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

Д.М. Борушко

*(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

[**Практические работы по Assemler** 8](#_Toc121263674)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1 8](#_Toc121263675)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 2 11](#_Toc121263676)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 3 17](#_Toc121263677)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 4 20](#_Toc121263678)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 5 24](#_Toc121263679)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 6 28](#_Toc121263680)

[**Практические работы по C#** 32](#_Toc121263681)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1 32](#_Toc121263682)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 2 38](#_Toc121263683)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 3 45](#_Toc121263684)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 4 52](#_Toc121263685)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 5 63](#_Toc121263686)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 6 69](#_Toc121263687)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 7 74](#_Toc121263688)

[ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ №8 79](#_Toc121263689)

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

# **Практические работы по Assemler**

## ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1

**Тема: «Работа с файлами»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Альшакова Е.Л.

Дата: 06.12.2022

**Цель работы**: Дано описание: X DD 100 DUP(?); числа со знаком. Записать в регистр EAX наименьший элемент массива X

**Ниже на рисунке представлен фрагмент работающего проекта:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Выполнение кода

**Код программы:**

format PE console

include 'win32a.inc'

start:

cinvoke printf, text1

cinvoke scanf, spp, N

stdcall min, X,[N]

mov edx,eax

cinvoke printf, text3, edx

cinvoke printf, text4

cinvoke system,p

invoke exit,0

entry start

section '.data' data readable writeable

text1 db 'Enter N: ',0

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

text4 db 10,13,'',0

p db 'pause',0

spp db '%d',0

X dd 13,34,65,66,7,54,3,2,5,68,8,5,37,43,3,76,100,43,6,8

N dd ?

section '.code' code readable executable

proc min num1, num2

mov ebx, [num1]

mov ecx, [num2]

mov eax,1000 ;сюда запишется мин эл

MIN1:cmp [ebx],eax

jge MIN2 ;если 1й эл меньше или равен второму то переходим

mov eax,[ebx]

MIN2:add ebx,4

loop MIN1

ret

endp

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'

**Вывод:** мною была написана программа для записи в регистр EAX наименьшего элемента массива X.

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 со стеком»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Альшакова Е.Л.

Дата: 06.12.2022

**Цель работы**: 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.

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

**Ниже на рисунке представлен фрагмент работающего проекта:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Выполнение кода

**Код программы:**

format PE console

include 'win32a.inc'

entry start

proc maxim

lea ebx,[X]

mov ecx,[N]

mov edx,[N]

inc edx

mov eax,0

mov [ID],ecx

cycl:

cmp eax,[ebx]

jg cycl1

mov [ID],edx

sub [ID],ecx

mov eax,[ebx]

cycl1:

add ebx,4

loop cycl

ret

endp

proc zamena

mov esi,X

mov ecx,[N]

mov edx,0

metka:

mov eax,[esi]

getabs:

neg eax ;находим модуль числа

js getabs

mov ebx,5

idiv ebx

mov ebx, 2

cmp ebx,edx

je metka1

jmp met

metka1:

mov eax, 0

mov [esi],eax

met:

mov edx,0

add esi,4

loop metka

ret

endp

start:

;;;;;;Обнулить отрицательные элементы

macro prntn massiv, N

{

local metka

mov esi,massiv

mov ecx,[N]

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

}

cinvoke printf,text2

prntn X, N

call zamena

cinvoke printf,text2

prntn X, N

call maxim

cinvoke printf,text1,eax

cinvoke printf,text3,[ID]

invoke sleep,5000

text1 db 10, 13, 'Max: %d', 0

text2 db 10,13,'Array: ', 0

text3 db 10, 13, 'Max elements ID: %d', 0

fmat db ' %d',0

p db 'pause', 0

ID dd ?

N dd 10

X dd 7,6,15,17,12,-12,4,0,-10,-22

data import

library msvcrt, 'MSVCRT.DLL',\

kernel132, 'KERNEL32.DLL'

import kernel132,\

sleep, 'Sleep'

import msvcrt,\

printf, 'printf',\

scanf,'scanf',\

exit, 'exit',\

setlocal, 'setlocal',\

system, 'system'

end data

**Вывод:** мною была разработана программа с использованием стека.

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: «Рекурсивные процедуры»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Альшакова Е.Л.

Дата: 06.12.2022

**Цель работы:** Разработать рекурсивную процедуру вычисления суммы чисел от 1 до 100.

**Ниже на рисунке представлен фрагмент работающего проекта:**

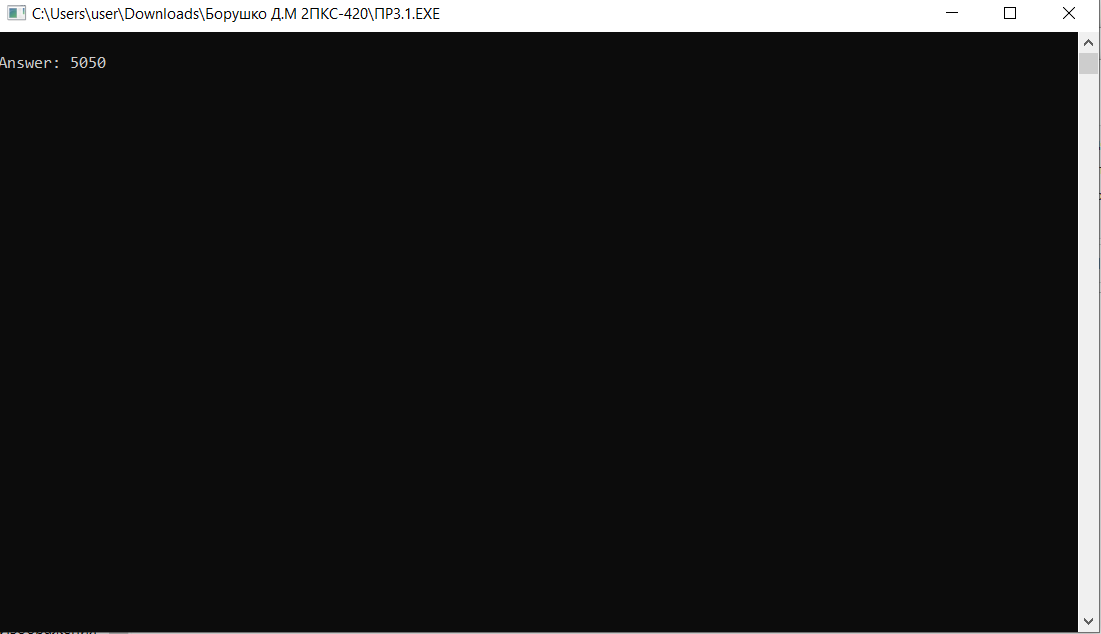
****

Рисунок 3. Выполнение кода

**Код программы:**

format PE console

include 'win32a.inc'

entry start

proc summa

cmp ebx, 0

jg f1

mov [sum], eax

ret

f1:

add eax, ebx

dec ebx

call summa

ret

endp

start:

;;;;;; рекурсивно вычислить сумму элементов от 0 до 100

mov eax,0

mov ebx,[n]

call summa

cinvoke printf,text4,[sum]

invoke sleep,5000

text4 db 10,13,'Answer: %d ', 0

fmat db ' %d',0

p db 'pause', 0

n dd 100

sum dd ?

data import

library msvcrt, 'MSVCRT.DLL',\

kernel132, 'KERNEL32.DLL'

import kernel132,\

sleep, 'Sleep'

import msvcrt,\

printf, 'printf',\

scanf,'scanf',\

exit, 'exit',\

setlocal, 'setlocal',\

system, 'system'

end data

**Вывод:** мною была разработана программа с использованием рекурсивных процедур.

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: «Основные арифметические инструкции»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Альшакова Е.Л.

Дата: 06.12.2022

**Цель работы:** Разместить в памяти компьютера матрицу 4\*3 целых

без знаковых чисел. Написать программу, позволяющую найти сумму всех

строк массива. Исходную матрицу и полученные суммы вывести на экран.

**Ниже на рисунке представлен фрагмент работающего проекта:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4. Выполнение кода

**Код программы:**

format PE console

entry start

include 'win32a.inc'

section '.data' data readable writeable

text1 db 'Massiv (3,5)',13,10,0

text5 db 13,10,'obsh sum=%d',13,10,0

text3 db 13,10,'sum stroki=%d',10,13,0

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

fmat db '%d',9,0

l dd 4

v dd 3

p db 'pause',0

sum dd 0

osum dd 0

mas dd 1,2,3

dd 5,7,8

dd 1,5,5

dd 2,4,5

start:

cinvoke printf, text1

macro prntn 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 [osum],eax

push ecx

cinvoke printf, fmat, eax

pop ecx

add esi, 4

pop ecx

loop met1

cinvoke printf,text3, [sum]

mov[sum],0

pop ecx

loop met2

}

prntn mas, 4,3

mov AX, 0

mov EBX, 0

mov ECX,5

lea EBX, [mas]

CYCL1: push ECX

mov ECX,2

mov DX,[EBX]

mov ESI,20

CYCL2:cmp DX, [EBX+ESI]

jge NEXT

mov DX,[EBX+ESI]

NEXT: add ESI,20

loop CYCL2

add AX,DX

pop ECX

add EBX,4

loop CYCL1

cinvoke printf, text5,[ osum]

cinvoke sleep,9000

cinvoke sleep,0

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'

**Вывод:** мною была написана программа с использованием основных арифметических инструкций.

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: «Обработка одномерных массивов с использованием команд языка Assembler»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Альшакова Е.Л.

Дата: 06.12.2022

**Цель работы:** Разработать программу обработки одномерного массива.

**Задание:** Написать программу определения заданной характеристики

последовательности чисел C1, C2, ..., Cn. Произведение всех чисел.

**Ниже на рисунке представлен фрагмент работающего проекта:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5. Выполнение кода

**Код программы:**

format PE console

include 'win32a.inc'

entry start

proc proizvedenie

cycl:

imul eax,[ebx]

add ebx,4

loop cycl

ret

endp

start:

;section '.code' code readable executable

macro prntn op1,op2

{

local metka

mov esi,op1

mov ecx,[op2]

mov eax,0

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

}

cinvoke printf,text1

prntn a,N

lea ebx,[a]

mov ecx,10

mov eax, [ebx]

call proizvedenie

cinvoke printf,text2,eax

cinvoke sleep,9000

cinvoke sleep,0

text1 db 'Massiv: ',0

text2 db 10, 13, 'proizvedenie=%d' ,0

a dd 1,1,1,1,1,2,1,1,1,6

N dd 10

z dd 6

fmat db ' %d',0

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'

**Вывод:** Мною была разработана программа обработки одномерных массивов.

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: «Цикл»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Альшакова Е.Л.

Дата: 06.12.2022

**Цель работы:** Разработать программу: циклически сдвинуть на K позиций влево элементы массива A. Программу разработать на основе пользовательских процедур. Параметры передаются через стек.

**Ниже на рисунке представлен фрагмент работающего проекта:**

Изображение выглядит как текст, снимок экрана, компьютер, монитор

Автоматически созданное описание

Рисунок 6. Выполнение кода

**Код программы:**

format PE console

include 'win32a.inc'

start:

macro prntn op1,op2

{

local metka

mov esi,op1

mov ecx,op2

mov eax,0

metka:

mov al,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

inc esi

loop metka

}

cinvoke printf, text2

cinvoke scanf, spp, K

mov eax,0

lea ebx,[A]

mov [ebx], byte 11

mov [ebx+1],byte 12

mov [ebx+2],byte 13

mov [ebx+3],byte 14

mov [ebx+4],byte 15

mov [ebx+5],byte 16

mov [ebx+6],byte 17

prntn A,[N]

stdcall sdvig,A,[N],[K] ;процедура выполняет сдвиг влево на K

;разрядов

cinvoke printf,text

prntn A,[N]

invoke sleep,9000

invoke exit,0

proc sdvig op4,op5,op6

mov ebx, [op4]

mov edi, [op5]

dec edi

mov cx, word [op6]

cycl:

pusha

mov ecx,edi

cycl2:

mov al,[ebx]

xchg al,[ebx+1]

mov [ebx],al

inc ebx

loop cycl2

popa

loop cycl

ret

endp

A db 7 dup(?)

N dd 7

K dd ?

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

text2 db 'Enter K: ',0

fmat db '%d',9,0

spp db '%d',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

**Вывод:** Мною была разработана программа для циклического сдвига значения на основе пользовательских процедур.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

# **Практические работы по C#**

## ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1

**Тема: «Написание консольного приложения для**

**вывода числового массива»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Киреева Г.И.

Дата: 06.12.2022

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

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

• Зоны главной и побочной диагоналей массива

• Зона выше главной и выше побочной диагоналей (верхний треугольник)

• Зона ниже главной и ниже побочной диагоналей (нижний треугольник)

• Зона выше главной и ниже побочной диагоналей (правый треугольник)

• Зона ниже главной и выше побочной диагоналей (левый треугольник).

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

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

**Ниже на рисунке представлен фрагмент работающего проекта:**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 7. Вывод числового массива

**Код программы:**

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.BackgroundColor = ConsoleColor.Gray;

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Clear();

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

Random rand = new Random();

Console.ForegroundColor = ConsoleColor.DarkCyan;

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("Борушко Дарья, Ефимов Константин 2ПКС-420");

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

Console.CursorLeft = 50;

Console.CursorTop = 10;

Console.WriteLine(" Цветной статичный массив ");

Console.WriteLine("\n");

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

{

Console.CursorLeft = 50;

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

{

mass[k, m] = rand.Next(10, 100);

if (k == m || k + m == 9)

{

if (mass[k, m] > max1)

{

max1 = mass[k, m];

}

if (mass[k, m] < min1)

{

min1 = mass[k, m];

}

sum1 += mass[k, m];

Console.ForegroundColor = ConsoleColor.DarkRed;

}

else if (k + m < 9 && k < m)

{

if (mass[k, m] > max2)

{

max2 = mass[k, m];

}

if (mass[k, m] < min2)

{

min2 = mass[k, m];

}

sum2 += mass[k, m];

Console.ForegroundColor = ConsoleColor.Black;

}

else if (k + m < 9 && m < k)

{

if (mass[k, m] > max3)

{

max3 = mass[k, m];

}

if (mass[k, m] < min3)

{

min3 = mass[k, m];

}

sum3 += mass[k, m];

Console.ForegroundColor = ConsoleColor.Blue;

}

else if (k + m > 9 && m < k)

{

if (mass[k, m] > max4)

{

max4 = mass[k, m];

}

if (mass[k, m] < min4)

{

min4 = mass[k, m];

}

sum4 += mass[k, m];

Console.ForegroundColor = ConsoleColor.DarkGreen;

}

else if (k + m > 9 && k < m)

{

if (mass[k, m] > max5)

{

max5 = mass[k, m];

}

if (mass[k, m] < min5)

{

min5 = mass[k, m];

}

sum5 += mass[k, m];

Console.ForegroundColor = ConsoleColor.Magenta;

}

else

{

Console.ForegroundColor = ConsoleColor.DarkGray;

}

Console.Write(mass[k, m] + " ");

}

Console.WriteLine();

}

Console.ForegroundColor = ConsoleColor.DarkRed;

Console.WriteLine("\n\t\t\t\t\t\t Диагонали: ");

Console.WriteLine("\t\t\t\t\t\t Сумма элементов диагоналей: " + sum1);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент диагоналей: " + max1);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент диагоналей: " + min1);

Console.ForegroundColor = ConsoleColor.Black;

Console.WriteLine("\n\t\t\t\t\t\t Верхний треугольник: ");

Console.WriteLine("\t\t\t\t\t\t Сумма элементов верхнего треугольника: " + sum2);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент верхнего треугольника: " + max2);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент верхнего треугольника: " + min2);

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("\n\t\t\t\t\t\t Левый треугольник: ");

Console.WriteLine("\t\t\t\t\t\t Сумма элементов левого треугольника: " + sum3);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент левого треугольника: " + max3);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент левого треугольника: " + min3);

Console.ForegroundColor = ConsoleColor.DarkGreen;

Console.WriteLine("\n\t\t\t\t\t\t Нижний треугольник: ");

Console.WriteLine("\t\t\t\t\t\t Сумма элементов нижнего треуголника: " + sum4);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент нижнего треугольника: " + max4);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент верхнего треугольника: " + min4);

Console.ForegroundColor = ConsoleColor.Magenta;

Console.WriteLine("\n\t\t\t\t\t\t Правый треугольник: ");

Console.WriteLine("\t\t\t\t\t\t Сумма элементов правого треуголника: " + sum5);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент правого треугольника: " + max5);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент правого треугольника: " + min5);

Console.ForegroundColor = ConsoleColor.Black;

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

}

}

}

**Вывод:** Мною было разработано консольное приложение для выводы числового массива.

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 функции»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Киреева Г.И.

Дата: 06.12.2022

**Цель работы:** Разработать приложение с использованием компонента DataGridView и многострочным textBox.

**Задание:** Создать форму вычислить функцию, под ней поле с функцией

Значение «а» вводится пользователем, при нажатии на кнопку вычислить - вычислить

сетка грид под этим, в которой столбцы x | y со значениями

под кнопкой вычислить кнопка записать, которая записывает в файл всю таблицу грид

под кнопкой записать кнопка считать, которая считывает из файла

кнопка выхода.

**Ниже на рисунке представлен фрагмент работающего проекта:**

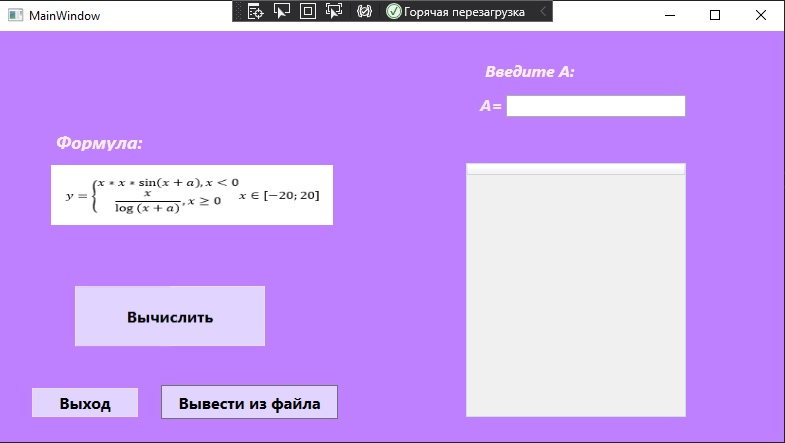


Рисунок 8. Вычисление функции

**Код программы:**

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 УП2

{

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("C:/Users/User/Downloads/УП2/Криничанская С.К. Борушко Д.М. УП2 2ПКС-420/УП2/Решения/решение.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("C:/Users/User/Downloads/УП2/Криничанская С.К. Борушко Д.М. УП2 2ПКС-420/УП2/Решения/решение.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;

}

}

}

**XAML:**

<Window x:Class="УП2.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:УП2"

mc:Ignorable="d"

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

Top="300"

Left="500">

<Grid Background="#FFFBF1F1">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="569\*"/>

<ColumnDefinition Width="20\*"/>

<ColumnDefinition Width="7\*"/>

<ColumnDefinition Width="4\*"/>

</Grid.ColumnDefinitions>

<TextBlock Name="label" HorizontalAlignment="Left" Margin="35,10,0,0" TextWrapping="Wrap" Text="Вычислить функцию" VerticalAlignment="Top" Height="23" Width="182" FontSize="18" RenderTransformOrigin="0.569,0.316" FontFamily="Corbel Light" FontWeight="Normal"/>

<TextBox Name="input" HorizontalAlignment="Left" Height="45" Margin="85,119,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="80" FontSize="18"/>

<TextBlock HorizontalAlignment="Left" Margin="55,121,0,0" TextWrapping="Wrap" Text="a=" VerticalAlignment="Top" Height="23" Width="30" FontSize="18" RenderTransformOrigin="0.544,1.369"/>

<Button IsDefault="true" Name="Calculate" Content="Вычислить" HorizontalAlignment="Left" Margin="263,126,0,0" VerticalAlignment="Top" Width="168" Height="30" FontSize="22" Click="Calculate\_Click" Background="#FF6EB885" Foreground="White" FontFamily="Corbel Light" RenderTransformOrigin="0.518,-0.254"/>

<Button Name="Write" Content="Записать в файл" HorizontalAlignment="Left" Margin="571,310,0,0" VerticalAlignment="Top" Width="156" Height="34" FontSize="18" Click="Write\_Click" RenderTransformOrigin="0.391,0.626" Background="#FF6EB885" FontFamily="Corbel Light" Foreground="White" />

<Button Name="Read" Content="Загрузить с файла" HorizontalAlignment="Left" Margin="571,252,0,0" VerticalAlignment="Top" Width="156" Height="34" FontSize="18" Click="Read\_Click" Foreground="White" Background="#FF6EB885" FontFamily="Corbel Light" RenderTransformOrigin="0.509,-0.129" />

<Button Name="Exit" Content="Выход" HorizontalAlignment="Left" Margin="571,372,0,0" VerticalAlignment="Top" Width="156" Height="32" FontSize="18" Click="Exit\_Click" Background="#FF9E0000" Foreground="White" />

<DataGrid Name="Data" HorizontalAlignment="Left" Height="235" Margin="55,169,0,0" VerticalAlignment="Top" Width="311"/>

<Button Name="ClearGridData" Content="Очистить таблицу" HorizontalAlignment="Left" Margin="453,126,0,0" VerticalAlignment="Top" Width="180" Height="30" FontSize="18" Click="ClearGridData\_Click" FontFamily="Corbel Light" Background="#FFECAF9E" Foreground="White" />

<Image Margin="35,39,359,320" Stretch="Fill" Source="Снимок экрана 2022-05-30 206945.png"/>

</Grid>

</Window>

**Вывод:** Мною было разработано приложение с использованием компонента DataGridView и многострочным textBox.

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: «Приложение с использованием ListBox»**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Киреева Г.И.

Дата: 06.12.2022

**Цель работы:** Разработать приложение с использованием компонента ListBox.

**Задание:** Создать программу с использованием листбокса. //wpf: работа с листбоксом

Добавить окно ввода продуктов, обработать исключения(пробелы и тд).Добавить кнопки:

• Добавить, она добавляет в листбокс введённое предложение

• Изменить, выбрать страну в листбоксе, при нажатии на кнопку открывается изменение этого слова с мигающим курсором

• Удалить, также нужно выбрать и нажать кнопку, появится окно подтверждения с да/нет

• Очистить список, то же самое, что и удаление, но предупреждение с восклицательным знаком

• Сортировать, сортирует список по алфавиту.

• Счетчик записей под окном ввода;

• Выход, закрывает консоль.

**Ниже на рисунке представлен фрагмент работающего проекта:**

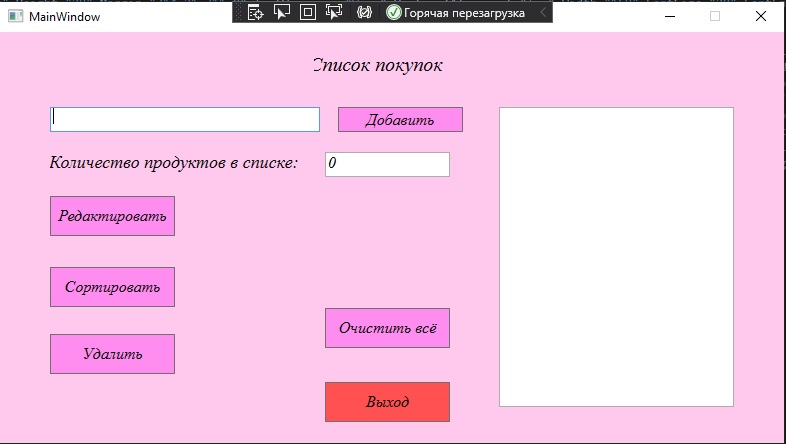


Рисунок 9. Выполнение кода

**Код программы:**

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 ПР4

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

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("Вы ввели пустую строку!");

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("Вы желаете удалить продукт 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 продукта.");

quantity\_products = 0;

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

}

input.Focus();

}

input.Focus();

}

else if (quantity\_products >= 0)

{

MessageBox.Show("Для удаления продукта из списка необходимо выделить его мышкой в окне списка.");

input.Focus();

}

else

{

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

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 желаете очистить весь список продуктов?", "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();

}

}

}

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("Для редактирования списка продуктов необходимо выделить его мышкой в окне списка.");

input.Focus();

}

}

}

}

**XAML:**

<Window x:Class="ПР4.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:ПР4"

mc:Ignorable="d"

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

<Grid Background="#FFFFC8ED">

<TextBlock HorizontalAlignment="Center" Height="28" Margin="314,21,311,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="169" FontSize="20" FontFamily="Times New Roman" FontStyle="Italic" RenderTransformOrigin="0.846,0.59"><Run

Text="Список"/><Run Text=" покупок"/></TextBlock>

<ListBox Name ="list" HorizontalAlignment="Right" Height="300" Margin="0,75,50,0" VerticalAlignment="Top" Width="235"/>

<TextBox Name ="input" Height="25" Margin="50,75,0,0" TextWrapping="Wrap" Width="270" HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBox Name ="quantity" Height="25" Margin="325,120,0,0" TextWrapping="Wrap" Width="125" HorizontalAlignment="Left" VerticalAlignment="Top" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" TextDecorations="{x:Null}"/>

<Button Name ="buttton\_add" Click ="click\_add" Content="Добавить" HorizontalAlignment="Left" Height="25" Margin="338,75,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" Background="#FFFF8DEF"/>

<Button Name ="button\_delete" Click ="click\_delete" Content="Удалить" HorizontalAlignment="Left" Height="40" Margin="50,302,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" Background="#FFFF8DEF"/>

<Button Name ="button\_edit" Click ="click\_edit" Content="Редактировать" HorizontalAlignment="Left" Height="40" Margin="50,164,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" Background="#FFFF8DEF"/>

<Button Name ="button\_sort" Click ="click\_sort" Content="Сортировать" HorizontalAlignment="Left" Height="40" Margin="50,235,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" Background="#FFFF8DEF"/>

<Button Name ="button\_clear" Click ="click\_clear" Content="Очистить всё" HorizontalAlignment="Left" Height="40" Margin="325,276,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" Background="#FFFF8DEF"/>

<Button Name ="button\_exit" Click ="click\_exit" Content="Выход" HorizontalAlignment="Left" Height="40" Margin="325,350,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" Background="#FFFF5151"/>

<TextBlock HorizontalAlignment="Center" Height="25" Margin="50,120,474.667,0" TextWrapping="Wrap" Text=" Количество продуктов в списке: " VerticalAlignment="Top" Width="270" FontSize="18" FontFamily="Times New Roman" FontStyle="Italic"/>

</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 № 4

**Тема: “События C#”**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата:29.11.2022

Москва 2022

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

**Ниже на рисунке представлен фрагмент работающего проекта:**

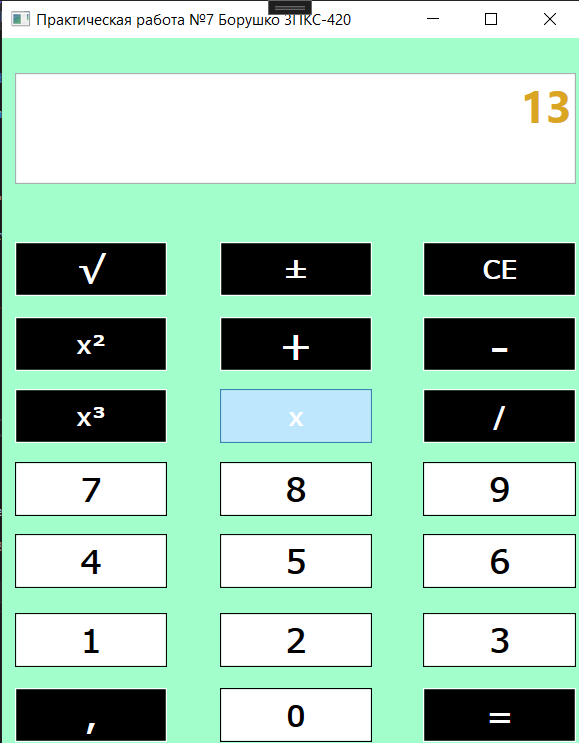
****

Рисунок 10. Калькулятор

**Код программы:**

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 Borushko\_calculator\_420

{

public partial class MainWindow : Window

{

private string key = "";

private string[] nums = { "", "" };

private string number = " ";

public MainWindow()

{

InitializeComponent();

calc.Foreground = Brushes.Goldenrod;

}

public Dictionary<string, Func<double, double, double>> bin\_operator = new Dictionary<string, Func<double, double, double>>

{

{ "+", (x, y) => x + y },

{ "-", (x, y) => x - y },

{ "x", (x, y) => x \* y },

{ "/", (x, y) => x / y },

};

public Dictionary<string, Func<double, double>> unar\_operator = new Dictionary<string, Func<double, double>>

{

{ "^2", x => x \* x },

{ "^3", x => x \* x \* x },

{ "±", x => -x },

{ "√", x => Math.Sqrt(x) },

};

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

{

if (calc.Text == "Failed")

{

calc.Text = "";

for (int i = 0; i < nums.Length; i++)

{

nums[i] = "";

}

number = "";

}

string button\_content = ((Button)e.OriginalSource).Content.ToString();

string old\_key = key;

if (isNum(button\_content))

{

calc.Text += button\_content;

number += button\_content;

}

else

{

if (button\_content == "=" || button\_content == "СE") { }

else

{

key = button\_content;

if (isNum(nums[0]) && key == "√")

{

calc.Text = "sqrt ";

calc.Text += nums[0];

key = "sqrt";

number = calc.Text.Replace("sqrt ", "\_");

}

else

{

if (key == "х²")

{

calc.Text += "^2";

number = calc.Text.Replace("^2", " ");

}

else if (key == "x³")

{

calc.Text += "^3";

number = calc.Text.Replace("^3", "\_");

}

else if (key == "±")

{

calc.Text += "±";

number = calc.Text.Replace("±", "\_");

}

else if (key == "√" || key == "sqrt")

{

calc.Text = "sqrt ";

key = "sqrt";

number = calc.Text.Replace("sqrt ", "\_");

}

else

{

if ((button\_content == "-" && calc.Text.Length == 0) || button\_content == ",")

{

calc.Text += button\_content;

number += button\_content;

}

else if (button\_content == "-" && number.Contains("\_"))

{

calc.Text += button\_content;

number += button\_content;

if (old\_key == "sqrt")

key = old\_key;

}

else

{

calc.Text += button\_content;

number += "\_";

}

}

}

}

}

if (button\_content == "СE")

{

calc.Text = "";

for (int i = 0; i < nums.Length; i++)

nums[i] = "";

number = "";

}

else if (button\_content == "=")

{

nums = number.Split('\_');

nums = nums.Where(value => value != "").ToArray();

string operation = "";

if (nums.Length == 1)

{

for (int i = 0; i < calc.Text.Length; i++)

{

if (calc.Text[i] == '^' && (calc.Text[i + 1] == '2' || calc.Text[i + 1] == '3'))

{

operation = calc.Text.Substring(i, 2);

break;

}

else if (key == "sqrt")

{

operation = "√";

break;

}

else if (calc.Text[i] == '!')

{

operation = calc.Text[i].ToString();

break;

}

else if (key == "±")

{

operation = "±";

break;

}

}

foreach (KeyValuePair<string, Func<double, double>> unar\_operate in unar\_operator)

{

if (unar\_operate.Key == operation)

{

if (operation == "√")

{

if (double.Parse(nums[0]) < 0)

calc.Text = "Failed";

else

calc.Text = unar\_operate.Value(double.Parse(nums[0])).ToString();

}

else

calc.Text = unar\_operate.Value(double.Parse(nums[0])).ToString();

break;

}

}

}

else

{

nums = nums.Where(value => value != "").ToArray();

for (int i = 0; i < calc.Text.Length; i++)

{

if ((calc.Text[i] == '+' || calc.Text[i] == '-' || calc.Text[i] == 'x' || calc.Text[i] == '/' || calc.Text[i] == '%') && i > 0)

{

operation = calc.Text[i].ToString();

break;

}

}

foreach (KeyValuePair<string, Func<double, double, double>> bin\_operate in bin\_operator)

{

if (bin\_operate.Key == operation)

{

if (operation == "/")

{

if (nums[1] == "0")

calc.Text = "Failed";

else

calc.Text = bin\_operate.Value(double.Parse(nums[0]), double.Parse(nums[1])).ToString();

}

else

{

calc.Text = bin\_operate.Value(double.Parse(nums[0]), double.Parse(nums[1])).ToString();

}

break;

}

}

}

nums[0] = calc.Text;

number = calc.Text;

}

}

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

{

MessageBoxResult result = MessageBox.Show("Вы точно хотите выйти?", "Выход", MessageBoxButton.OKCancel);

if (result == MessageBoxResult.Cancel)

e.Cancel = true;

base.OnClosing(e);

}

private static bool isNum(string str)

{

double num = 0;

return double.TryParse(str, out num);

}

}

}

**XAML:**

<Window x:Class="Borushko\_calculator\_420.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:Borushko\_calculator\_420"

mc:Ignorable="d"

Title="Практическая работа №7 Борушко 3ПКС-420" Height="606.366" Width="476.525">

<Grid Height="573" VerticalAlignment="Top" Background="#FFA2FFCC">

<TextBox x:Name="calc" HorizontalAlignment="Left" Height="89" Margin="10,28,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="449" FontSize="36" FontWeight="Bold" TextAlignment="Right"/>

<Button Content="√" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,163,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="±" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,163,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="24">

</Button>

<Button Content="СE" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,163,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

</Button>

<Button Content="х²" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,223,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

</Button>

<Button Content="x³" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,281,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

</Button>

<Button Content="7" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,339,0,0" VerticalAlignment="Top" Width="122" BorderBrush="Black" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="4" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,397,0,0" VerticalAlignment="Top" Width="122" BorderBrush="#FF0B0A0A" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="1" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,460,0,0" VerticalAlignment="Top" Width="122" BorderBrush="#FF0B0A0A" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="," HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,520,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="+" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,223,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="32">

</Button>

<Button Content="x" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,281,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

</Button>

<Button Content="8" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,339,0,0" VerticalAlignment="Top" Width="122" BorderBrush="Black" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="5" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,397,0,0" VerticalAlignment="Top" Width="122" BorderBrush="#FF0B0A0A" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="2" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,460,0,0" VerticalAlignment="Top" Width="122" BorderBrush="Black" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="0" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,520,0,0" VerticalAlignment="Top" Width="122" BorderBrush="Black" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="24">

</Button>

<Button Content="-" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,223,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="36">

</Button>

<Button Content="/" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,281,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

</Button>

<Button Content="9" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,339,0,0" VerticalAlignment="Top" Width="122" BorderBrush="Black" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26" >

</Button>

<Button Content="6" Click="Button\_Click" Height="43" Margin="337,397,0,0" VerticalAlignment="Top" BorderBrush="Black" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26" RenderTransformOrigin="0.492,0.521" HorizontalAlignment="Left" Width="122"/>

<Button Content="3" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,460,0,0" VerticalAlignment="Top" Width="122" BorderBrush="#FF0B0A0A" Background="White" Foreground="Black" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

</Button>

<Button Content="=" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,520,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="Black" Foreground="White" FontFamily="MS Reference Sans Serif" FontSize="26" FontWeight="Bold">

</Button>

</Grid>

</Window>

**Вывод:** мною была разработана программа - калькулятор на языке C#, используя один обработчик для нескольких событий.

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: “Проверка орфографии”**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата: 01.12.2022

Москва 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:**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 11. Форма проекта

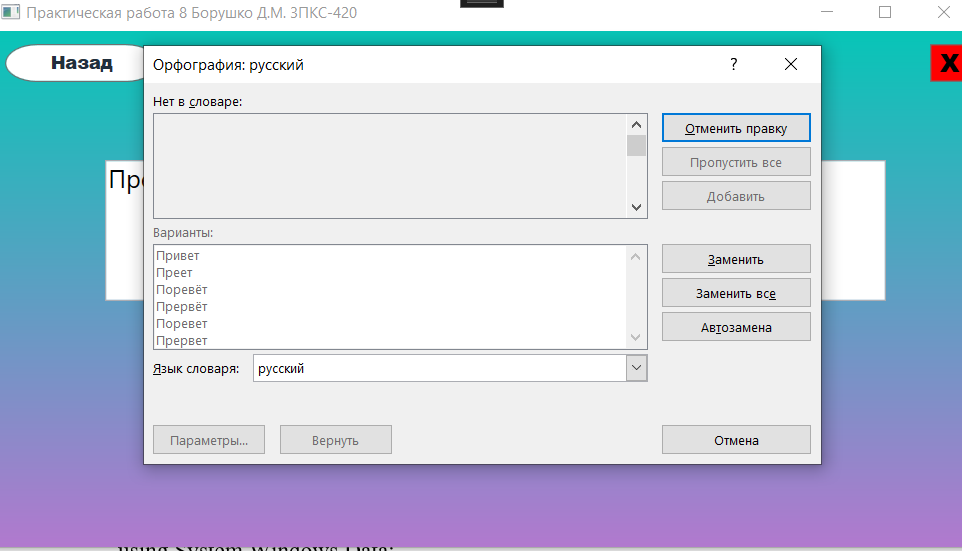
****

Рисунок 12. Проверка орфографии

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 13. Замена слова

**Код программы:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

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.Shapes;

namespace Orfografia

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

textBox.Clear();

textBox.TabIndex = 0;

button1.TabIndex = 1;

}

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

{

this.Hide();

Window MainWindow = new MainWindow();

MainWindow.Show();

}

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

{

this.Close();

}

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

{

var Word1 = new Microsoft.Office.Interop.Word.Application();

Word1.Visible = false;

var Doc = Word1.Documents.Add();

Doc.Words.First.InsertBefore(textBox.Text);

Doc.CheckSpelling();

var NewText = Doc.Content.Text;

textBox.Text = NewText;

Word1.Documents.Close(Microsoft.Office.Interop.Word.WdSaveOptions.wdDoNotSaveChanges);

Word1.Quit();

Word1 = null;

}

}

}

**XAML:**

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

mc:Ignorable="d"

Title="Практическая работа 8 Борушко Д.М. 3ПКС-420" Height="450" Width="800" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.Background>

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

<GradientStop Color="#FF07C5B7" Offset="0"/>

<GradientStop Color="#FFB179CE" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Button Name ="button1" Content="Проверка орфографии" HorizontalAlignment="Center" Height="48" Margin="10,253,0,0" Click="button1\_Click" VerticalAlignment="Top" Width="315" Foreground="#FFF4F4F4" Background="#FF121212" FontSize="23" FontStyle="Italic" FontFamily="Arial Black">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="25"/>

</Style>

</Button.Resources>

</Button>

<Button Name ="button2" Content="Назад" Foreground="#212F3C" Background="White" HorizontalAlignment="Left" Height="31" Margin="10,10,0,0" VerticalAlignment="Top" Width="123" FontFamily="Arial Black" FontSize="14">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="25"/>

</Style>

</Button.Resources>

</Button>

<Button Name ="button3" Content="X" HorizontalAlignment="Left" Height="31" Margin="750,10,0,0" VerticalAlignment="Top" Width="32" Background="Red" Foreground="Black" FontSize="20" FontFamily="Arial Black"/>

<TextBox Name ="textBox" HorizontalAlignment="Left" Height="113" TextWrapping="Wrap" VerticalAlignment="Top" Width="625" Margin="90,103,0,0" FontSize="20" />

</Grid>

</Window>

**Вывод:** Мною была разработана программа, которая предлагает пользователю ввести какие-либо слова, предложения в текстовое поле и после нажатия соответствующей кнопки проверить орфографию введенного текста. Для непосредственной проверки орфографии была использована функция Checkspelling объектной библиотеки MS Word.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

## ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 6

**Тема: “Создание таблицы”**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата: 04.12.2022

Москва 2022

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

**Ниже на рисунке представлен фрагмент работающего проекта:**

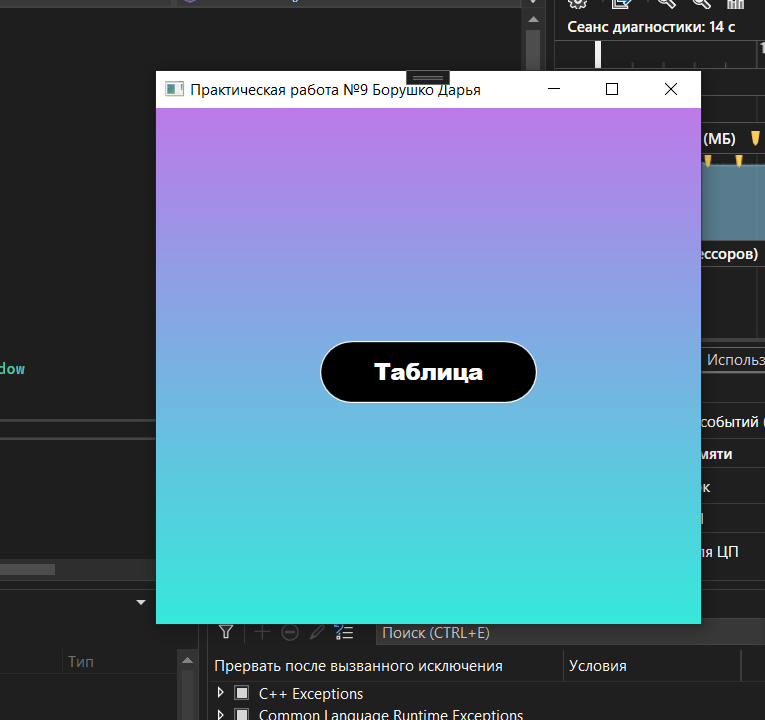
****

Рисунок 14. Форма проекта

**Изображение выглядит как стол

Автоматически созданное описание**

Рисунок 15. Пример таблицы

**Код программы:**

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.Shapes;

namespace WpfApp2

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

String[] Name = { "Мой номер", "Руководитель", "Полиция", "Зав-отделения", "Фёдор Александрович", "Мама-рабочий", "Ольга Николаевна", "Скорая", "Колледж" };

String[] Phone = { "+79034567878", "+79657658899", "102", "+7 (499) 553-14-28", "516-99-00 доп 26-85", "+79265482323", "89234370812", "103", "+7 (925) 613-09-67" };

var Word1 = new Microsoft.Office.Interop.Word.Application();

Word1.Visible = true;

var Doc = Word1.Documents.Add();

Word1.Selection.TypeText("ТАБЛИЦА ТЕЛЕФОНОВ");

var t1 = Microsoft.Office.Interop.Word.WdDefaultTableBehavior.wdWord9TableBehavior;

var t2 = Microsoft.Office.Interop.Word.WdAutoFitBehavior.wdAutoFitContent;

Word1.ActiveDocument.Tables.Add(Word1.Selection.Range, 9, 2, t1, t2);

for (int i = 1; i <= 9; i++)

{

Word1.ActiveDocument.Tables[1].Cell(i, 1).Range.InsertAfter(Name[i - 1]);

Word1.ActiveDocument.Tables[1].Cell(i, 2).Range.InsertAfter(Phone[i - 1]);

}

var t3 = Microsoft.Office.Interop.Word.WdUnits.wdLine;

var stroka9 = 9;

Word1.Selection.MoveDown(t3, stroka9, null);

Word1.Selection.TypeText("Основной список номеров");

var maz = @"C:\Users\user\OneDrive\Документы\Работы за 3 курс\УП 01.01\ПР№9 Борушко Д.М. 3ПКС-420\WpfApp2\WpfApp2";

Word1.ActiveDocument.SaveAs(maz);

}

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

{

MessageBoxResult result = MessageBox.Show("Вы точно хотите выйти?", "Выход", MessageBoxButton.OKCancel);

if (result == MessageBoxResult.Cancel)

e.Cancel = true;

base.OnClosing(e);

}

}

}

**XAML:**

<Window x:Class="WpfApp2.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2"

mc:Ignorable="d"

Title="Практическая работа №9 Борушко Дарья" Height="450" Width="450" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.Background>

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

<GradientStop Color="#FFBC7AE9" Offset="0"/>

<GradientStop Color="#FF36E8DB" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Button Content="Таблица" Foreground="White" Background="Black" HorizontalAlignment="Center" Height="50" VerticalAlignment="Top" Width="174" Click="Button\_Word" FontFamily="Arial Black" FontWeight="Bold" FontSize="18" BorderBrush="White" Margin="0,186,0,0">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="25"/>

</Style>

</Button.Resources>

</Button>

</Grid>

</Window>

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

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

## ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 7

**Тема: “Функции Excel”**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата: 04.12.2022

Москва 2022

**Цель работы**: создать собственный проект с использованием функций MS Excel не используя функцию нахождения числа Pi.

**Ниже на рисунке представлен фрагмент работающего проекта:**

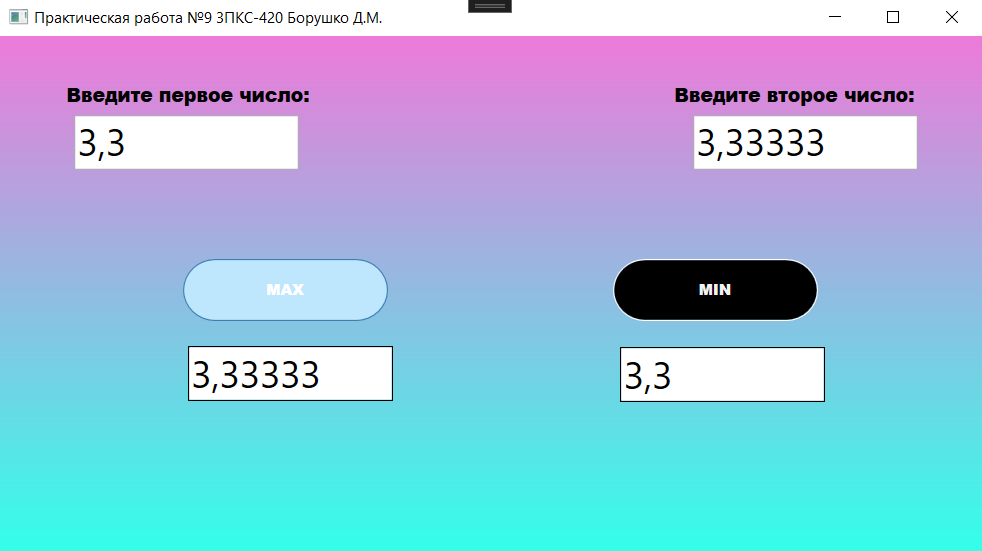
****

Рисунок 16. Нахождение максимального и минимального числа

**Код программы:**

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.Shapes;

namespace function

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

double x = Convert.ToDouble(Text\_A1.Text);

double y = Convert.ToDouble(Text\_A2.Text); ;

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

var MAX = XL.WorksheetFunction.Max(x, y);

Text\_MAX.Text = Convert.ToString(MAX);

XL.Quit();

}

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

{

double x = Convert.ToDouble(Text\_A1.Text);

double y = Convert.ToDouble(Text\_A2.Text); ;

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

var MIN = XL.WorksheetFunction.Min(x, y);

Text\_MIN.Text = Convert.ToString(MIN);

XL.Quit();

}

}

}

**XAML:**

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

mc:Ignorable="d"

Title="Практическая работа №9 3ПКС-420 Борушко Д.М." Height="450" Width="800" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.Background>

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

<GradientStop Color="#FFED7AD9"/>

<GradientStop Color="#FF33FFEA" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Button Name ="MAX" Content="MAX" Foreground="White" Background="Black" HorizontalAlignment="Left" Height="50" Margin="146,178,0,0" VerticalAlignment="Top" Width="164" Click="Button\_MAX" FontFamily="Arial Black" BorderBrush="White">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="25"/>

</Style>

</Button.Resources>

</Button>

<TextBox Name="Text\_A1" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="554,63,0,0" Height="44" Width="180" Foreground="Black" FontSize="30"/>

<TextBox Name="Text\_A2" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="59,63,0,0" Height="44" Width="180" Foreground="Black" FontSize="30"/>

<TextBox Name="Text\_MAX" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="150,248,0,0" Height="44" Width="164" Foreground="Black" FontSize="30" BorderBrush="Black"/>

<Button x:Name ="MIN" Content="MIN" Foreground="#FFF3F3F3" Background="Black" HorizontalAlignment="Left" Height="50" Margin="490,178,0,0" VerticalAlignment="Top" Width="164" Click="Button\_MIN" FontFamily="Arial Black" BorderBrush="White">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="25"/>

</Style>

</Button.Resources>

</Button>

<TextBox x:Name="Text\_MIN" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="496,249,0,0" Height="44" Width="164" Foreground="Black" FontSize="30" BorderBrush="Black"/>

<TextBlock HorizontalAlignment="Left" Height="36" Margin="53,39,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="216" FontSize="15" FontFamily="Arial Black" Foreground="Black"><Run Text="Введите первое "/><Run Text="число"/><Run Text=":"/></TextBlock>

<TextBlock HorizontalAlignment="Center" Height="24" Margin="515,39,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="222" FontSize="15" FontFamily="Arial Black" Foreground="Black"><Run Text="Введите "/><Run Text="второе"/><Run Text=" "/><Run Text="число"/><Run Text=":"/></TextBlock>

</Grid>

</Window>

**Вывод:** мною был разработан собственный проект с использованием функций MS Excel, для нахождения максимального и минимального числа из двух введенных.

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: “Работа с файлами в различных кодировках”**

Студент: Борушко Д.М.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата: 04.12.2022

Москва 2022

**Цель работы**: 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 рисунке представлен фрагмент работающего проекта:**

****

Рисунок 17. Исходный документ

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 18. Выбор метода сохранения

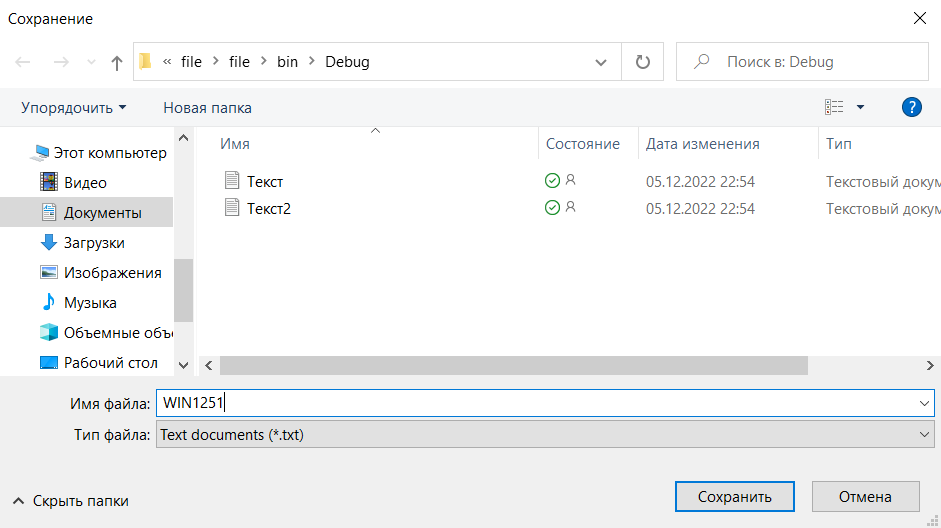
****

Рисунок 19. Сохранение

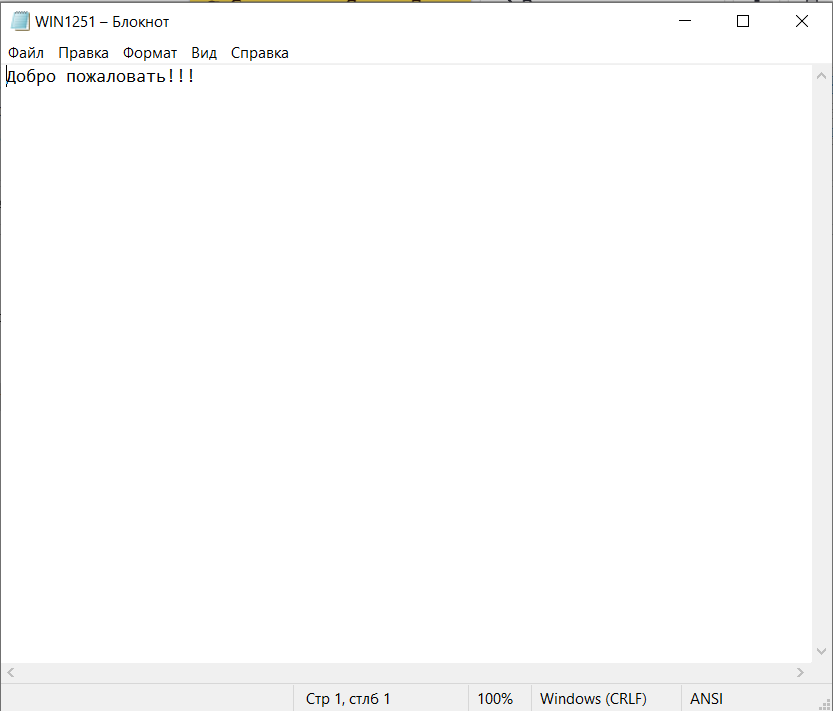
****

Рисунок 20. Сохраненный документ

**Код программы:**

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.Printing;

namespace file

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private System.Drawing.Printing.PrintDocument PrintDocument = new System.Drawing.Printing.PrintDocument();

private Microsoft.Win32.SaveFileDialog save = new Microsoft.Win32.SaveFileDialog();

private Microsoft.Win32.OpenFileDialog open = new Microsoft.Win32.OpenFileDialog();

public MainWindow()

{

InitializeComponent();

}

private void uni\_open(object sender, RoutedEventArgs e)

{

save.Filter = "Text documents (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

if (save.ShowDialog() == true)

{

StreamWriter writer = new StreamWriter(save.OpenFile(), Encoding.GetEncoding("UTF-8"));

writer.Write(input.Text);

writer.Close();

}

}

private void win\_open(object sender, RoutedEventArgs e)

{

save.Filter = "Text documents (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

if (save.ShowDialog() == true)

{

string filename = save.FileName;

StreamReader streamReader = new StreamReader(filename, Encoding.GetEncoding("windows-1251"));

string text = streamReader.ReadToEnd();

input.Text = text;

streamReader.Close();

streamReader.Dispose();

}

}

private void rtf\_open(object sender, RoutedEventArgs e)

{

open.Filter = "RTF (\*.RTF)|\*.RTF";

if (open.ShowDialog() == true)

{

string filename = open.FileName;

StreamReader streamReader = new StreamReader(filename, Encoding.GetEncoding("UTF-8"));

string text = streamReader.ReadToEnd();

input.Text = text;

streamReader.Close();

streamReader.Dispose();

}

}

private void save\_bin(object sender, RoutedEventArgs e)

{

save.Filter = "All files (\*.\*)|\*.\*";

if (save.ShowDialog() == true)

{

BinaryWriter writer = new BinaryWriter(save.OpenFile(), System.Text.Encoding.Default);

writer.Write(input.Text);

writer.Close();

}

}

private void save\_uni(object sender, RoutedEventArgs e)

{

save.Filter = "Text documents (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

if (save.ShowDialog() == true)

{

StreamWriter writer = new StreamWriter(save.OpenFile(), Encoding.GetEncoding("UTF-8"));

writer.Write(input.Text);

writer.Close();

}

}

private void save\_win(object sender, RoutedEventArgs e)

{

save.Filter = "Text documents (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

if (save.ShowDialog() == true)

{

StreamWriter writer = new StreamWriter(save.OpenFile(), Encoding.GetEncoding("Windows-1251"));

writer.Write(input.Text);

writer.Close();

}

}

private void save\_rtf(object sender, RoutedEventArgs e)

{

save.Filter = "RTF (\*.RTF)|\*.RTF";

if (save.ShowDialog() == true)

{

StreamWriter writer = new StreamWriter(save.OpenFile(), System.Text.Encoding.Default);

writer.Write(input.Text);

writer.Close();

}

}

}

}

**XAML:**

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

mc:Ignorable="d"

Title="Практическая работа №11 Борушко Д.М. 3ПКС-420" MinHeight="450" MinWidth="750" Width="750.573" Height="350.049" MaxWidth="750" MaxHeight="350">

<Grid Margin="0,0,-9,0" Background="#FF5DEDFF">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="337\*"/>

<ColumnDefinition Width="33\*"/>

<ColumnDefinition Width="89\*"/>

<ColumnDefinition Width="300\*"/>

</Grid.ColumnDefinitions>

<Menu Height="25" VerticalAlignment="Top" Foreground="White" Grid.ColumnSpan="4" Background="Black">

<MenuItem Header="Файл" Height="25" Width="67" FontSize="18" FontFamily="Arial Black" Foreground="White" BorderBrush="White" Background="Black">

<MenuItem Header="Открыть в формате Unicode" Click="uni\_open" Foreground="Black" Background="#00000000" />

<MenuItem Header="Открыть в формате Win1251 " Click="win\_open" Foreground="Black" Background="#00000000" />

<MenuItem Header="Открыть в формате RTF " Foreground="Black" Click="rtf\_open" />

<Separator />

<MenuItem Header="Сохранить как бинарный файл " Foreground="Black" Click="save\_bin" />

<MenuItem Header="Сохранить как... " Background="#00000000" Foreground="Black" >

<MenuItem Header="В формате Unicode" Click="save\_uni" ></MenuItem>

<Separator />

<MenuItem Header="В формате Win1251" Click="save\_win" ></MenuItem>

<Separator />

<MenuItem Header="В формате RTF" Click="save\_rtf" ></MenuItem>

</MenuItem>

<Separator />

<MenuItem Header="Распечатать" Foreground="Black"/>

</MenuItem>

</Menu>

<TextBlock HorizontalAlignment="Left" Height="35" Margin="10,45,0,0" TextWrapping="Wrap" Text="Введите текст:" VerticalAlignment="Top" Width="252" Foreground="Black" FontFamily="Arial Black" FontSize="22"/>

<TextBox Name="input" HorizontalAlignment="Left" Height="319" Margin="10,85,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="714" FontFamily="MS Reference Sans Serif" FontSize="18" Background="White" BorderBrush="Black" Grid.ColumnSpan="4"/>

</Grid>

</Window>

**Вывод:** мною был разработан простой текстовый редактор на языке C#.