sw.Write(result[i].Y+"\n");

}

sw.Close();

MessageBox.Show("Запись успешно произведена!");

}

private void Read\_Click(object sender, RoutedEventArgs e) //загрузить с файла

{

StreamReader sr = new StreamReader("A:/c++/formula wpf/formula wpf/file.txt");

if (sr.ReadLine()=="")

{

MessageBox.Show("Файл пуст!");

return;

}

result.Clear();

Data.ItemsSource = null;

for (int i = -20; i < 21; i++)

{

result.Add(new list(i, sr.ReadLine()));

}

sr.Close();

Data.ColumnWidth = 97;

Data.ItemsSource = result;

}

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

{

this.Close();

}

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

{

Data.ItemsSource = null;

}

}

}

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:**

**Работа с элементом ListBox с использованием C# WPF**

Студент: Гнеушев Я. Р.

Группа: 2ПКС-420

Преподаватель: Киреева Г. И.

Дата: 19. июня. 2022 г

**Цель работы**: Работа с листбоксом

- добавить окно ввода страны, обработать исключения(пробелы и тд)

-кнопка добавить, она добавляет в листбокс введённое предложение

-кнопка изменить, выбрать страну в листбоксе, 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

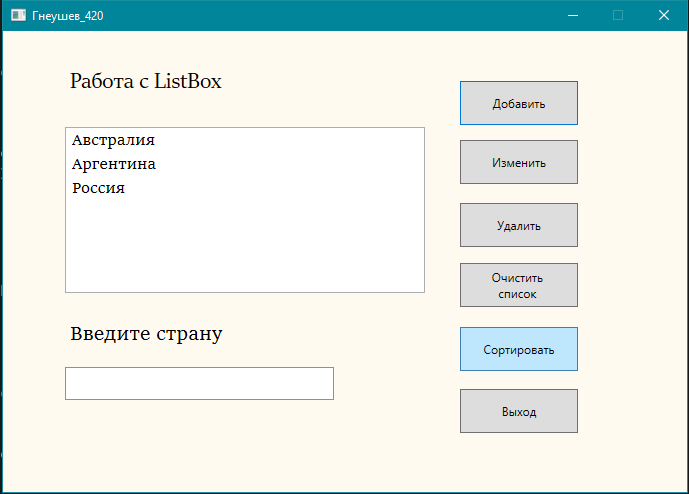
-очистить список, тоже самое, что и удаление, но предупреждение с восклицательным знаком

-сортировку она не показала, но думаю просто по алфавиту

-счетчик записей под окном ввода

-кнопка выхода

**Ниже на рисунке представлен фрагмент работающего проекта**



**Код программы:**

<Window x:Class="WpfApp5.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:WpfApp5"

mc:Ignorable="d"

WindowStartupLocation="CenterScreen"

ResizeMode="CanMinimize"

Title="Гнеушев\_420" MinHeight="500" MaxHeight="500" MinWidth="700" MaxWidth="700">

<Grid Background="FloralWhite">

<ListBox Name="listbox" HorizontalAlignment="Left" Height="166" Margin="62,96,0,0" VerticalAlignment="Top" Width="360" RenderTransformOrigin="0.497,0.507" FontSize="16" FontFamily="Sitka Text"/>

<Label Content="Работа с ListBox" HorizontalAlignment="Left" Margin="62,31,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Sitka Banner"/>

<TextBox Name="textbox" HorizontalAlignment="Left" Height="33" Margin="62,336,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="269" FontSize="18" />

<Label Content="Введите страну" HorizontalAlignment="Left" Margin="62,286,0,0" VerticalAlignment="Top" FontSize="20" Height="36" FontFamily="Sitka Text"/>

<Button Name="Add" IsDefault="True" Content="Добавить" HorizontalAlignment="Left" Margin="457,50,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Add\_Click"/>

<Button Name="Edit" Content="Изменить" HorizontalAlignment="Left" Margin="457,109,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Edit\_Click"/>

<Button Name="Delete" Content="Удалить" HorizontalAlignment="Left" Margin="457,172,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Delete\_Click"/>

<Button Name="Clear" Content="Очистить &#xD;&#xA; список" HorizontalAlignment="Left" Margin="457,232,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Clear\_Click" />

<Button Name="Sort" Content="Сортировать" HorizontalAlignment="Left" Margin="457,296,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Sort\_Click"/>

<Button Content="Выход" HorizontalAlignment="Left" Margin="457,358,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Button\_Click"/>

</Grid>

</Window>

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;

namespace WpfApp5

{

public partial class MainWindow : Window

{

char[] arr = new char[] {'!','@','#','$','%','^','&','\*','(',')','\_','+','=','\\','|','/','?'};

public MainWindow()

{

InitializeComponent();

Show();

textbox.Focus();

}

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

{

this.Close();

}

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

{

if (textbox.Text.Trim().Length == 0)

{

MessageBox.Show("Введите страну!");

textbox.Clear();

textbox.Focus();

return;

}

if (listbox.Items.Contains(textbox.Text))

{

MessageBox.Show("Данная страна уже присутствует в списке!");

textbox.Clear();

textbox.Focus();

return;

}

foreach(var i in textbox.Text)

{

if (char.IsDigit(i) || arr.Contains(i))

{

MessageBox.Show("Введите страну!");

textbox.Focus();

return;

}

}

listbox.Items.Add(textbox.Text.Trim());

textbox.Clear();

textbox.Focus();

}

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

{

if (listbox.Items.Count == 0)

{

MessageBox.Show("Список пуст!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Exclamation);

return;

}

if (MessageBox.Show("Вы точно хотите очистить список?","Подтвердите действие",MessageBoxButton.YesNo,MessageBoxImage.Question) == MessageBoxResult.Yes)

{

listbox.Items.Clear();

textbox.Focus();

}

}

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

{

if (listbox.Items.Count == 0)

{

MessageBox.Show("Список пуст!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Exclamation);

return;

}

listbox.Items.SortDescriptions.Add(

new System.ComponentModel.SortDescription("",

System.ComponentModel.ListSortDirection.Ascending));

textbox.Focus();

}

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

{

if (listbox.SelectedItems.Count == 0)

{

MessageBox.Show("Ничего не выбрано!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Exclamation);

return;

}

textbox.Text = (string)listbox.SelectedItem;

listbox.Items.Remove(listbox.SelectedItem);

textbox.Focus();

textbox.SelectionStart = textbox.Text.Length;

}

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

{

if (listbox.SelectedItems.Count == 0)

{

MessageBox.Show("Ничего не выбрано!", "Внимание", MessageBoxButton.OK, MessageBoxImage.Exclamation);

return;

}

listbox.Items.RemoveAt(listbox.SelectedIndex);

}

}

}

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:**

**Вычисление ветвящейся функции с использованием C# WPF**

Студент: Гнеушев Я. Р.

Группа: 2ПКС-420

Преподаватель: Киреева Г. И.

Дата: 19. июня. 2022 г

**Цель работы**: Создать базу данных «Торговое предприятие». 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 стол

Автоматически созданное описание

Изображение выглядит как стол

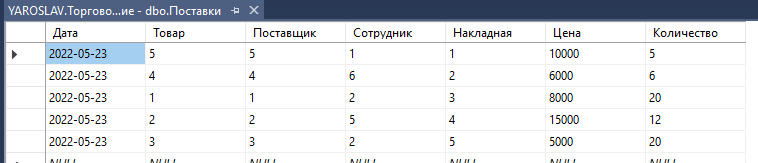
Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание



Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Федеральное государственное образовательное бюджетное   
учреждение высшего образования

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

**(колледж информатики и программирования)**

**ОТЧЕТ   
по практической (самостоятельной) работе**

**Практическая работа № 7**

**Студента:** Гнеушева Я. Р.

**Дисциплина /Профессиональный модуль:** УП-01.01

|  |  |  |
| --- | --- | --- |
| **Группа:** ПКС-420 |  | **Преподаватель:** |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Альшакова Е. Л.  И. В. Сибирев/ |
|  |  | **Дата выполнения:** |
|  |  | 26. ноября. 2022 г |
|  |  | **Оценка за работу:**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

Москва   
2022

Постановка задачи

Разработать калькулятор на языке С#. В проекте использовать прием

создания одного обработчика для нескольких событий.

Листинг программы

MainWindow.xaml

<Window x:Class="Calculator\_WPF.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:Calculator\_WPF"

mc:Ignorable="d"

Title="Калькулятор Гнеушев 420 ПР1" Height="600" Width="600" WindowStartupLocation="CenterScreen" ResizeMode="NoResize" Background="Gainsboro">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="10\*"/>

<ColumnDefinition Width="109\*"/>

</Grid.ColumnDefinitions>

<Button Content="7" HorizontalAlignment="Left" Margin="48,145,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.ColumnSpan="2" Background="#FFF39696" Cursor="Hand"/>

<Button Content="8" HorizontalAlignment="Left" Margin="98.5,145,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button Content="9" HorizontalAlignment="Left" Margin="198.5,145,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button Content="4" HorizontalAlignment="Left" Margin="48,249,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.ColumnSpan="2" Background="#FFF39696" Cursor="Hand"/>

<Button Content="5" HorizontalAlignment="Left" Margin="98.5,249,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button Content="6" HorizontalAlignment="Left" Margin="198.5,249,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button Content="2" HorizontalAlignment="Left" Margin="98.5,353,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button Content="1" HorizontalAlignment="Left" Margin="48,353,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.ColumnSpan="2" Background="#FFF39696" Cursor="Hand"/>

<Button Content="3" HorizontalAlignment="Left" Margin="198.5,353,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button Content="0" HorizontalAlignment="Left" Margin="48,451,0,0" VerticalAlignment="Top" Width="280" Height="80" FontSize="30" Click="Button0\_Click" Grid.ColumnSpan="2" Background="#FFF39696" Cursor="Hand"/>

<TextBox Name="screen" TextAlignment="Center" IsEnabled="False" HorizontalAlignment="Left" Height="56" Margin="48,45,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="500" FontSize="40" Grid.ColumnSpan="2" Background="#FFF9AEAE"/>

<Button HorizontalContentAlignment="Center" Content="+" HorizontalAlignment="Left" Margin="310.5,145,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="40" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button HorizontalContentAlignment="Center" Content="-" HorizontalAlignment="Left" Margin="310.5,249,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="40" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button HorizontalContentAlignment="Center" Content="\*" HorizontalAlignment="Left" Margin="310.5,353,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="40" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button HorizontalContentAlignment="Center" Content="/" HorizontalAlignment="Left" Margin="310.5,451,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="32" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button HorizontalContentAlignment="Center" Content="." HorizontalAlignment="Left" Margin="418.5,451,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="40" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button Content="=" HorizontalAlignment="Left" VerticalContentAlignment="Center" HorizontalContentAlignment="Center" Margin="418,353,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="60" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button HorizontalContentAlignment="Center" Content="CA" HorizontalAlignment="Left" Margin="418.5,145,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

<Button HorizontalContentAlignment="Center" Content="DEL" HorizontalAlignment="Left" Margin="418,249,0,0" VerticalAlignment="Top" Width="80" Height="80" FontSize="30" Click="Button0\_Click" Grid.Column="1" Background="#FFF39696" Cursor="Hand"/>

</Grid>

</Window>

MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

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 Calculator\_WPF

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button0\_Click(object sender, RoutedEventArgs e)

{

var txt = (e.Source as Button).Content.ToString();

if (new string[9] {"1", "2", "3", "4", "5", "6", "7", "8", "9" }.Contains(txt))

{

screen.Text += txt;

}

else if (txt == "0")

{

if (screen.Text.Length != 0)

{

screen.Text += "0";

}

}

else if(new string[4] { "+", "-", "\*", "/" }.Contains(txt) && !new char[5] { '+', '-', '\*', '/', '.' }.Contains(screen.Text[screen.Text.Length - 1]))

{

screen.Text += txt;

}

else if(txt == ".")

{

if (screen.Text.Length == 0 || !new char[5] { '+', '-', '\*', '/', '.' }.Contains(screen.Text[screen.Text.Length - 1]))

if (screen.Text.Count(x => x == '.') < screen.Text.Count(x => x == '+' || x == '-' || x == '\*' || x == '/') + 1)

screen.Text += ".";

}

else if(txt == "CA")

{

screen.Text = "";

}

else if(txt == "=" && screen.Text.Trim().Length > 0)

{

try

{

screen.Text = Eval(screen.Text).ToString().Replace(',', '.');

}

catch (OverflowException)

{

MessageBox.Show("Переполнение!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch (DivideByZeroException)

{

MessageBox.Show("Деление на ноль!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch (Exception)

{

MessageBox.Show("Ошибка в выражении!");

}

}

else if(txt == "DEL")

{

if (screen.Text.Length > 0)

screen.Text = screen.Text.Substring(0, screen.Text.Length - 1);

}

}

static Double Eval(String expression)

{

System.Data.DataTable table = new System.Data.DataTable();

return Convert.ToDouble(table.Compute(expression, String.Empty));

}

protected override void OnClosing(CancelEventArgs e)

{

e.Cancel = MessageBox.Show("Вы действительно хотите выйти?", "Подтвердите действие", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No;

}

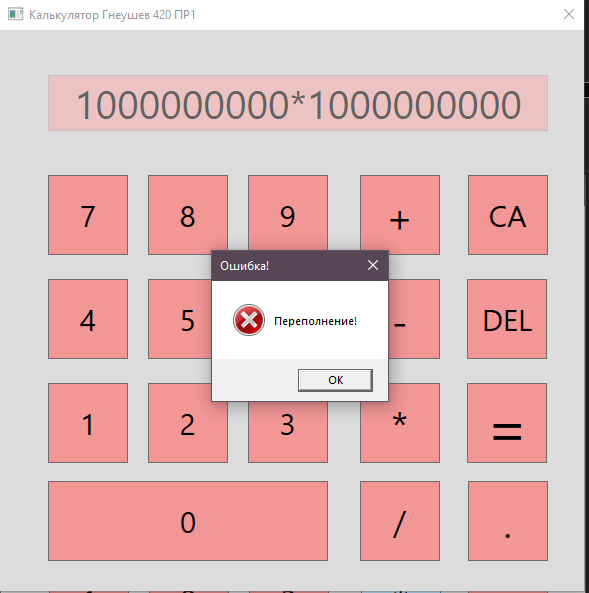
}

}

Изображение выглядит как текст, электроника

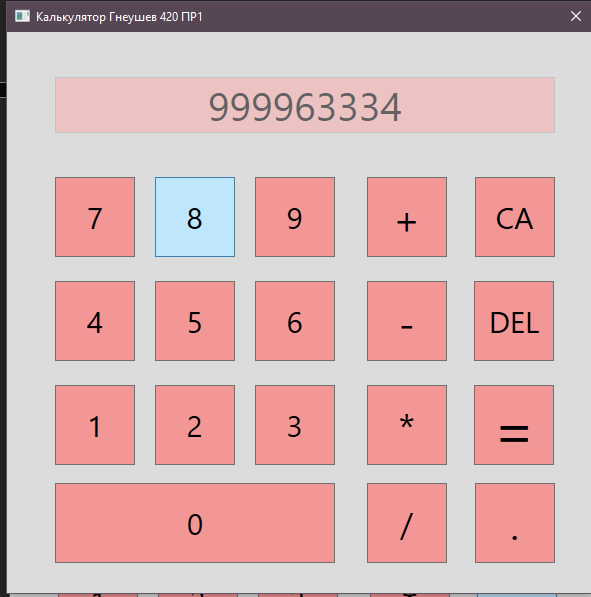
Автоматически созданное описание





Изображение выглядит как текст, электроника

Автоматически созданное описание



Федеральное государственное образовательное бюджетное   
учреждение высшего образования

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

**(колледж информатики и программирования)**

**ОТЧЕТ   
по практической (самостоятельной) работе**

**Практическая работа № 8**

**Студента:** Гнеушева Я. Р.

**Дисциплина /Профессиональный модуль:** УП-01.01

|  |  |  |
| --- | --- | --- |
| **Группа:** ПКС-420 |  | **Преподаватель:** |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Альшакова Е. Л.  И. В. Сибирев/ |
|  |  | **Дата выполнения:** |
|  |  | 26. ноября. 2022 г |
|  |  | **Оценка за работу:** |

Москва   
2022

Постановка задачи

Разработать программу, которая предлагает пользователю ввести какие-

либо слова, предложения в текстовое поле и после нажатия соответствующей

кнопки проверить орфографию введенного текста. Для непосредственной

проверки орфографии воспользуемся функцией Checkspelling объектной

библиотеки MS Word.

MainWindow.xaml

<Window x:Class="up\_check\_orf.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:up\_check\_orf"

mc:Ignorable="d"

Title="Гнеушев\_420 УП ПР2" Height="450" Width="800" ResizeMode="NoResize">

<Grid Background="Cornsilk">

<TextBox HorizontalAlignment="Left" Name ="text" Margin="100,143,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="612" Height="162" FontSize="20" Background="MintCream"/>

<TextBlock HorizontalAlignment="Center" Height="40" Margin="0,36,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="800" FontSize="25" TextAlignment="Center">

<Run Language="ru-ru" Text="ПРОВЕРКА ТЕКСТА НА ОРФОГРАФИЮ"/>

</TextBlock>

<TextBlock HorizontalAlignment="Center" Height="33" Margin="0,106,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="384" FontWeight="DemiBold" FontSize="24" TextAlignment="Center">

<Run Language="ru-ru" Text="Введите текст "/>

</TextBlock>

<Button Content="Проверить текст" HorizontalAlignment="Left" Height="36" Margin="557,330,0,0" VerticalAlignment="Top" Width="155" Click="Button\_Click" FontWeight="DemiBold" FontSize="16" Background="Bisque" BorderThickness="2" BorderBrush="DarkGray" IsDefault="True" Cursor="Hand"/>

</Grid>

</Window>

MainWindow.xaml.cs

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 Word = Microsoft.Office.Interop.Word;

using System.Reflection;

namespace up\_check\_orf

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

var Word = new Microsoft.Office.Interop.Word.Application();

var textIsCorrect = Word.CheckSpelling(text.Text);

if (textIsCorrect)

{

MessageBoxResult result = MessageBox.Show("В тексте нет ошибок!", "Всё правильно", MessageBoxButton.OK, MessageBoxImage.Asterisk);

}

else

{

MessageBoxResult result = MessageBox.Show("В тексте есть одна или несколько ошибок!", "Есть ошибки", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

protected override void OnClosing(System.ComponentModel.CancelEventArgs e)

{

e.Cancel = MessageBox.Show("Вы действительно хотите выйти?", "Подтвердите действие", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No;

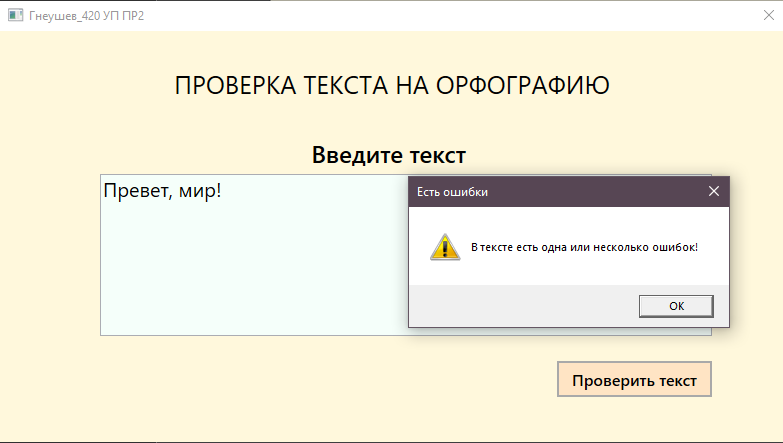
}

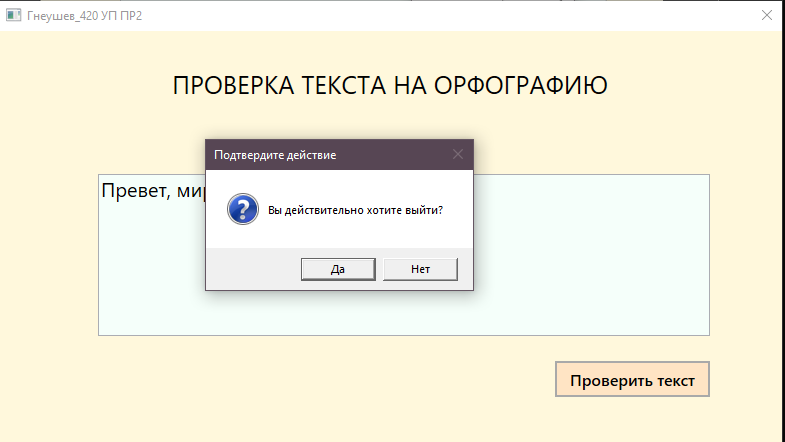
}

}

Изображение выглядит как текст

Автоматически созданное описание





Федеральное государственное образовательное бюджетное   
учреждение высшего образования

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

**(колледж информатики и программирования)**

**ОТЧЕТ   
по практической (самостоятельной) работе**

**Практическая работа № 9**

**Студента:** Гнеушева Я. Р.

**Дисциплина /Профессиональный модуль:** УП-01.01

|  |  |  |
| --- | --- | --- |
| **Группа:** ПКС-420 |  | **Преподаватель:** |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Альшакова Е. Л.  И. В. Сибирев/ |
|  |  | **Дата выполнения:** |
|  |  | 26. ноября. 2022 г |
|  |  | **Оценка за работу: \_\_\_\_\_\_\_\_\_\_\_\_** |

Москва   
2022

Постановка задачи

Необходимо разработать программу, которая при нажатии на кнопку создает

таблицу в MS Word.

MainWindow.xaml

<Window x:Class="WORD\_TABLE\_pr3.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:WORD\_TABLE\_pr3"

mc:Ignorable="d"

Title="Гнеушев\_420 УП пр3" Height="450" Width="800" ResizeMode="NoResize" Background="BlanchedAlmond" WindowStartupLocation="CenterScreen">

<Grid>

<Label Content="Создание таблиц в MS Word" HorizontalAlignment="Center" Margin="0,37,0,0" VerticalAlignment="Top" FontSize="40" FontWeight="Bold" FontFamily="Consolas" HorizontalContentAlignment="Center" Width="800"/>

<Button Content="Создать" HorizontalAlignment="Center" VerticalAlignment="Center" Height="70" Width="146" FontSize="30" FontWeight="Bold" FontFamily="Consolas" Background="Bisque" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" Click="Button\_Click"/>

</Grid>

</Window>

MainWindow.xaml.cs

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 Word = Microsoft.Office.Interop.Word;

namespace WORD\_TABLE\_pr3

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

var Word = new Microsoft.Office.Interop.Word.Application();

var doc = new Microsoft.Office.Interop.Word.Document();

object objMissing = System.Reflection.Missing.Value;

object endOfDoc = "\\endofdoc";

Word.Visible = true;

doc = Word.Documents.Add(ref objMissing, ref objMissing, ref objMissing, ref objMissing);

int rows = new Random().Next(1, 11);

int cols = new Random().Next(1, 11);

Microsoft.Office.Interop.Word.Table table;

Microsoft.Office.Interop.Word.Range rangeOfWord = doc.Bookmarks.get\_Item(ref endOfDoc).Range;

table = doc.Tables.Add(rangeOfWord, rows, cols, objMissing, objMissing);

table.Range.ParagraphFormat.SpaceAfter = 8;

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

{

for (int j = 1; j <= cols; j++)

{

table.Cell(i, j).Range.Text = $"Строка {i}, Столбец {j}";

}

}

try

{

table.Borders.Shadow = true;

}

catch (Exception) { }

Word.Quit();

}

protected override void OnClosing(System.ComponentModel.CancelEventArgs e)

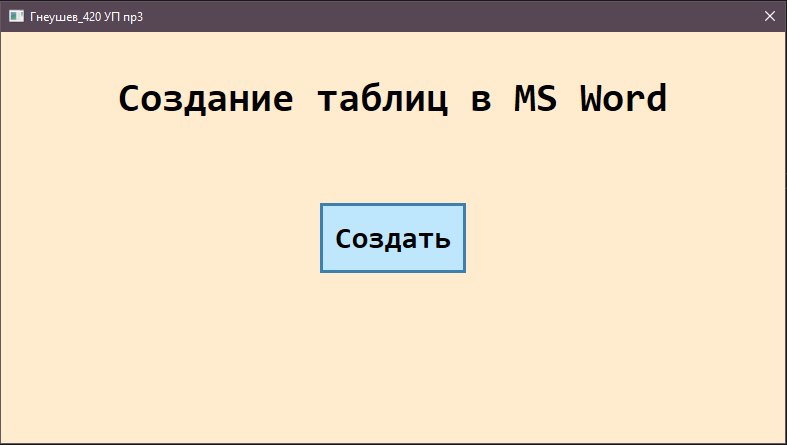
{

e.Cancel = MessageBox.Show("Вы действительно хотите выйти?", "Подтвердите действие", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No;

}

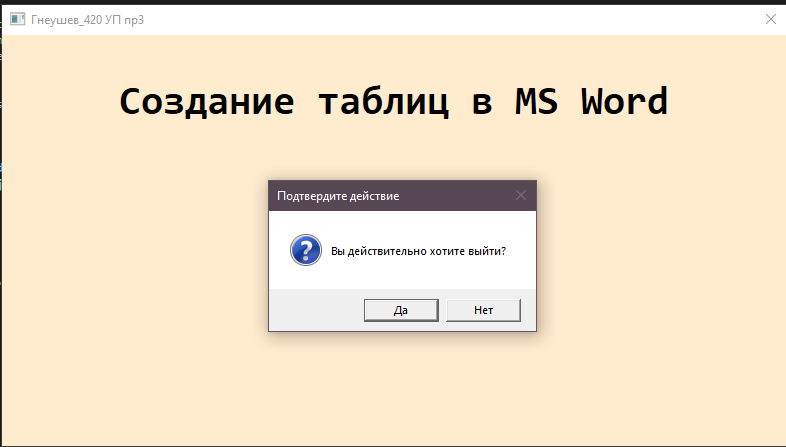
}

}



Изображение выглядит как стол

Автоматически созданное описание



Федеральное государственное образовательное бюджетное   
учреждение высшего образования

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

**(колледж информатики и программирования)**

**ОТЧЕТ   
по практической (самостоятельной) работе**

**Практическая работа № 10**

**Студента:** Гнеушева Я. Р.

**Дисциплина /Профессиональный модуль:** УП-01.01

|  |  |  |
| --- | --- | --- |
| **Группа:** ПКС-420 |  | **Преподаватель:** |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Альшакова Е. Л.  И. В. Сибирев/ |
|  |  | **Дата выполнения:** |
|  |  | 26. ноября. 2022 г |
|  |  | **Оценка за работу:**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

Москва   
2022

Постановка задачи

Cоздайте собственные проекты с использованием функций MS

Excel не используя функцию нахождения числа Pi.

MainWindow.xaml

<Window x:Class="WPF\_\_\_EXCEL.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:WPF\_\_\_EXCEL"

mc:Ignorable="d"

Title="Гнеушев\_420 WPF + EXCEL ПР4" Height="600" Width="1000" ResizeMode="NoResize" Background="LightSteelBlue">

<Grid>

<ListBox Name="listbox" Margin="50,74,278,118" Background="#FFFDF4DF" FontSize="25" FontFamily="Consolas" Foreground="Black"/>

<Label Content="SUM" Margin="799,20,37,497" FontSize="36" HorizontalContentAlignment="Center" Width="164" Height="64" FontFamily="Consolas"/>

<Label Name ="sum" Content="" Margin="799,86,24,404" FontSize="25" HorizontalContentAlignment="Center" Background="#FFFDF4DF" FontFamily="Consolas" Foreground="Black"/>

<Label Name="avg" Content="" HorizontalAlignment="Left" Margin="485,494,0,0" VerticalAlignment="Top" FontSize="25" HorizontalContentAlignment="Center" Width="164" Height="80" Background="#FFFDF4DF" FontFamily="Consolas" Foreground="Black"/>

<Label Content="AVG" Margin="485,437,351,64" FontSize="36" HorizontalContentAlignment="Center" Width="164" FontFamily="Consolas"/>

<Label Content="100 random numbers from 1 to 1000" Margin="36,9,264,511" FontSize="36" HorizontalContentAlignment="Center" Height="64" FontFamily="Consolas"/>

<Label Name="mode" Content="" HorizontalAlignment="Left" Margin="799,421,0,0" VerticalAlignment="Top" FontSize="25" HorizontalContentAlignment="Center" Width="164" Height="82" Background="#FFFDF4DF" FontFamily="Consolas" Foreground="Black"/>

<Label Content="MODE" Margin="799,357,37,163" FontSize="36" HorizontalContentAlignment="Center" Width="164" Height="64" FontFamily="Consolas"/>

<Label Name="min" Content="" HorizontalAlignment="Left" Margin="71,494,0,0" VerticalAlignment="Top" FontSize="25" HorizontalContentAlignment="Center" Width="164" Height="80" Background="#FFFDF4DF" FontFamily="Consolas" Foreground="Black"/>

<Label Content="MIN" Margin="71,437,765,49" FontSize="36" HorizontalContentAlignment="Center" Width="164" FontFamily="Consolas" RenderTransformOrigin="0.495,0.119"/>

<Label Name="max" Content="" HorizontalAlignment="Left" Margin="279,494,0,0" VerticalAlignment="Top" FontSize="25" HorizontalContentAlignment="Center" Width="164" Height="80" Background="#FFFDF4DF" FontFamily="Consolas" Foreground="Black"/>

<Label Content="MAX" Margin="279,437,557,49" FontSize="36" HorizontalContentAlignment="Center" Width="164" FontFamily="Consolas"/>

<Label Name="disp" Content="" HorizontalAlignment="Left" Margin="799,0,0,0" VerticalAlignment="Center" FontSize="25" HorizontalContentAlignment="Center" Width="164" Height="80" Background="#FFFDF4DF" FontFamily="Consolas" Foreground="Black"/>

<Label Content="DISPERSION" Margin="780,180,6,306" FontSize="33" HorizontalContentAlignment="Center" FontFamily="Consolas"/>

</Grid>

</Window>

MainWindow.xaml.cs

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.ComponentModel;

using System.Data;

using System.Drawing;

using Excel = Microsoft.Office.Interop.Excel;

namespace WPF\_\_\_EXCEL

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

var XL = new Microsoft.Office.Interop.Excel.Application();

List<int> nums = new List<int>();

for(int i = 0; i < 100; i++)

{

int num = (int)XL.WorksheetFunction.RandBetween(1, 1\_000);

nums.Add(num);

listbox.Items.Add(num);

}

sum.Content = XL.WorksheetFunction.Sum(nums.ToArray());

avg.Content = XL.WorksheetFunction.Average(nums.ToArray());

mode.Content = XL.WorksheetFunction.Mode(nums.ToArray());

min.Content = XL.WorksheetFunction.Min(nums.ToArray());

max.Content = XL.WorksheetFunction.Max(nums.ToArray());

disp.Content = Math.Round(XL.WorksheetFunction.Var(nums.ToArray()),2);

XL.Quit();

}

protected override void OnClosing(System.ComponentModel.CancelEventArgs e)

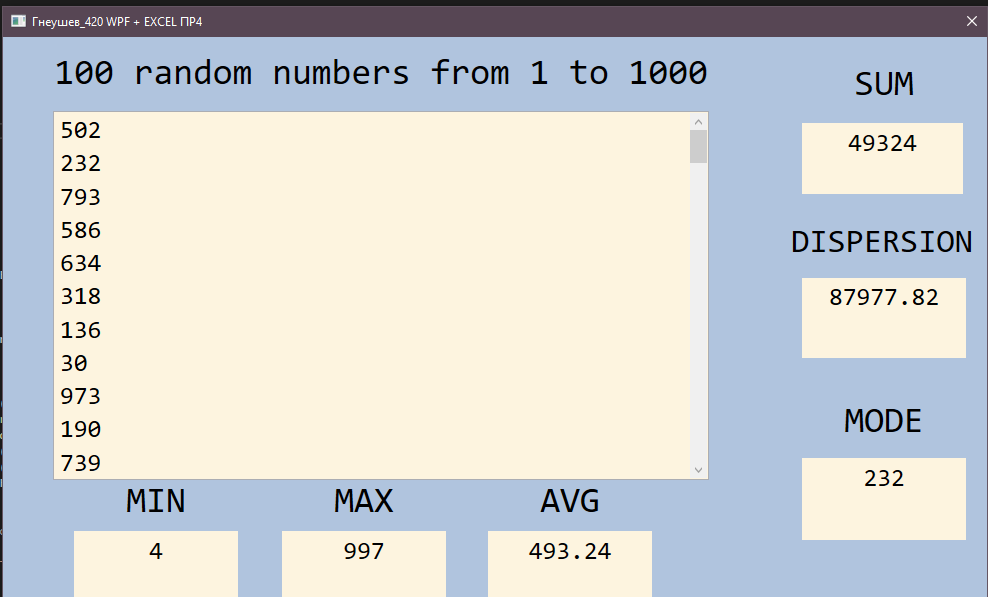
{

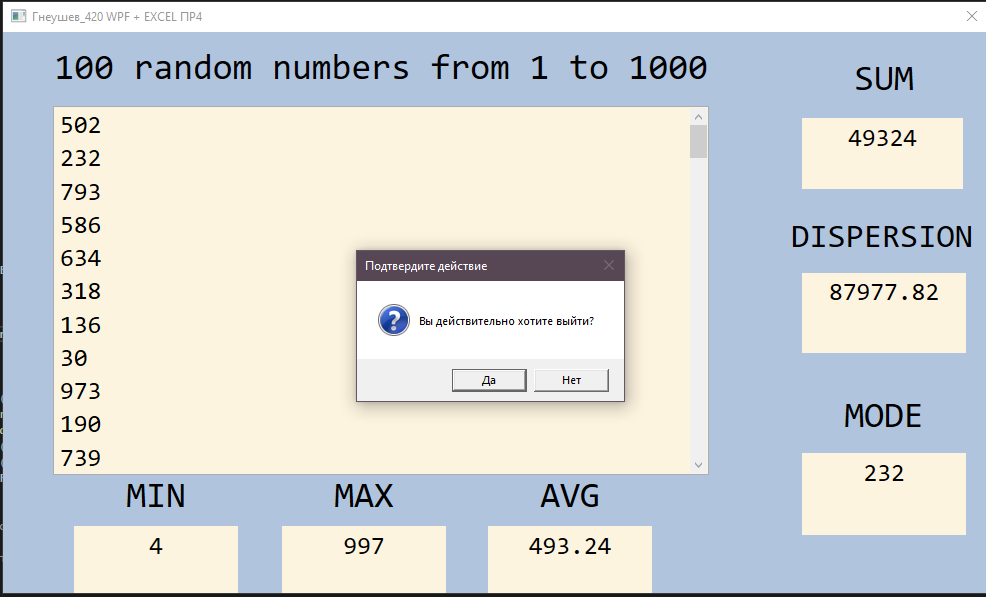
e.Cancel = MessageBox.Show("Вы действительно хотите выйти?", "Подтвердите действие", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No;

}

}

}





Федеральное государственное образовательное бюджетное   
учреждение высшего образования

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

**(колледж информатики и программирования)**

**ОТЧЕТ   
по практической (самостоятельной) работе**

**Практическая работа № 11**

**Студента:** Гнеушева Я. Р.

**Дисциплина /Профессиональный модуль:** УП-01.01

|  |  |  |
| --- | --- | --- |
| **Группа:** ПКС-420 |  | **Преподаватель:** |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Альшакова Е. Л.  И. В. Сибирев/ |
|  |  | **Дата выполнения:** |
|  |  | 26. ноября. 2022 г |
|  |  | **Оценка за работу:**  **\_** |

Москва   
2022

Постановка задачи

Цель работы: написать программу на языке C#, которая реализует

работу с файлами в различных кодировках.

Программа должна обладать следующим функционалом:

Чтение информации из файла в кодировке Unicode

Чтение информации из файла в кодировке Win1251

Чтение информации из файла с разрешением RTF

Чтение бинарных файлов

Сохранение информации в файл с кодировкой Unicode

Сохранение информации в файл с кодировкай Win1251

Сохранение информации в файл с разрешением RTF

Сохранение информации в бинарный файл

Печать текстового документа.

MainWindow.xaml

<Window x:Class="up\_encoding.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:up\_encoding"

mc:Ignorable="d"

Title="Гнеушев\_420 уп ПР5" Height="600" Width="800" ResizeMode="NoResize">

<Grid Background="Beige">

<TextBlock HorizontalAlignment="Center" Height="45" Margin="0,36,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="800" FontSize="25" FontWeight="Bold" TextAlignment="Center" Text="РАБОТА С ФАЙЛАМИ В РАЗЛИЧНЫХ КОДИРОВКАХ " FontFamily="Consolas"/>

<TextBlock HorizontalAlignment="Center" Name ="what\_to\_do" Height="26" Margin="0,98,0,0" TextWrapping="Wrap" VerticalAlignment="Top" FontSize="20" Width="800" TextAlignment="Center" FontFamily="Consolas">

<Run Language="ru-ru" Text="Введите текст" FontWeight="Bold"/>

</TextBlock>

<Button Content="Win1251" HorizontalAlignment="Left" Height="45" Margin="403,313,0,0" VerticalAlignment="Top" Width="113" Click="Button\_Click\_win1251" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<Button Content="RTF" HorizontalAlignment="Left" Height="45" Margin="157,313,0,0" VerticalAlignment="Top" Width="112" Click="Button\_Click\_RTF" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<Button Content="bin" HorizontalAlignment="Left" Height="45" Margin="282,313,0,0" VerticalAlignment="Top" Width="112" Click="Button\_Click\_bin" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness= "3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<Button Content="Unicode" HorizontalAlignment="Left" Height="45" Margin="526,313,0,0" VerticalAlignment="Top" Width="112" Click="Button\_Click\_Uni" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<RichTextBox Name ="text\_to\_save" HorizontalAlignment="Center" Height="148" Margin="0,129,0,0" VerticalAlignment="Top" Width="512" FontSize="18" TextBlock.FontFamily="Consolas" FontWeight="DemiBold" Background="BlanchedAlmond" BorderThickness="1.5" BorderBrush="Crimson">

<FlowDocument>

<Paragraph>

<Run Text="" FontSize="16"/>

</Paragraph>

</FlowDocument>

</RichTextBox>

<Label Content="Запись" HorizontalAlignment="Center" Margin="0,272,0,0" VerticalAlignment="Top" FontFamily="Consolas" FontSize="30"/>

<Label Content="Чтение" HorizontalAlignment="Center" Margin="0,358,0,0" VerticalAlignment="Top" FontFamily="Consolas" FontSize="30"/>

<Button Content="Win1251" HorizontalAlignment="Left" Height="45" Margin="403,410,0,0" VerticalAlignment="Top" Width="113" Click="Button\_Click\_win1251\_Read" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<Button Content="RTF" HorizontalAlignment="Left" Height="45" Margin="157,410,0,0" VerticalAlignment="Top" Width="112" Click="Button\_Click\_RTF\_Read" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<Button Content="bin" HorizontalAlignment="Left" Height="45" Margin="282,410,0,0" VerticalAlignment="Top" Width="112" Click="Button\_Click\_bin\_Read" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness= "3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<Button Content="Unicode" HorizontalAlignment="Left" Height="45" Margin="526,410,0,0" VerticalAlignment="Top" Width="112" Click="Button\_Click\_Uni\_Read" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

<Button Content="Напечатать" HorizontalAlignment="Left" Height="45" Margin="332,470,0,0" VerticalAlignment="Top" Width="122" Click="Button\_Click\_Print" FontWeight="DemiBold" FontSize="20" Background="Bisque" BorderThickness= "3" BorderBrush="Crimson" Cursor="Hand" TextBlock.FontFamily="Consolas"/>

</Grid>

</Window>

MainWindow.xaml.cs

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.IO;

using System.Drawing.Printing;

using System.Drawing;

namespace up\_encoding

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

const string BASE\_DIR = @"C:\Users\gneus\Desktop";

public MainWindow()

{

InitializeComponent();

}

string StringFromRichTextBox(RichTextBox rtb)

{

return new TextRange(rtb.Document.ContentStart, rtb.Document.ContentEnd).Text;

}

private void Button\_Click\_win1251(object sender, RoutedEventArgs e)

{

string file = BASE\_DIR+@"\test\_win1251.txt";

string msg = StringFromRichTextBox(text\_to\_save);

string[] str = { msg };

File.WriteAllLines(file, str, Encoding.GetEncoding(1251));

MessageBox.Show("Успешно!");

}

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

{

string msg = StringFromRichTextBox(text\_to\_save);

string fileName = BASE\_DIR + @"\test\_bin.bin";

using (BinaryWriter writer = new BinaryWriter(File.Open(fileName, FileMode.OpenOrCreate)))

{

writer.Write(msg);

}

MessageBox.Show("Успешно!");

}

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

{

UnicodeEncoding unicode = new UnicodeEncoding();

String unicodeString = StringFromRichTextBox(text\_to\_save);

Byte[] encodedBytes = unicode.GetBytes(unicodeString);

String decodedString = unicode.GetString(encodedBytes);

using (StreamWriter writer = new StreamWriter(BASE\_DIR + @"\test\_uni.txt", false, Encoding.UTF8))

{

writer.WriteLine(decodedString);

}

MessageBox.Show("Успешно!");

}

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

{

TextRange text = new TextRange(text\_to\_save.Document.ContentStart, text\_to\_save.Document.ContentEnd);

string fileName = BASE\_DIR + @"\test\_rtf.rtf";

using (FileStream fs = File.Open(fileName, FileMode.OpenOrCreate))

{

text.Save(fs, DataFormats.Rtf);

}

MessageBox.Show("Успешно!");

}

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

{

// suppose that we have a test.txt at E:\

string filePath = BASE\_DIR + @"\test\_rtf.rtf";

if (!File.Exists(filePath))

{

MessageBox.Show("Файл не найден!\nУбедитесь, что сохранили его");

return;

}

string rtf = File.ReadAllText(filePath);

using (MemoryStream stream = new MemoryStream(Encoding.Default.GetBytes(rtf)))

{

text\_to\_save.Selection.Load(stream, DataFormats.Rtf);

}

MessageBox.Show("Успешно!");

}

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

{

string filePath = BASE\_DIR + @"\test\_bin.bin";

if (!File.Exists(filePath))

{

MessageBox.Show("Файл не найден!\nУбедитесь, что сохранили его");

return;

}

var text = File.ReadAllText(filePath);

text\_to\_save.Selection.Text = text;

MessageBox.Show("Успешно!");

}

private void Button\_Click\_win1251\_Read(object sender, RoutedEventArgs e)

{

string filePath = BASE\_DIR + @"\test\_win1251.txt";

if (!File.Exists(filePath))

{

MessageBox.Show("Файл не найден!\nУбедитесь, что сохранили его");

return;

}

var text = File.ReadAllText(filePath, Encoding.GetEncoding("windows-1251"));

text\_to\_save.Selection.Text = text;

MessageBox.Show("Успешно!");

}

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

{

string filePath = BASE\_DIR + @"\test\_uni.txt";

if (!File.Exists(filePath))

{

MessageBox.Show("Файл не найден!\nУбедитесь, что сохранили его");

return;

}

var text = File.ReadAllText(filePath);

text\_to\_save.Selection.Text = text;

MessageBox.Show("Успешно!");

}

protected override void OnClosing(System.ComponentModel.CancelEventArgs e)

{

e.Cancel = MessageBox.Show("Вы действительно хотите выйти?", "Подтвердите действие", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No;

}

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

{

string s = text\_to\_save.Selection.Text;

PrintDocument p = new PrintDocument();

p.PrintPage += delegate (object sender1, PrintPageEventArgs e1)

{

e1.Graphics.DrawString(s, new Font("Times New Roman", 12), new SolidBrush(System.Drawing.Color.Black), new RectangleF(0, 0, p.DefaultPageSettings.PrintableArea.Width, p.DefaultPageSettings.PrintableArea.Height));

};

try

{

p.Print();

}

catch (Exception ex)

{

throw new Exception("Проверьте подключение к принтеру", ex);

}

}

}

}

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

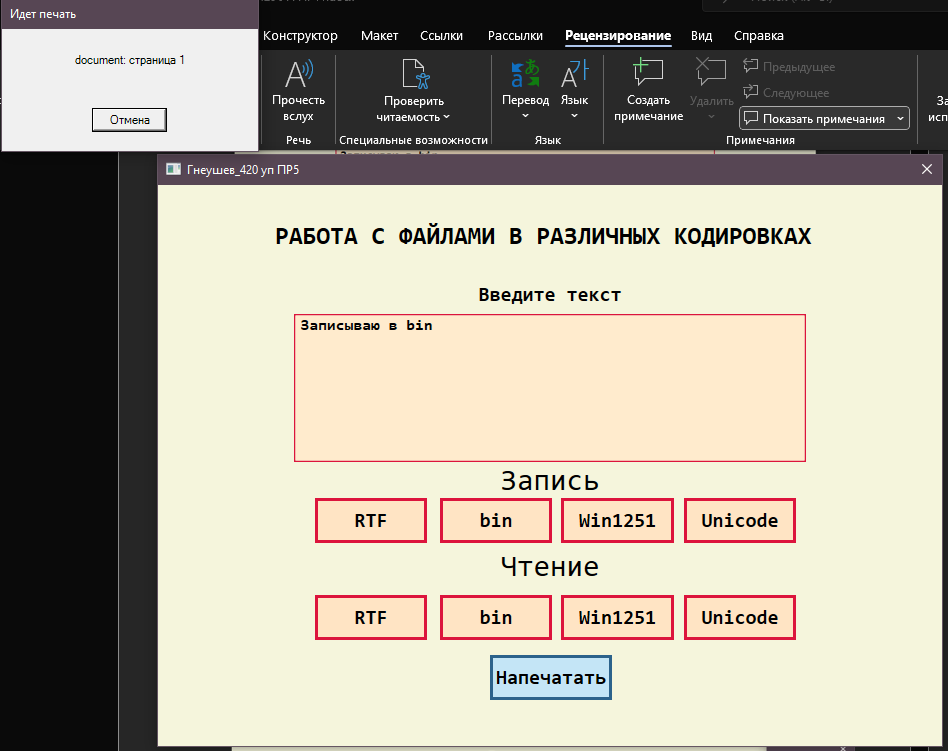
Нажал чтение Unicode и чтение bin

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание



Изображение выглядит как стол

Автоматически созданное описание

