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)*

Выполнила:

обучающийся учебной группы № 420

А.Ю. Топчиева

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

Е.Л. Альшакова

*(И.О. Фамилия)*

И.В. Сибирев

*(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**

[Локальные данные процедур 8](#_Toc121287418)

[Работа с процедурами 11](#_Toc121287419)

[Обработка двумерных массивов 14](#_Toc121287420)

[Написание консольного приложения для вывода числового массива 17](#_Toc121287421)

[Создание приложения с использованием компонента DataGridView и многострочным textBox 21](#_Toc121287422)

[Создание приложения с использованием компонента ListBox 27](#_Toc121287423)

[Создание приложения «Калькулятор» 32](#_Toc121287424)

[Создание приложения для проверки орфографии 36](#_Toc121287425)

[Создание приложения для организации взаимодействия с MS Word 39](#_Toc121287426)

[Создание приложения с использованием функций MS Excel 43](#_Toc121287427)

[Создание поточного приложения 47](#_Toc121287428)

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:**

Локальные данные процедур

Студент: А.Ю. Топчиева

Группа: 2ПКС-420

Преподаватель: Е.Л. Альшакова

Дата: 19.06.2022

**Цель работы:** разместить в памяти компьютера массив из N целых беззнаковых чисел. Написать программу, позволяющую найти минимальный элемент массива и его номер.

**Ниже на рисунке представлено консольное окно работающей программы**



Рисунок 1. Результат работы программы

**Код программы Accembler:**

format PE console

include 'win32a.inc'

entry start

section '.data' data readable writeable

text1 db 'Enter N: ',0

text3 db 10,13,'index = %d',0

text2 db 10,13,'min element= %d',0

text4 db 10,13,'',0

spp db '%d',0

fpp db '%d',9,0

p db 'pause',0

X dd 13,20,65,6,7,54,3,9,5,68,8,32,37,43,12

N dd ?

index dd ? ;индекс первого элемента начинается с нуля

section '.code' code readable executable

proc min num1, n ;процедура вывода минимального эл-та массива и его индекс

mov ebx, [num1]

mov ecx, [n]

mov eax,[num1]

MIN1:cmp [ebx],eax ;сравниваем

jg MIN2 ;если первый элемент меньше, то переходим на метку

mov eax,[ebx]

mov edx, [n]

sub edx, ecx

mov [index], edx

MIN2:add ebx,4

loop MIN1

ret

endp

start:;макросс для вывода

macro prntn op1,op2

{

local met

mov ebx,op1

mov ecx,op2

mov eax,0

met:

mov eax,[ebx]

push ecx

cinvoke printf,fpp,eax

pop ecx

add ebx, 4

loop met

}

cinvoke printf, text1

cinvoke scanf, spp, N

prntn X,[N];вывод масива

stdcall min, X,[N] ;вызов процеуры

cinvoke printf, text2,eax

cinvoke printf, text3,[index]

cinvoke printf, text4

cinvoke system,p

invoke exit,0

; import data in the same section

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data

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:**

Работа с процедурами

Студент: А.Ю. Топчиева

Группа: 2ПКС-420

Преподаватель: Е.Л. Альшакова

Дата: 19.06.2022

**Цель работы:** ввести одномерные массивы a(5) и b(5), состоящие из

произвольных чисел и определить в каком из массивов больше

положительных элементов. Получить и вывести новый массив с, состоящий

из положительных элементов массивов a и b. Программу разработать на

основе пользовательских процедур (функций).

**Ниже на рисунке представлено консольное окно работающей программы**

Рисунок 2. Результат работы программы

**Код программы Accembler:**

format PE console

include 'win32a.inc'

entry start

section '.data' data readable writeable

text1 db 'Enter n: ',0

text2 db 'Enter n1: ',0

text3 db 10,13,'Massiv a: %d',0

text5 db 10,13,'Massiv b: %d',0

text4 db 10,13,'',10,13,0

text6 db 10,13,'',10,13,0

p db 'pause',0

fpp db ' %d',0

fmat db '%d',9,0

a dd 13, 34, -65, -66, -32

b dd 43,-15,-65, -76,-7

c dd 100 dup(?)

n dd ?

n1 dd ?

k dd ?

section '.code' code readable executable

proc kolich

xor eax, eax

cycl:

cmp [esi], dword 0

jle met

inc eax

mov edx, [esi]

mov [ebx],edx

add ebx,4

met: add esi,4

loop cycl

ret

endp

start:

macro prntn op1,op2

{

local met

mov ebx,op1

mov ecx,op2

mov eax,0

met:

mov eax,[ebx]

push ecx

cinvoke printf,fpp,eax

pop ecx

add ebx, 4

loop met

}

cinvoke printf, text1

cinvoke scanf, fpp, n

cinvoke printf, text2

cinvoke scanf, fpp, n1

prntn a,[n]

cinvoke printf, text6

prntn b,[n1]

cinvoke printf, text6

lea esi,[a]

mov ecx,[n]

lea ebx,[c]

call kolich

mov [k],eax

lea esi,[b]

mov ecx,[n1]

call kolich

push eax

cmp [k], eax

jle m

cinvoke printf, text3, [k]

jmp m2

m:

cinvoke printf, text5, eax

m2:

pop eax

add [k], eax

cinvoke printf, text6

prntn c,[k]

cinvoke printf, text4

cinvoke system,p

invoke sleep, 9000

invoke exit,0

section '.relocs' fixups readable writeable

section '.idata' import data readable writeable

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

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:**

Обработка двумерных массивов

Студент: А.Ю. Топчиева

Группа: 2ПКС-420

Преподаватель: Е.Л. Альшакова

Дата: 19.06.2022

**Цель работы:** разместить в памяти компьютера матрицу 4\*3 целых беззнаковых чисел. Написать программу, позволяющую увеличить в 2 раза элементы матрицы с нечетными индексами. Исходную матрицу и измененную матрицу вывести на экран.

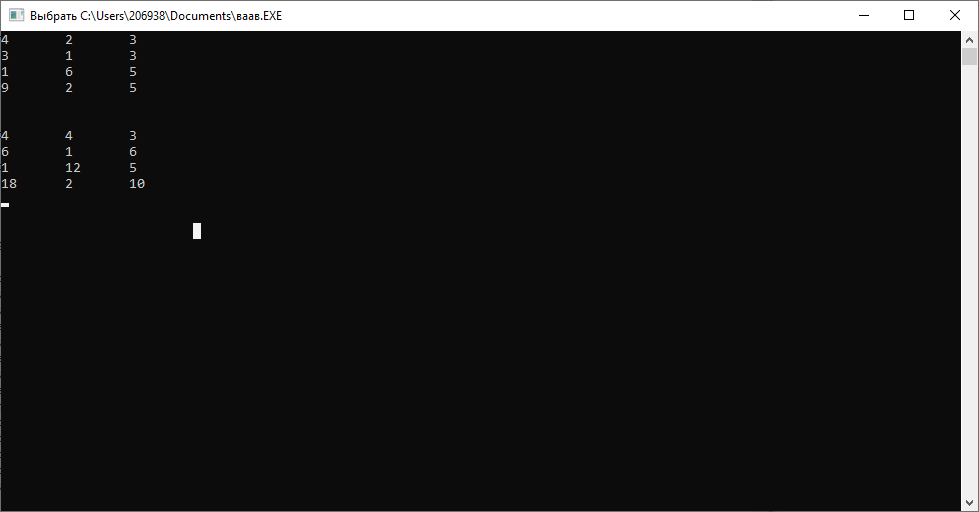
**Ниже на рисунке представлено консольное окно работающей программы**

Рисунок 3. Результат работы программы

**Код программы Accembler:**

format PE console

entry start

include 'win32a.inc'

section '.code' code readable executable

proc mas

met2: push ecx

mov ecx,[N]

met1: mov eax,[esi]

shl eax,1 ;сдвиг влево

mov [esi], eax

add esi,8

loop met1

pop ecx

loop met2

ret

endp

start:

macro prnt2d op1,op2,op3

{

local met1,met2

mov esi,op1

mov ecx,op2

met2:push ecx

mov ecx,op3

met1:push ecx

mov eax, [esi]

push ecx

cinvoke printf, fmat, eax

pop ecx

add esi,4

pop ecx

loop met1

cinvoke printf,text3

pop ecx

loop met2

}

prnt2d X,[N],[M]

lea esi,[X+4]

mov ecx,[M]

call mas

cinvoke printf, text4

prnt2d X,[N],[M]

invoke sleep,60000

invoke exit,0

section '.data' data readable writeable

text db 10, 13, '', 10, 13, 0

text3 db '',10,13,0

text4 db 10,13, '',10,13,0

fmat db '%d',9,0

spp db 10,13,'%d',0

X dd 4,2,3

dd 3,1,3

dd 1,6,5

dd 9,2,5

Y dd 100 dup(?)

N dd 4

M dd 3

; import data in the same section

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data

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:**

# Написание консольного приложения для вывода числового массива

Студент: А.Ю. Топчиева

Группа: 2ПКС-420

Преподаватель: И.В. Киреева

Дата: 20.06.2022

**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 рисунках представлены фрагменты работающего проекта**

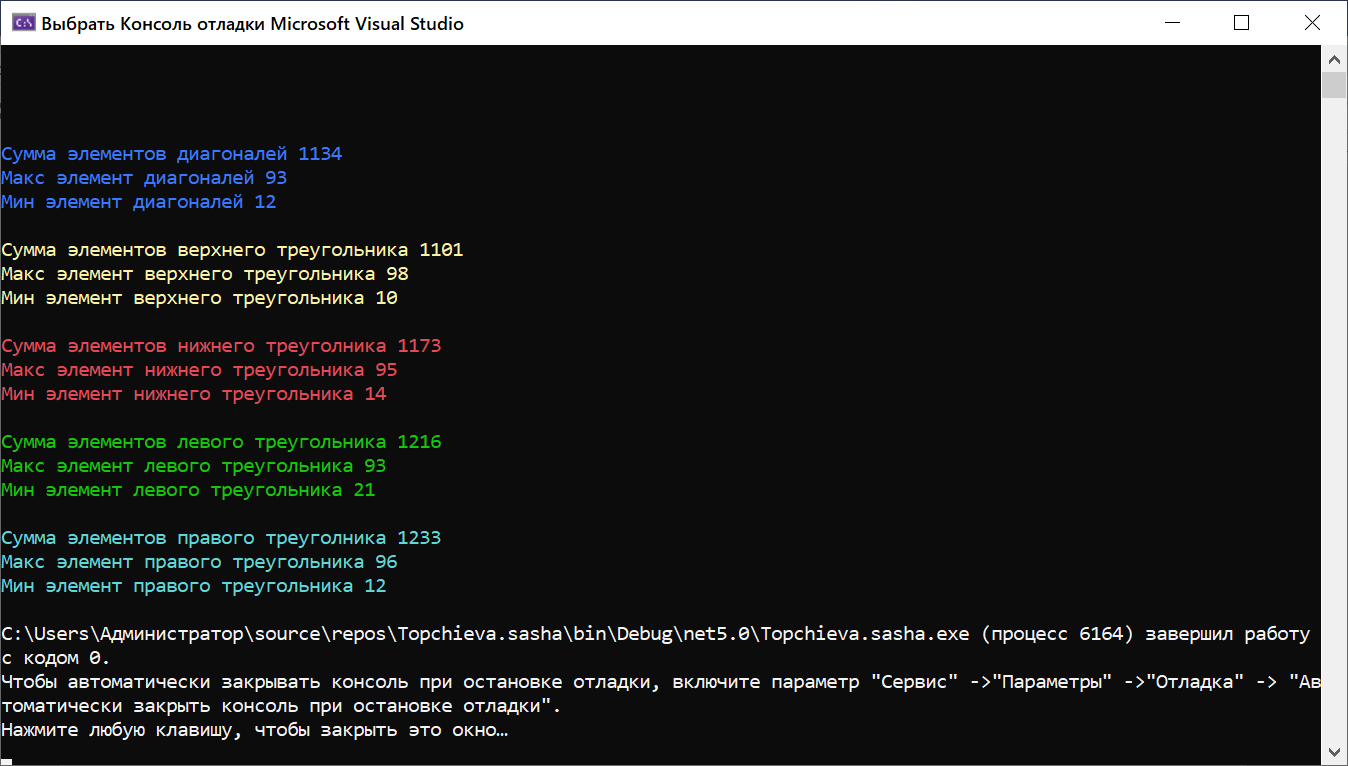
Рисунок 4. Вывод массива

Рисунок 5. Вычисления

**Код программы С#:**

class Program

{

static void Main(string[] args)

{

Console.Clear();

int[,] mass = new int[10, 10];

Random rand = new Random();

Console.ForegroundColor = ConsoleColor.White;

int sum1 = 0, sum2 = 0, sum3 = 0, sum4 = 0, sum5 = 0;

int max1 = 0, max2 = 0, max3 = 0, max4 = 0, max5 = 0;

int min1 = 100, min2 = 100, min3 = 100, min4 = 100, min5 = 100;

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

Console.CursorTop = 5

for (int y = 0; y < 10; y++)

{

Console.CursorLeft = 50;

for (int x = 0; x < 10; x++)

{

mass[y, x] = rand.Next(10, 100);

if (y == x || y + x == 9) //диагонали

{

if (mass[y, x] > max1)

{

max1 = mass[y, x];

}

if (mass[y, x] < min1)

{

min1 = mass[y, x];

}

sum1 += mass[y, x];

Console.ForegroundColor = ConsoleColor.Blue;

}

else if (y + x < 9 && y < x) //верхний треугольник

{

if (mass[y, x] > max2)

{

max2 = mass[y, x];

}

if (mass[y, x] < min2)

{

min2 = mass[y, x];

}

sum2 += mass[y, x];

Console.ForegroundColor = ConsoleColor.Yellow;

}

else if (y + x > 9 && x < y) //левый треугольник

{

if (mass[y, x] > max4)

{

max4 = mass[y, x];

}

if (mass[y, x] < min4)

{

min4 = mass[y, x];

}

sum4 += mass[y, x];

Console.ForegroundColor = ConsoleColor.Red;

}

else if (y + x < 9 && x < y) //нижний треугольник

{

if (mass[y, x] > max3)

{

max3 = mass[y, x];

}

if (mass[y, x] < min3)

{

min3 = mass[y, x];

}

sum3 += mass[y, x];

Console.ForegroundColor = ConsoleColor.Green;

}

else if (y + x > 9 && y < x) //правый треугольник

{

if (mass[y, x] > max5)

{

max5 = mass[y, x];

}

if (mass[y, x] < min5)

{

min5 = mass[y, x];

}

sum5 += mass[y, x];

Console.ForegroundColor = ConsoleColor.Cyan;

}

else

{

Console.ForegroundColor = ConsoleColor.White;

}

Console.Write(mass[y, x] + " ");

}

Console.WriteLine();

}

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

Console.ForegroundColor = ConsoleColor.Blue; //Диагонали

Console.WriteLine("\nСумма элементов диагоналей " + sum1);

Console.WriteLine("Макс элемент диагоналей " + max1);

Console.WriteLine("Мин элемент диагоналей " + min1);

Console.ForegroundColor = ConsoleColor.Yellow; //Верхний треуг

Console.WriteLine("\nСумма элементов верхнего треугольника " + sum2);

Console.WriteLine("Макс элемент верхнего треугольника " + max2);

Console.WriteLine("Мин элемент верхнего треугольника " + min2);

Console.ForegroundColor = ConsoleColor.Red; // Нижний треуг

Console.WriteLine("\nСумма элементов нижнего треуголника " + sum4);

Console.WriteLine("Макс элемент нижнего треугольника " + max4);

Console.WriteLine("Мин элемент нижнего треугольника " + min4);

Console.ForegroundColor = ConsoleColor.Green; // Левый треуг

Console.WriteLine("\nСумма элементов левого треугольника " + sum3);

Console.WriteLine("Макс элемент левого треугольника " + max3);

Console.WriteLine("Мин элемент левого треугольника " + min3);

Console.ForegroundColor = ConsoleColor.Cyan; //Правый треуг

Console.WriteLine("\nСумма элементов правого треуголника " + sum5);

Console.WriteLine("Макс элемент правого треугольника " + max5);

Console.WriteLine("Мин элемент правого треугольника " + min5);

Console.ForegroundColor = ConsoleColor.White;

}

}

}

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:**

# Создание приложения с использованием компонента DataGridView и многострочным textBox

Студент: А.Ю. Топчиева

Группа: 2ПКС-420

Преподаватель: И.В. Киреева

Дата: 20.06.2022

**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**

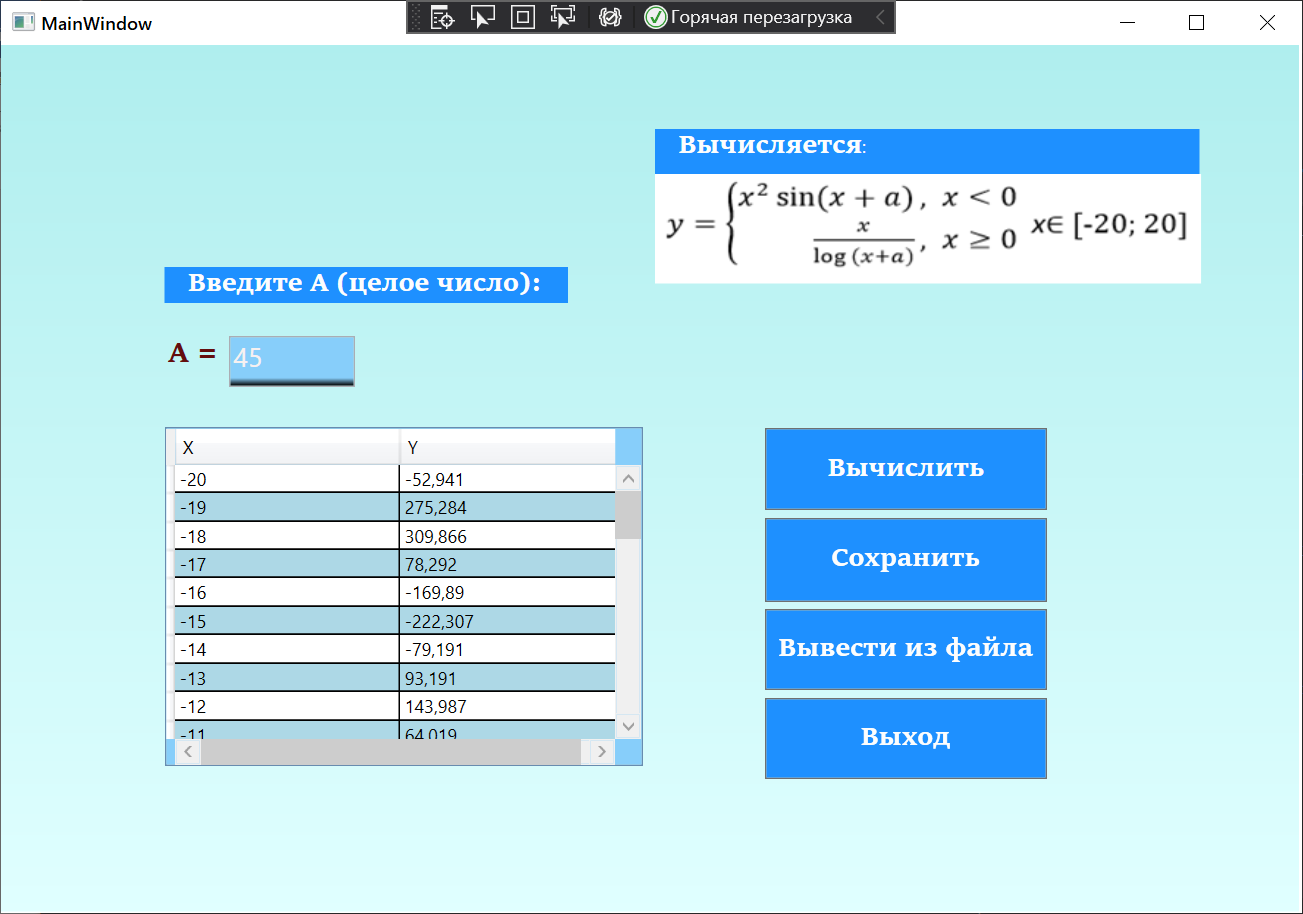


Рисунок 6. Вычисление ветвящийся функции

**Код программы C#:**

using System;

using System.IO;

using System.Windows;

namespace WpfApp31

{

public class Lst

{

public int X { get; set; }

public string Y { get; set; }

public Lst(int x, string y)

{

X = x;

Y = y;

}

}

public partial class MainWindow : Window

{

List<Lst> result = new List<Lst>();

List<Lst> res2 = new List<Lst>();

public MainWindow()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

textbox.TabIndex = 0;

textbox.Focus();

}

private void but2\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

private void but1\_Click(object sender, RoutedEventArgs e)

{

int a, x;

double y;

try

{

a = Int32.Parse(textbox.Text);

for (x = -20; x <= 20; x++)

{

if (textbox.Text == "")

{

MessageBox.Show("Введите значение а!");

}

if (x + a == 1)

{

y = x / Math.Log(x + a);

result.Add(new Lst(x, "Не вычисляется"));

}

if (x < 0)

{

y = Math.Pow(x, 2) \* Math.Sin(x + a);

result.Add(new Lst(x, Math.Round(y, 3).ToString()));

}

else

{

y = x / Math.Log(x + a);

result.Add(new Lst(x, Math.Round(y, 3).ToString()));

}

}

grid.ColumnWidth = 150;

grid.ItemsSource = result;

}

catch (FormatException)

{

MessageBox.Show("Неправильный формат!");

textbox.Focus();

return;

}

catch (OverflowException)

{

MessageBox.Show("Переполнение!");

textbox.Focus();

return;

}

}

string path = "C:\\Users\\Администратор\\Desktop\\Топчиева 2.txt";

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

{

using (StreamWriter sw = new StreamWriter(path, false))

{

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

{

sw.WriteLineAsync(result[i].X + " " + result[i].Y);

}

MessageBox.Show("Сохранено в файл!");

sw.Close();

}

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

{

try

{

using (StreamReader sr = new StreamReader(path))

{

string line = sr.ReadLine();

while (line != null)

{

Console.WriteLine(line);

string A = line.Split(' ').First();

string B = line.Split(' ').Last();

res2.Add(new Lst(x: int.Parse(A), y: B));

}

grid.ItemsSource = res2;

}

}

catch { }

}

}

private void grd\_SelectionChanged(object sender, SelectionChangedEventArgs e) {}

private void TextBox\_TextChanged(object sender, TextChangedEventArgs e){}

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

}

}

**Код xaml:**

Window x:Class="WpfApp31.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:WpfApp31"

mc:Ignorable="d"

Title="MainWindow" Height="616.154" Width="881.818">

<Grid Margin="0,0,2,1" Height="Auto" VerticalAlignment="Stretch" HorizontalAlignment="Stretch">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="263\*"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#AFEEEE" Offset="0"/>

<GradientStop Color="#E0FFFF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<DataGrid Name="grid" ItemsSource="{Binding}" AlternatingRowBackground="LightBlue" AlternationCount="2" Margin="109.215,254.842,0,96.667" SelectionChanged="grd\_SelectionChanged" HorizontalAlignment="Left" Width="318.623" UseLayoutRounding="False">

<DataGrid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#87CEFA" Offset="1"/>

</LinearGradientBrush>

</DataGrid.Background>

</DataGrid>

<TextBlock HorizontalAlignment="Left" Margin="109,148,0,0" TextWrapping="Wrap" Text=" Введите А (целое число):" VerticalAlignment="Top" Height="24" Width="269" FontSize="18" FontFamily="Sitka Text" FontWeight="Bold" Foreground="#FFFDFDFD">

<TextBlock.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#1E90FF" Offset="1"/>

</LinearGradientBrush>

</TextBlock.Background>

</TextBlock>

<TextBlock HorizontalAlignment="Left" Margin="109,194,0,0" TextWrapping="Wrap" Text="А =" VerticalAlignment="Top" FontSize="20" Foreground="#FF640D0D" FontWeight="Bold" FontFamily="Sitka Text" RenderTransformOrigin="0.371,1.393" Width="38" Height="24" TextAlignment="Center"/>

<TextBox Name="textbox" HorizontalAlignment="Left" Height="34" Margin="152,194,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="84" TextChanged="TextBox\_TextChanged" FontSize="18" RenderTransformOrigin="1.292,0.141" Foreground="#FFF7F1F1" SelectionBrush="#FFF2E9EA">

<TextBox.Background>

<LinearGradientBrush EndPoint="0.5,1" MappingMode="RelativeToBoundingBox" StartPoint="0.5,0">

<GradientStop Color="Black" Offset="1"/>

<GradientStop Color="#87CEFA" Offset="0.842"/>

</LinearGradientBrush>

</TextBox.Background>

</TextBox>

<Button Name="butt1" Content="Вычислить" HorizontalAlignment="Left" Margin="509,255,0,0" VerticalAlignment="Top" Width="188" Height="55" Click="but1\_Click" FontSize="18" FontWeight="Bold" FontFamily="Sitka Text" Foreground="#FFFDFDFD" RenderTransformOrigin="0.416,-0.205">

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#1E90FF" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Button Name="butt2" Content="Выход" HorizontalAlignment="Left" Margin="509,435,0,0" VerticalAlignment="Top" Width="188" Height="54" Click="but2\_Click" FontSize="18" FontWeight="Bold" FontFamily="Sitka Text" OpacityMask="#FFF7F5F5" Foreground="#FFFDFDFD" RenderTransformOrigin="0.457,-0.269">

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#1E90FF" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<TextBlock HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top" Margin="436,56,0,0" Height="30" Width="363" Foreground="#FFFDFDFD">

<TextBlock.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#1E90FF" Offset="1"/>

</LinearGradientBrush>

</TextBlock.Background><Run FontWeight="Bold" FontSize="18" FontFamily="Sitka Text" Text=" Вычисляется"/><Run Text=":"/></TextBlock>

<TextBlock HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top" Margin="536,142,0,0"/>

<Image Margin="350,68,0,375" HorizontalAlignment="Left" Width="347" VerticalAlignment="Top"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Height="56" Margin="509,315,0,0" VerticalAlignment="Top" Width="188" Click="Write\_Click" FontSize="18" FontWeight="Bold" FontFamily="Sitka Text" OpacityMask="#FFF7F5F5" Foreground="#FFFDFDFD" RenderTransformOrigin="0.457,-0.269" >

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#1E90FF" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Button Content="Вывести из файла" HorizontalAlignment="Left" Height="54" Margin="509,376,0,0" VerticalAlignment="Top" Width="188" Click="Button\_Click\_2" FontSize="18" FontWeight="Bold" FontFamily="Sitka Text" OpacityMask="#FFF7F5F5" Foreground="#FFFDFDFD" RenderTransformOrigin="0.457,-0.269">

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#1E90FF" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Image Margin="436,86,62,418.667" Source="Image1.png" Stretch="Fill"/>

</Grid>

</Window>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 6**

**Тема:**

# Создание приложения с использованием компонента ListBox

Студент: А.Ю. Топчиева

Группа: 2ПКС-420

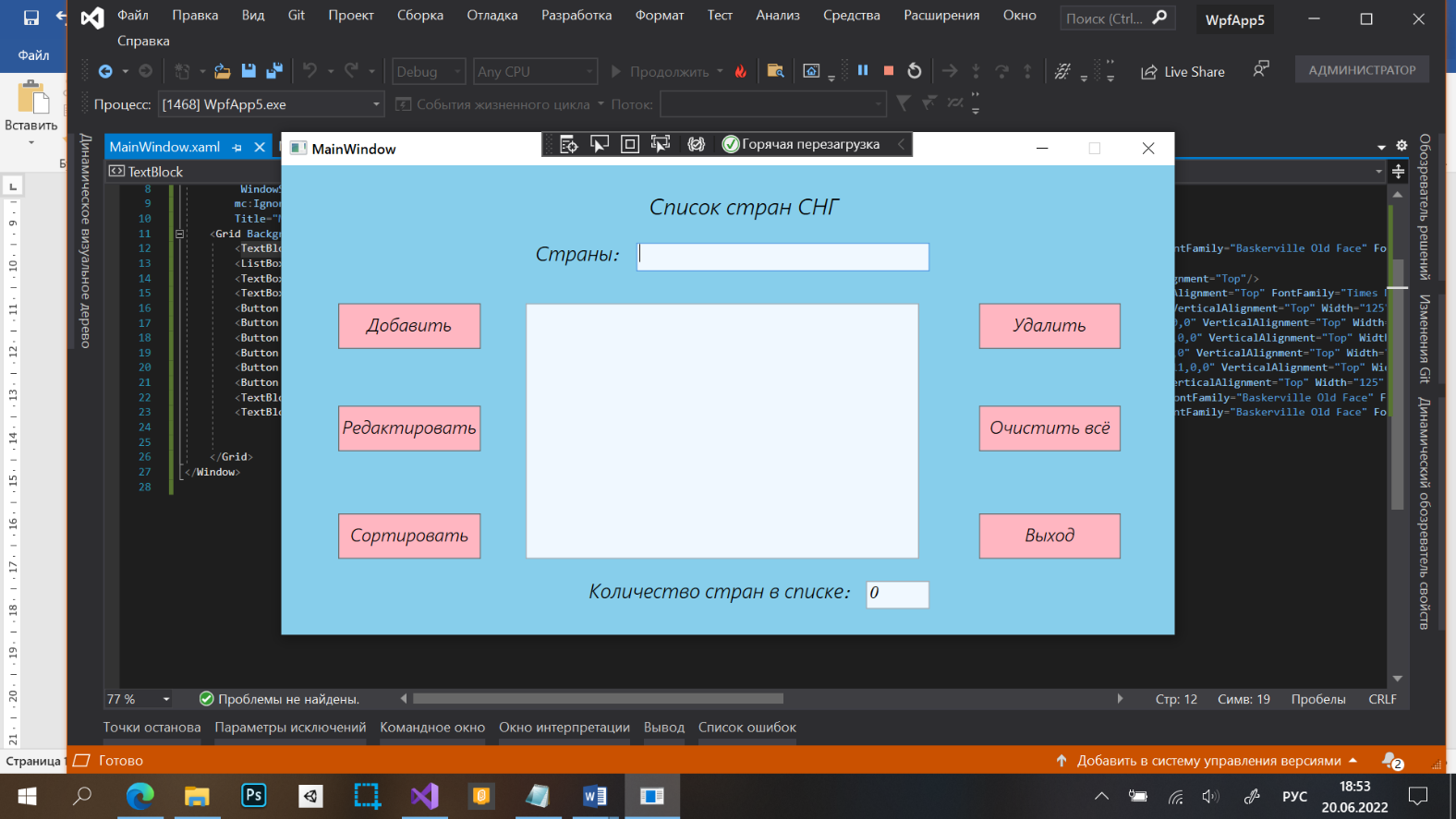
Преподаватель: И.В. Киреева

Дата: 20.06.2022

**Цель работы:** работа с ListBox

В проекте должны быть:

* окно ввода страны, обработать исключения(пробелы и тд)
* кнопка «добавить», она добавляет в ListBox введённое предложение
* кнопка «изменить», выбрать страну в ListBox, при нажатии на кнопку открывается изменение этого слова с мигающим курсором
* кнопка «удалить», также нужно выбрать и нажать кнопку, появится окно подтверждения с да/нет
* очистить список, тоже самое, что и удаление, но предупреждение с восклицательным знаком
* сортировка по алфавиту
* счетчик записей под окном ввода
* кнопка выхода

**Ниже на рисунках представлены фрагменты работающего проекта**

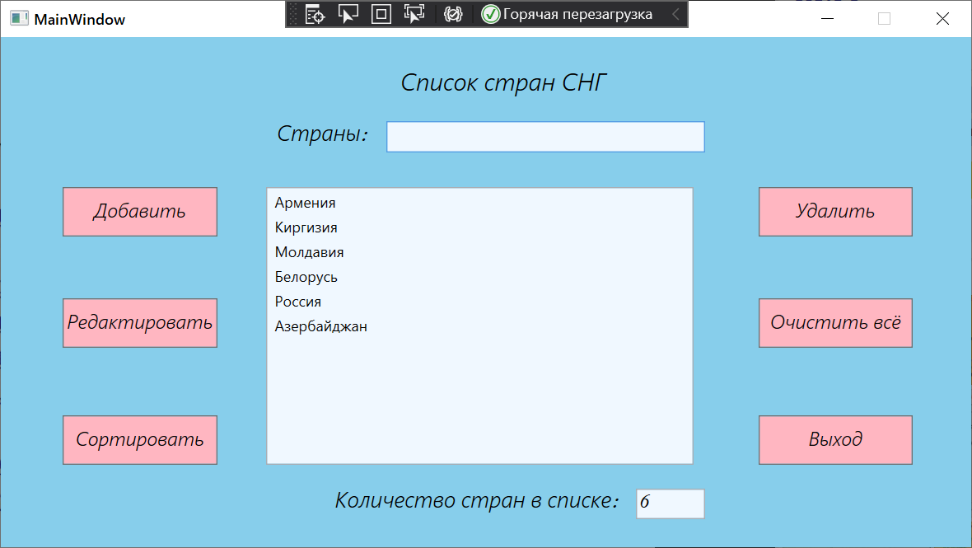
Рисунок 7. Форма проекта

Рисунок 8. Ввод в ListBox

**Код программы С#:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApp5

{

public partial class MainWindow : Window

{

int quantity\_products = 0; //количество продуктов

public MainWindow()

{

InitializeComponent();

input.Focus();

quantity.Text = quantity\_products.ToString();

}

private void click\_exit(object sender, RoutedEventArgs e)//выход из организации списка (с закрытием списка)

{

this.Close();

}

private void click\_add(object sender, RoutedEventArgs e)//добавление продукта

{

if (input.Text.Trim() == "")

{

MessageBox.Show("Введенно неверное значение!");

input.Focus();

}

else

{

quantity\_products += 1;

quantity.Text = quantity\_products.ToString();

list.Items.Add(input.Text.Trim());

input.Clear();

input.Focus();

}

}

private void click\_delete(object sender, RoutedEventArgs e)//удаление продукта

{

if (list.SelectedValue != null)

{

MessageBoxResult result = MessageBox.Show("Удалить страну из списка?", "Удалить", MessageBoxButton.YesNo, MessageBoxImage.Information);

if (result == MessageBoxResult.Yes)

{

list.Items.Remove(list.SelectedValue);

quantity\_products -= 1;

quantity.Text = quantity\_products.ToString();

if (quantity\_products < 0)

{

MessageBox.Show("Списк пуст, невозможно удаление элемента.");

quantity\_products = 0;

quantity.Text = quantity\_products.ToString();

}

input.Focus();

}

input.Focus();

}

else if (quantity\_products >= 0) { MessageBox.Show("Выделите элемент в списке, который хотите удалить."); input.Focus();}

else

{

MessageBox.Show("Сначало добавте элемент в список.");

quantity\_products = 0;

quantity.Text = quantity\_products.ToString();

input.Focus();

}

}

private void click\_clear(object sender, RoutedEventArgs e)//очистка списка

{

MessageBoxResult result = MessageBox.Show("Вы уверены, что хотите очистить весь список?", "Очистить всё", MessageBoxButton.YesNo, MessageBoxImage.Warning);

if (result == MessageBoxResult.Yes)

{

if (quantity\_products <= 0)

{

MessageBox.Show("Очищение пустого списка невозможно.");

input.Focus();

}

else

{

list.Items.Clear();

quantity\_products = 0;

quantity.Text = quantity\_products.ToString();

input.Focus();

}

}

input.Focus();

}

private void click\_sort(object sender, RoutedEventArgs e)//сортировка

{

list.Items.SortDescriptions.Add(new System.ComponentModel.SortDescription("", System.ComponentModel.ListSortDirection.Ascending));

input.Focus();

}

private void click\_edit(object sender, RoutedEventArgs e)//редактирование

{

if (list.SelectedValue != null)

{

try

{

input.Text = list.SelectedValue.ToString();

list.Items.Remove(list.SelectedValue);

quantity\_products -= 1;

quantity.Text = quantity\_products.ToString();

}

catch { }

}

else

{

MessageBox.Show("Выделите элемент в окне списка для его редактирования!");

input.Focus();

}

}

}

}

**Код xaml:**

<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"

ResizeMode="CanMinimize"

xmlns:local="clr-namespace:WpfApp5"

WindowStartupLocation="CenterScreen"

mc:Ignorable="d"

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

<Grid Background="SkyBlue" >

<TextBlock HorizontalAlignment="Center" Height="37" Margin="248,26,176.667,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="370" FontSize="20" FontFamily="Baskerville Old Face" FontStyle="Italic" RenderTransformOrigin="0.407,0.246"><Run Text="&#x9;Список "/><Run Text="стран СНГ"/></TextBlock>

<ListBox Background="AliceBlue" Name ="list" HorizontalAlignment="Right" Height="225" Margin="0,121,224.667,0" VerticalAlignment="Top" Width="346"/>

<TextBox Background="AliceBlue" Name ="input" Height="25" Margin="312,68,0,0" TextWrapping="Wrap" Width="258" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBox Background="AliceBlue" Name ="quantity" Height="25" Margin="514,365,0,0" TextWrapping="Wrap" Width="56" HorizontalAlignment="Left" VerticalAlignment="Top" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" TextDecorations="{x:Null}" RenderTransformOrigin="2.294,0.782"/>

<Button Background="LightPink" Name ="buttton\_add" Click ="click\_add" Content="Добавить" HorizontalAlignment="Left" Height="40" Margin="50,121,0,0" VerticalAlignment="Top" Width="125" FontFamily="Baskerville Old Face" FontSize="16" FontStyle="Italic"/>

<Button Background="LightPink" Name ="button\_delete" Click ="click\_delete" Content="Удалить" HorizontalAlignment="Left" Height="40" Margin="613,121,0,0" VerticalAlignment="Top" Width="125" FontFamily="Baskerville Old Face" FontSize="16" FontStyle="Italic" RenderTransformOrigin="0.53,-0.594"/>

<Button Background="LightPink" Name ="button\_edit" Click ="click\_edit" Content="Редактировать" HorizontalAlignment="Left" Height="40" Margin="50,211,0,0" VerticalAlignment="Top" Width="125" FontFamily="Baskerville Old Face" FontSize="16" FontStyle="Italic"/>

<Button Background="LightPink" Name ="button\_sort" Click ="click\_sort" Content="Сортировать" HorizontalAlignment="Left" Height="40" Margin="50,306,0,0" VerticalAlignment="Top" Width="125" FontFamily="Baskerville Old Face" FontSize="16" FontStyle="Italic"/>

<Button Background="LightPink" Name ="button\_clear" Click ="click\_clear" Content="Очистить всё" HorizontalAlignment="Left" Height="40" Margin="613,211,0,0" VerticalAlignment="Top" Width="125" FontFamily="Baskerville Old Face" FontSize="16" FontStyle="Italic"/>

<Button Background="LightPink" Name ="button\_exit" Click ="click\_exit" Content="Выход" HorizontalAlignment="Left" Height="40" Margin="613,306,0,0" VerticalAlignment="Top" Width="125" FontFamily="Baskerville Old Face" FontSize="16" FontStyle="Italic"/>

<TextBlock HorizontalAlignment="Center" Height="25" Margin="248,365,200.667,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="346" FontSize="18" FontFamily="Baskerville Old Face" FontStyle="Italic" RenderTransformOrigin="0.491,0.453"><Run Text=" Количество "/><Run Text="стран"/><Run Text=" в списке: "/></TextBlock>

<TextBlock HorizontalAlignment="Center" Height="25" Margin="156,68,446.667,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="192" FontSize="18" FontFamily="Baskerville Old Face" FontStyle="Italic" RenderTransformOrigin="1.18,0.843"><Run Text="&#x9;"/><Run Text="Страны"/><Run Text=": "/></TextBlock>

</Grid>

</Window>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 7**

**Тема:**

# Создание приложения «Калькулятор»

Студент: А.Ю. Топчиева

Группа: 3ПКС-420

Преподаватель: Е.Л. Альшакова

Преподаватель: И.В. Сибирев

Дата: 05.12.2022

**Цель работы:** разработать калькулятор на языке С#. В проекте использовать прием создания одного обработчика для нескольких событий.

**Ниже на рисунке представлен фрагмент работающего проекта**

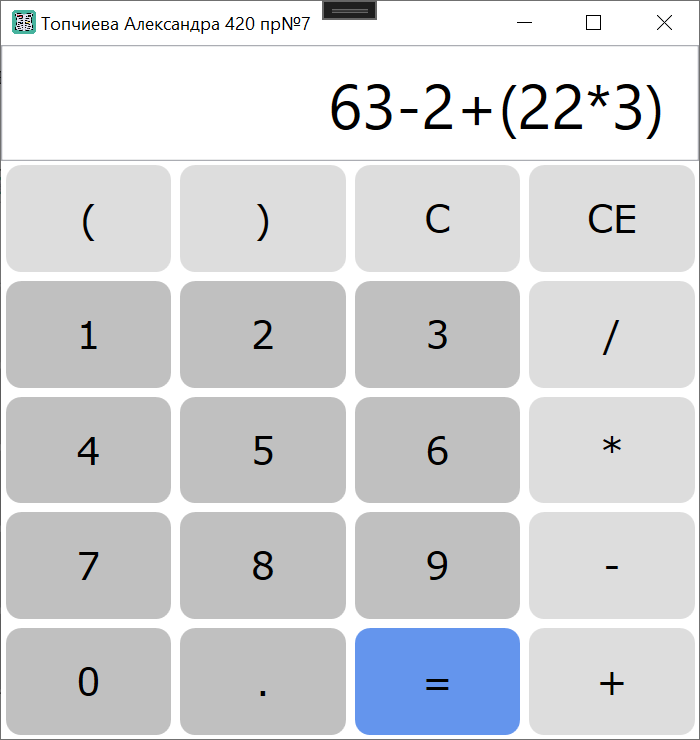


Рисунок 9. Калькулятор

**Код программы C#:**

using System;

using System.Data;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

namespace WpfApp\_пр7

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

foreach (UIElement o1 in MainRoot.Children)

{

if(o1 is Button)

{

((Button)o1).Click += Button\_Click;

}

}

}

//ограничение на ввод

private void textBox1\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

e.Handled = "0123456789+-\*/().".IndexOf(e.Text) < 0;

}

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

{

try

{

string str = (string)((Button)e.OriginalSource).Content;

if (str == "CE") { textLabel.Text = ""; }

else if (str == "C") { textLabel.Text = textLabel.Text.Remove(textLabel.Text.Length - 1, 1); }

else if (str == "=")

{

textLabel.Text = new DataTable().Compute(textLabel.Text, null).ToString();

textLabel.Text = textLabel.Text.Replace(",", ".");

}

else { textLabel.Text += str; }

}

catch (Exception ex) //обработка исключения

{

MessageBox.Show(ex.Message);

}

}

}

}

**Код xaml:**

<Window x:Class="WpfApp\_пр7.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:WpfApp\_пр7"

mc:Ignorable="d"

Icon="C:\Users\noteb\OneDrive\Рабочий стол\Для УП 01.ico"

Title="Топчиева Александра 420 пр№7" Height="500" Width="480" >

<Window.Resources>

<Style TargetType="Button" x:Key="RoundButton" >

<Style.Resources>

<Style TargetType="Border" >

<Setter Property="CornerRadius" Value="10"/>

<Setter Property="Control.Margin" Value="3" />

<Setter Property="Control.FontFamily" Value="Verdana" />

<Setter Property="Control.FontSize" Value="25" />

</Style>

</Style.Resources>

</Style>

</Window.Resources>

<Grid x:Name="MainRoot" Background="White">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<TextBox Grid.Row="0" Grid.ColumnSpan="4" PreviewTextInput="textBox1\_PreviewTextInput" x:Name="textLabel" ToolTip="Введите значения для вычисления" Padding="20,0,20,0" FontSize="42" HorizontalContentAlignment="Right" VerticalContentAlignment="Center" Margin="0,0,0,80" Grid.RowSpan="2" />

<Button BorderThickness="0" Grid.Row="1" Grid.Column="0" Style="{DynamicResource RoundButton}" >(</Button>

<Button BorderThickness="0" Grid.Row="1" Grid.Column="1" Style="{DynamicResource RoundButton}" >)</Button>

<Button BorderThickness="0" Grid.Row="1" Grid.Column="2" Style="{DynamicResource RoundButton}" >C </Button>

<Button BorderThickness="0" Grid.Row="1" Grid.Column="3" Style="{DynamicResource RoundButton}">CE</Button>

<Button BorderThickness="0" Grid.Row="2" Grid.Column="0" Style="{DynamicResource RoundButton}" Background="#C0C0C0">1</Button>

<Button BorderThickness="0" Grid.Row="2" Grid.Column="1" Style="{DynamicResource RoundButton}" Background="#C0C0C0">2</Button>

<Button BorderThickness="0" Grid.Row="2" Grid.Column="2" Style="{DynamicResource RoundButton}" Background="#C0C0C0">3</Button>

<Button BorderThickness="0" Grid.Row="2" Grid.Column="3" Style="{DynamicResource RoundButton}">/</Button>

<Button BorderThickness="0" Grid.Row="3" Grid.Column="0" Style="{DynamicResource RoundButton}" Background="#C0C0C0">4</Button>

<Button BorderThickness="0" Grid.Row="3" Grid.Column="1" Style="{DynamicResource RoundButton}" Background="#C0C0C0">5</Button>

<Button BorderThickness="0" Grid.Row="3" Grid.Column="2" Style="{DynamicResource RoundButton}" Background="#C0C0C0">6</Button>

<Button BorderThickness="0" Grid.Row="3" Grid.Column="3" Style="{DynamicResource RoundButton}">\*</Button>

<Button BorderThickness="0" Grid.Row="4" Grid.Column="0" Style="{DynamicResource RoundButton}" Background="#C0C0C0">7</Button>

<Button BorderThickness="0" Grid.Row="4" Grid.Column="1" Style="{DynamicResource RoundButton}" Background="#C0C0C0">8</Button>

<Button BorderThickness="0" Grid.Row="4" Grid.Column="2" Style="{DynamicResource RoundButton}" Background="#C0C0C0">9</Button>

<Button BorderThickness="0" Grid.Row="4" Grid.Column="3" Style="{DynamicResource RoundButton}" >-</Button>

<Button BorderThickness="0" Grid.Row="5" Grid.Column="0" Style="{DynamicResource RoundButton}" Background="#C0C0C0">0</Button>

<Button BorderThickness="0" Grid.Row="5" Grid.Column="1" Style="{DynamicResource RoundButton}" Background="#C0C0C0">.</Button>

<Button BorderThickness="0" Grid.Row="5" Grid.Column="2" Style="{DynamicResource RoundButton}" Background="#6495ED">=</Button>

<Button BorderThickness="0" Grid.Row="5" Grid.Column="3" Style="{DynamicResource RoundButton}" >+</Button>

</Grid>

</Window>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 8**

**Тема:**

# Создание приложения для проверки орфографии

Студент: А.Ю. Топчиева

Группа: 3ПКС-420

Преподаватель: Е.Л. Альшакова

Преподаватель: И.В. Сибирев

Дата: 05.12.2022

**Цель работы:** 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.

**Ниже на рисунке представлен фрагмент работающего проекта**

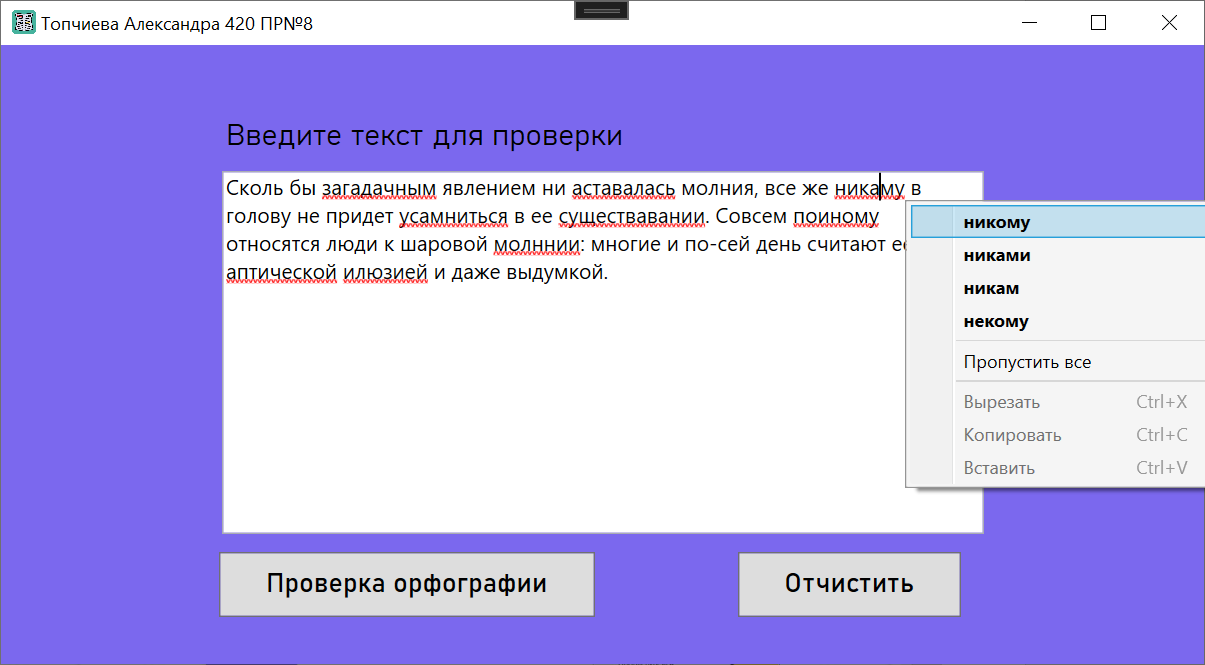


Рисунок 10. Форма проекта

**Код программы C#:**

using System.Windows;

namespace WpfApp\_пр8

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

text.SpellCheck.IsEnabled = true;

}

private void Button\_Click2(object sender, RoutedEventArgs e)

{

text.Clear();

}

}

}

**Код xaml:**

<Window x:Class="WpfApp\_пр8.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:WpfApp\_пр8"

mc:Ignorable="d"

Icon="C:\Users\noteb\OneDrive\Рабочий стол\Для УП 01.ico"

Title="Топчиева Александра 420 ПР№8" Height="450" Width="800" Background="#7B68EE">

<Grid>

<TextBox Name="text" HorizontalAlignment="Center" TextWrapping="Wrap" VerticalAlignment="Top" Width="508" Height="242" Margin="0,84,0,0" FontSize="14"/>

<Button Content="Проверка орфографии"ToolTip="Проверка орфографии введеного текста" HorizontalAlignment="Left" Height="43" Margin="145,338,0,0" VerticalAlignment="Top" Width="251" FontSize="18" FontFamily="Bahnschrift" Click="Button\_Click"/>

<Label Content="Введите текст для проверки" HorizontalAlignment="Left" Height="35" Margin="145,44,0,0" VerticalAlignment="Top" Width="480" FontSize="20" FontFamily="Bahnschrift Light"/>

<Button Content="Отчистить" ToolTip="Отчистить всю панель от текста" HorizontalAlignment="Left" Height="43" Margin="491,338,0,0" VerticalAlignment="Top" Width="149" FontSize="18" FontFamily="Bahnschrift" Click="Button\_Click2"/>

</Grid>

</Window>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 9**

**Тема:**

# Создание приложения для организации взаимодействия с MS Word

Студент: А.Ю. Топчиева

Группа: 3ПКС-420

Преподаватель: Е.Л. Альшакова

Преподаватель: И.В. Сибирев

Дата: 05.12.2022

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

**Ниже на рисунках представлены фрагменты работающего проекта**

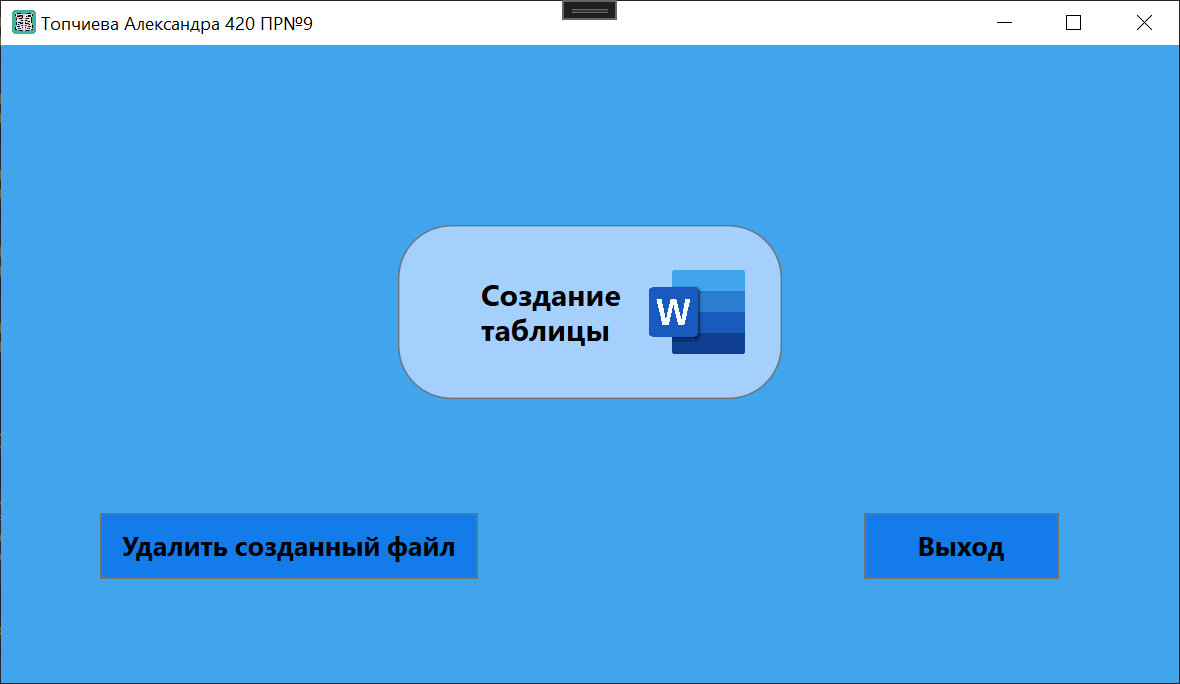


Рисунок 11. Форма проекта

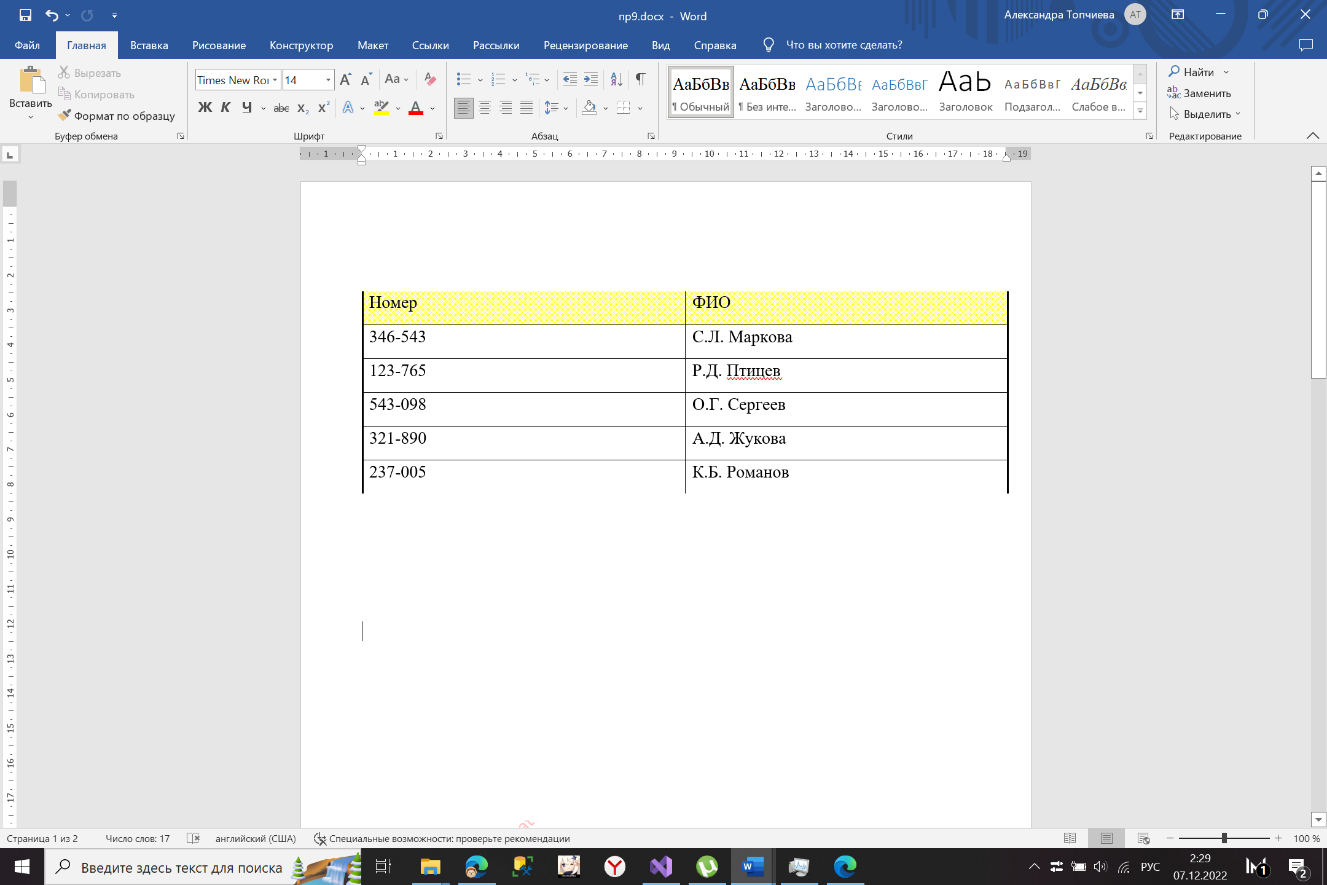


Рисунок 12. Пример таблицы

**Код программы C#:**

using Syncfusion.DocIO.DLS;

using System;

using System.Diagnostics;

using System.IO;

using System.Windows;

using Window = System.Windows.Window;

namespace WpfApp\_пр9

{

public partial class MainWindow : Window

{

public MainWindow()

{

}

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

{

WordDocument document = new WordDocument();

IWSection section = document.AddSection();

IWTable table = section.AddTable();

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

style.CharacterFormat.FontName = "Times New Roman";

style.CharacterFormat.FontSize = 14;

style.ParagraphFormat.BeforeSpacing = 0;

style.ParagraphFormat.AfterSpacing = 8;

style.ParagraphFormat.LineSpacing = 14;

table.ResetCells(6, 2);

WTableRow row = table.Rows[0];

table.ApplyStyle(BuiltinTableStyle.TableGrid4);

row.IsHeader = true;

row.Cells[0].AddParagraph().AppendText("Номер");

row = table.Rows[0];

row.Cells[1].AddParagraph().AppendText("ФИО");

row = table.Rows[1];

row.Cells[0].AddParagraph().AppendText("346-543");

row.Cells[1].AddParagraph().AppendText("С.Л. Маркова");

row = table.Rows[2];

row.Cells[0].AddParagraph().AppendText("123-765");

row.Cells[1].AddParagraph().AppendText("Р.Д. Птицев");

row = table.Rows[3];

row.Cells[0].AddParagraph().AppendText("543-098");

row.Cells[1].AddParagraph().AppendText("О.Г. Сергеев");

row = table.Rows[4];

row.Cells[0].AddParagraph().AppendText("321-890");

row.Cells[1].AddParagraph().AppendText("А.Д. Жукова");

row = table.Rows[5];

row.Cells[0].AddParagraph().AppendText("237-005");

row.Cells[1].AddParagraph().AppendText("К.Б. Романов");

section.AddParagraph();

document.Save("C:\\f\\пр9.docx");

Process wordProcess = new Process();

wordProcess.StartInfo.FileName = "C:\\f\\пр9.docx";

wordProcess.StartInfo.UseShellExecute = true;

wordProcess.Start();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

Environment.Exit(0);

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

string path = @"C:\f\пр9.docx";

File.Delete(path);

if(!File.Exists(path))

{

MessageBox.Show("Файл удален");

}

}

}

}

**Код xaml:**

<Window x:Class="WpfApp\_пр9.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:WpfApp\_пр9"

mc:Ignorable="d"

Icon="C:\Users\noteb\OneDrive\Рабочий стол\Для УП 01.ico"

Title="Топчиева Александра 420 ПР№9" Height="450" Width="800" Background="#FF41A5EE">

<Grid>

<Button HorizontalAlignment="Center" Background="#FFA5CFFC" Height="116" Margin="0,120,0,0" VerticalAlignment="Top" Width="256" Click="Button\_Click" FontFamily="Arial Rounded MT Bold" FontSize="20">

<Image Source="C:\Users\noteb\OneDrive\Рабочий стол\word\_logo\_icon.png" Width="219" HorizontalAlignment="Right" Height="75" ></Image>

<Button.Resources>

<Style TargetType="{x:Type Border}">

<Setter Property="CornerRadius" Value="35"/>

</Style>

</Button.Resources>

</Button>

<Button Content="Выход" HorizontalAlignment="Left" Height="44" Margin="575,312,0,0" VerticalAlignment="Top" Width="130" Click="Button\_Click\_1" FontFamily="Arial Rounded MT Bold" FontSize="18" Background="#FF147BEA"/>

<Button Content="Удалить созданный файл" HorizontalAlignment="Left" Height="44" Margin="66,312,0,0" VerticalAlignment="Top" Width="252" Click="Button\_Click\_2" FontFamily="Arial Rounded MT Bold" FontSize="18" Background="#FF147BEA"/>

<Label Content="Создание &#xD;&#xA;таблицы" HorizontalAlignment="Left" Height="56" Margin="315,150,0,0" VerticalAlignment="Top" Width="125" FontFamily="Arial Rounded MT Bold" FontSize="20"/>

</Grid>

</Window>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 10**

**Тема:**

# Создание приложения с использованием функций MS Excel

Студент: А.Ю. Топчиева

Группа: 3ПКС-420

Преподаватель: Е.Л. Альшакова

Преподаватель: И.В. Сибирев

Дата: 05.12.2022

**Цель работы:** создание собственного проекта с использованием функций MS Excel не используя функцию нахождения числа Pi.

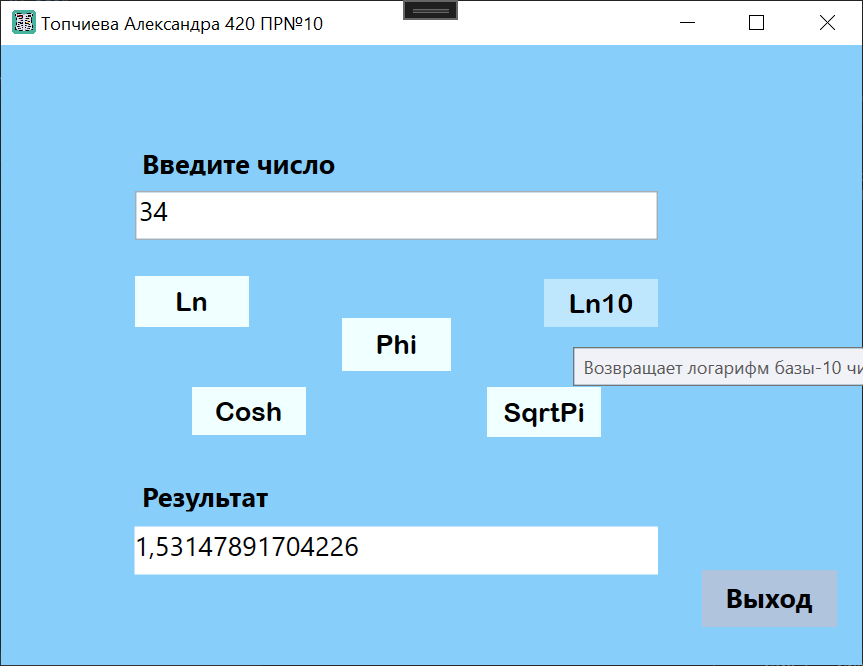
**Ниже на рисунке представлен фрагмент работающего проекта**

Рисунок 13. Пример формы

**Код программы C#:**

using System;

using System.Windows;

namespace WpfApp\_пр10

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

try

{

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

var ln = xl.WorksheetFunction.Ln(Convert.ToDouble(tbo.Text));

tbl.Text = Convert.ToString(ln);

xl.Quit();

}

catch (Exception ex) //обработка исключения

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

try

{

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

var log = xl.WorksheetFunction.Log10(Convert.ToDouble(tbo.Text));

tbl.Text = Convert.ToString(log);

xl.Quit();

}

catch (Exception ex) //обработка исключения

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

try

{

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

var cosh = xl.WorksheetFunction.Cosh(Convert.ToDouble(tbo.Text));

tbl.Text = Convert.ToString(cosh);

xl.Quit();

}

catch (Exception ex) //обработка исключения

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

try

{

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

var sqrtpi = xl.WorksheetFunction.SqrtPi(Convert.ToDouble(tbo.Text));

tbl.Text = Convert.ToString(sqrtpi);

xl.Quit();

}

catch (Exception ex) //обработка исключения

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_4(object sender, RoutedEventArgs e)

{

try

{

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

var phi = xl.WorksheetFunction.Phi(Convert.ToDouble(tbo.Text));

tbl.Text = Convert.ToString(phi);

xl.Quit();

}

catch (Exception ex) //обработка исключения

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_5(object sender, RoutedEventArgs e)

{

Environment.Exit(0);

}

}

}

**Код xaml:**

<Window x:Class="WpfApp\_пр10.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:WpfApp\_пр10"

mc:Ignorable="d"

Icon="C:\Users\noteb\OneDrive\Рабочий стол\Для УП 01.ico"

Title="Топчиева Александра 420 ПР№10" Height="450" Width="600" Background="#87CEFA">

<Grid>

<Button Content="Ln" BorderThickness="0" HorizontalAlignment="Left" Background="Azure" ToolTip="Возвращает натуральный логарифм числа" Height="34" Margin="89,154,0,0" VerticalAlignment="Top" Width="76" Click="Button\_Click" FontFamily="Arial Rounded MT Bold" FontSize="18"/>

<TextBlock Name="tbl" Background="White" HorizontalAlignment="Left" Height="32" Margin="89,321,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="349" FontSize="18"/>

<TextBox Name="tbo" HorizontalAlignment="Left" Height="33" Margin="89,97,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="349" FontSize="18"/>

<Button Content="Ln10" BorderThickness="0" HorizontalAlignment="Left" Background="Azure" ToolTip="Возвращает логарифм базы-10 числа" Height="32" Margin="362,156,0,0" VerticalAlignment="Top" Width="76" Click="Button\_Click\_1" FontSize="18" FontFamily="Arial Rounded MT Bold"/>

<Button Content="Cosh" BorderThickness="0" HorizontalAlignment="Left" Background="Azure" ToolTip="Возвращает гиперболический косин числа" Height="32" Margin="127,228,0,0" VerticalAlignment="Top" Width="76" Click="Button\_Click\_2" FontSize="18" FontFamily="Arial Rounded MT Bold"/>

<Button Content="SqrtPi" BorderThickness="0" HorizontalAlignment="Left" Background="Azure" ToolTip="Возвращает квадратный корень (номер \* pi)" Height="33" Margin="324,228,0,0" VerticalAlignment="Top" Width="76" Click="Button\_Click\_3" FontSize="18" FontFamily="Arial Rounded MT Bold" RenderTransformOrigin="0.601,0.427"/>

<Label Content="Введите число" HorizontalAlignment="Left" Height="28" Margin="89,64,0,0" VerticalAlignment="Top" Width="148" FontFamily="Arial Rounded MT Bold" FontSize="18"/>

<Label Content="Результат" HorizontalAlignment="Left" Height="30" Margin="89,286,0,0" VerticalAlignment="Top" Width="180" FontFamily="Arial Rounded MT Bold" FontSize="18"/>

<Button Content="Выход" BorderThickness="0" HorizontalAlignment="Left" Background="#B0C4DE" Height="38" Margin="467,350,0,0" VerticalAlignment="Top" Width="90" Click="Button\_Click\_5" FontFamily="Arial Rounded MT Bold" FontSize="18"/>

<Button Content="Phi" BorderThickness="0" HorizontalAlignment="Left" Background="Azure" ToolTip="Возвращает значение функции плотности для стандартного нормального распределения" Click="Button\_Click\_4" Height="35" Margin="227,182,0,0" VerticalAlignment="Top" Width="73" FontFamily="Arial Rounded MT Bold" FontSize="18"/>

</Grid>

</Window>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 11**

**Тема:**

# Создание поточного приложения

Студент: А.Ю. Топчиева

Группа: 3ПКС-420

Преподаватель: Е.Л. Альшакова

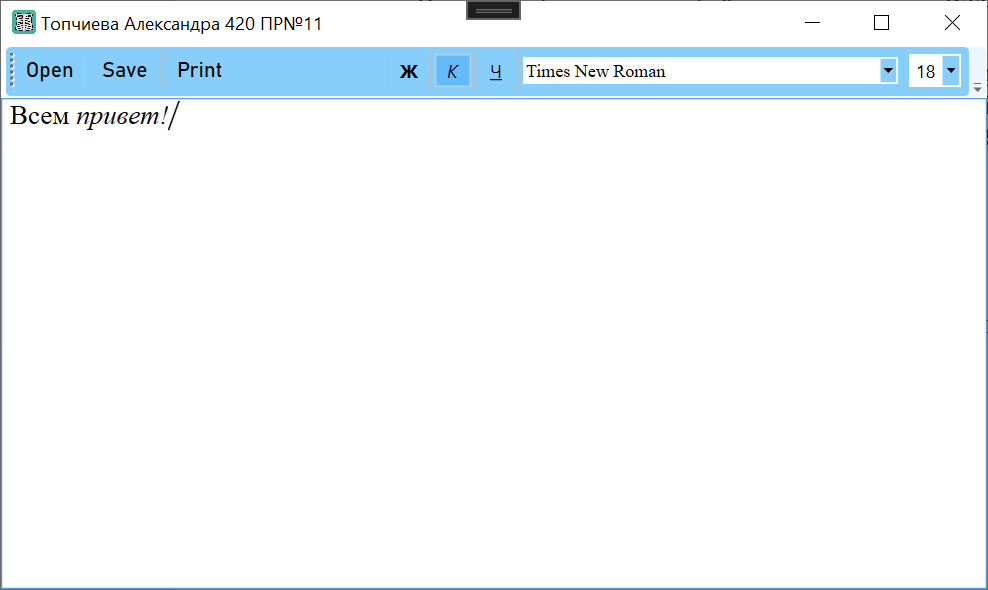
Преподаватель: И.В. Сибирев

Дата: 05.12.2022

**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.

**Ниже на рисунках представлены фрагменты работающего проекта**

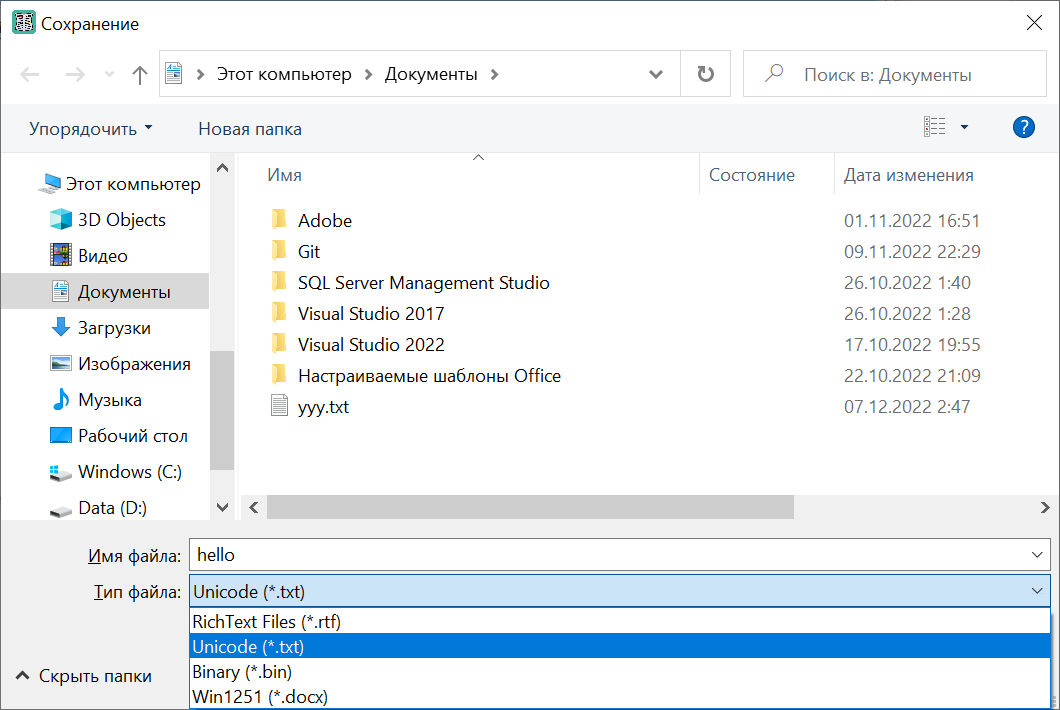
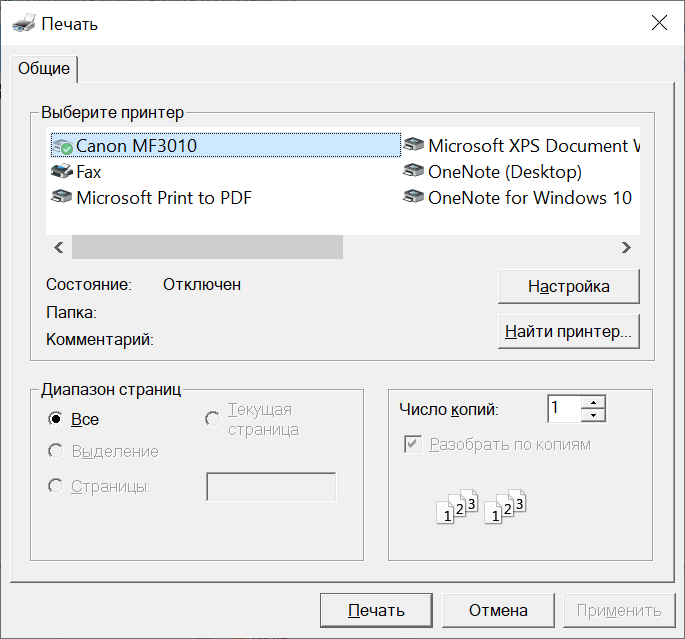
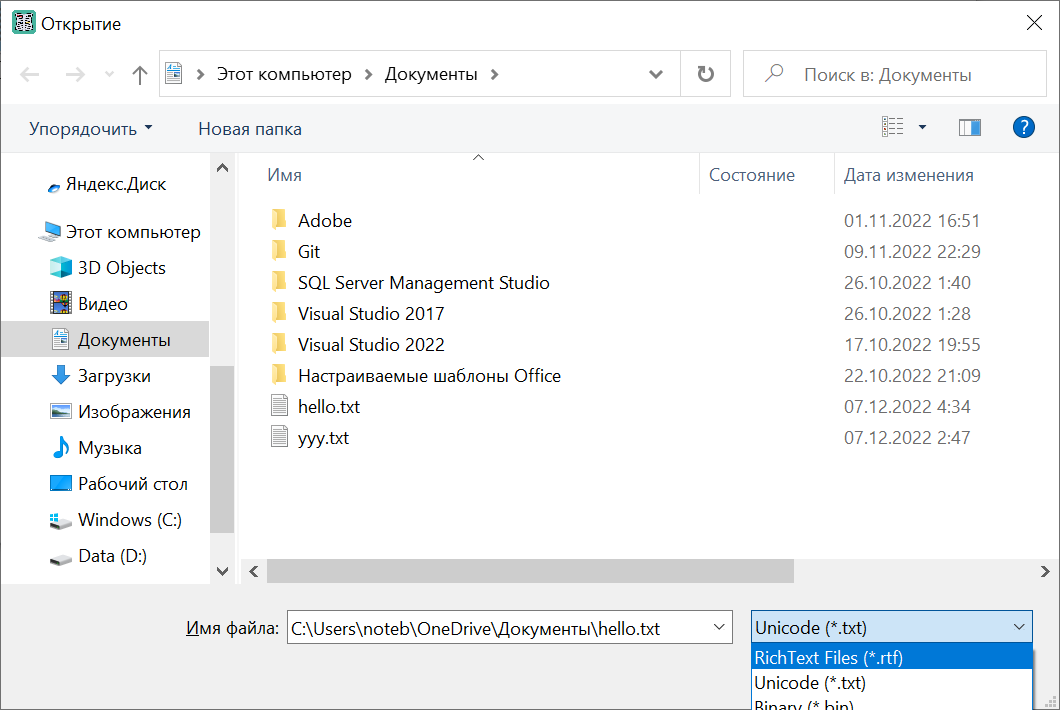
Рисунок 14. Исходный документ

Рисунок 15. Сохранение документа

Рисунок 16. Открытие документа Рисунок 17. Печать документа

**Код программы C#:**

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Documents;

using System.Windows.Media;

namespace WpfApp\_пр11

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

fontCombobox.ItemsSource = Fonts.SystemFontFamilies.OrderBy(f => f.Source);

fontSizeBox.ItemsSource = new List<double>() { 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 36, 48, 72 };

fontCombobox.SelectedItem = textEditor.FontFamily;

fontSizeBox.SelectedItem = textEditor.FontSize;

}

private async void openBtn\_Click(object sender, RoutedEventArgs e)

{

await Task.Delay(0);

try

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "RichText Files (\*.rtf)|\*.rtf|Unicode (\*.txt)|\*.txt|Binary (\*.bin)|\*.bin|Win1251 (\*.docx)|\*.docx|All Files (\*.\*)|\*.\*";

if (ofd.ShowDialog() == true)

{

TextRange doc = new TextRange(textEditor.Document.ContentStart, textEditor.Document.ContentEnd);

using (FileStream fs = new FileStream(ofd.FileName, FileMode.Open))

{

if (System.IO.Path.GetExtension(ofd.FileName).ToLower() == ".rtf")

doc.Load(fs, DataFormats.Rtf);

if (System.IO.Path.GetExtension(ofd.FileName).ToLower() == ".txt")

doc.Load(fs, DataFormats.Text);

if (System.IO.Path.GetExtension(ofd.FileName).ToLower() == ".bin")

doc.Load(fs, DataFormats.Text);

if (System.IO.Path.GetExtension(ofd.FileName).ToLower() == ".docx")

doc.Load(fs, DataFormats.Text);

}

}

}

catch (Exception ex) {MessageBox.Show(ex.Message);}

}

private async void saveBtn\_Click(object sender, RoutedEventArgs e)

{

await Task.Delay(0);

try

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "RichText Files (\*.rtf)|\*.rtf|Unicode (\*.txt)|\*.txt|Binary (\*.bin)|\*.bin|Win1251 (\*.docx)|\*.docx";

if (sfd.ShowDialog() == true)

{

TextRange doc = new TextRange(textEditor.Document.ContentStart, textEditor.Document.ContentEnd);

using (FileStream fs = File.Create(sfd.FileName))

{

if (System.IO.Path.GetExtension(sfd.FileName).ToLower() == ".rtf")

doc.Save(fs, DataFormats.Rtf);

if (System.IO.Path.GetExtension(sfd.FileName).ToLower() == ".docx")

using (StreamWriter writer = new StreamWriter(fs, Encoding.GetEncoding(1251)))

writer.WriteLine(doc.Text);

if (System.IO.Path.GetExtension(sfd.FileName).ToLower() == ".txt")

using (StreamWriter writer = new StreamWriter(fs, Encoding.UTF8))

writer.WriteLine(doc.Text);

if (System.IO.Path.GetExtension(sfd.FileName).ToLower() == ".bin")

using (BinaryWriter writer = new BinaryWriter(fs))

writer.Write(doc.Text);

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

try

{

PrintDialog pd = new PrintDialog();

if ((pd.ShowDialog() == true))

{

pd.PrintVisual(textEditor as Visual, "Print Visual");

pd.PrintDocument((((IDocumentPaginatorSource)textEditor.Document).DocumentPaginator), "Print Document");

}

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

private void ComboBox\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (fontCombobox.SelectedItem != null && textEditor != null)

{

textEditor.Selection.ApplyPropertyValue(RichTextBox.FontFamilyProperty, fontCombobox.SelectedItem);

textEditor.Focus();

}

}

private void ComboBox\_TextChanged(object sender, RoutedEventArgs e)

{

double size;

if (textEditor != null)

if (Double.TryParse(fontSizeBox.Text, out size))

{

textEditor.Selection.ApplyPropertyValue(Inline.FontSizeProperty, size);

textEditor.Focus();

}

}

private void textEditor\_SelectionChanged(object sender, RoutedEventArgs e)

{

try

{

object temp = textEditor.Selection.GetPropertyValue(Inline.FontWeightProperty);

btnBold.IsChecked = (temp != DependencyProperty.UnsetValue) && (temp.Equals(FontWeights.Bold));

temp = textEditor.Selection.GetPropertyValue(Inline.FontStyleProperty);

btnItalic.IsChecked = (temp != DependencyProperty.UnsetValue) && (temp.Equals(FontStyles.Italic));

temp = textEditor.Selection.GetPropertyValue(Inline.TextDecorationsProperty);

btnUnderline.IsChecked = (temp != DependencyProperty.UnsetValue) && (temp.Equals(TextDecorations.Underline));

temp = textEditor.Selection.GetPropertyValue(Inline.FontFamilyProperty);

fontCombobox.SelectedItem = temp;

temp = textEditor.Selection.GetPropertyValue(Inline.FontSizeProperty);

fontSizeBox.Text = temp.ToString();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

}

}

**Код xaml:**

<Window x:Class="WpfApp\_пр11.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:WpfApp\_пр11"

mc:Ignorable="d"

Icon="C:\Users\noteb\OneDrive\Рабочий стол\Для УП 01.ico"

Title="Топчиева Александра 420 ПР№11" Height="400" Width="700">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="35" />

<RowDefinition />

</Grid.RowDefinitions>

<ToolBar Height="35" Grid.Row="0" Background="#87CEFA">

<Button Content="Open" x:Name="openBtn" Click="openBtn\_Click" Height="22" Width="45" FontFamily="Bahnschrift" FontSize="14"/>

<Separator />

<Button Content="Save" x:Name="saveBtn" Click="saveBtn\_Click" HorizontalAlignment="Left" Height="30" Width="45" FontFamily="Bahnschrift" FontSize="14"/>

<Separator />

<Button Content="Print" x:Name="printBtn" Click="printBtn\_Click" Height="38" Width="45" FontFamily="Bahnschrift" FontSize="14"/>

<Separator />

<ToggleButton Height="30" Width="95"/>

<Separator />

<ToggleButton x:Name="btnBold" Command="EditingCommands.ToggleBold" Width="24" FontWeight="Bold">Ж</ToggleButton>

<Separator />

<ToggleButton x:Name="btnItalic" Command="EditingCommands.ToggleItalic" Width="24" FontStyle="Italic">К</ToggleButton>

<Separator />

<ToggleButton x:Name="btnUnderline" Command="EditingCommands.ToggleUnderline" Width="24">

<ToggleButton.Content>

<TextBlock TextDecorations="Underline">Ч</TextBlock>

</ToggleButton.Content>

</ToggleButton>

<Separator />

<ComboBox IsEditable="False" x:Name="fontCombobox" Width="250" SelectionChanged="ComboBox\_SelectionChanged">

<ComboBox.ItemTemplate>

<DataTemplate>

<TextBlock Text="{Binding}" FontFamily="{Binding}" />

</DataTemplate>

</ComboBox.ItemTemplate>

</ComboBox>

<Separator />

<ComboBox x:Name="fontSizeBox" IsEditable="True" TextBoxBase.TextChanged="ComboBox\_TextChanged">

</ComboBox>

</ToolBar>

<RichTextBox x:Name="textEditor" Grid.Row="2" SelectionChanged="textEditor\_SelectionChanged">

<RichTextBox.Resources>

<Style TargetType="{x:Type Paragraph}">

<Setter Property="Margin" Value="0"/>

</Style>

</RichTextBox.Resources>

</RichTextBox>

</Grid>

</Window>