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:

обучающаяся учебной группы № 420

С.К. Криничанская

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

И. В. Сибирев

*(И.О. Фамилия)*

Е. Л. Альшакова

*(И.О. Фамилия)*

**Москва**

**20­­­­22**

Содержание

[Перечень заданий/работ, выполненных в ходе учебной практики 3](#_Toc121267525)

[Практическая работа №1 8](#_Toc121267526)

[Практическая работа №2 10](#_Toc121267527)

[Практическая работа №3 12](#_Toc121267528)

[Практическая работа №4 14](#_Toc121267529)

[Практическая работа №5 17](#_Toc121267530)

[Практическая работа №6 20](#_Toc121267531)

[Практическая работа №7 24](#_Toc121267532)

[Практическая работа №8 28](#_Toc121267533)

[Практическая работа №9 33](#_Toc121267534)

[Практическая работа №10 35](#_Toc121267535)

[Практическая работа №11 39](#_Toc121267536)

[Практическая работа №12 43](#_Toc121267537)

[Практическая работа №13 48](#_Toc121267538)

# **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. |

# **Практическая работа №1**

Задание:

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

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'

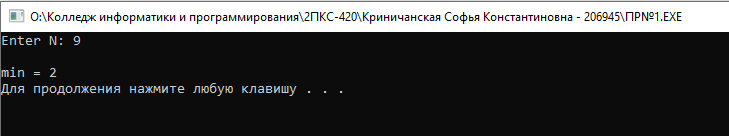


Рисунок 1. Выводимый результат

# **Практическая работа №2**

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.

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]

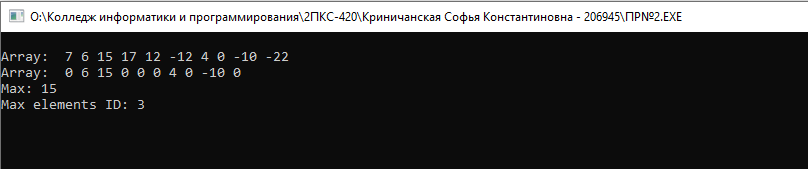


Рисунок 2. Выводимый результат

# **Практическая работа №3**

Задание:

Разработать рекурсивную процедуру вычисления суммы чисел от 1 до 100.

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

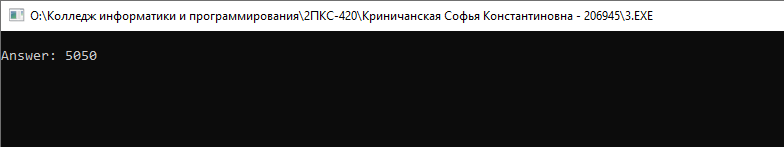


Рисунок 2. Выводимый результат

# **Практическая работа №4**

Задание:

Разместить в памяти компьютера матрицу 5\*4 целых  
без знаковых чисел. Написать программу, позволяющую найти сумму всех  
строк массива. Исходную матрицу и полученные суммы вывести на экран.

Код программы:

format PE console

include 'win32a.inc'

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

}

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

}

prnt2d X,[M],[N]

lea ebx,[Y]

lea esi,[X]

mov ecx,[M]

met2: push ecx

mov ecx,[N]

xor eax,eax

met1: add eax,[esi]

add esi,4

loop met1

mov [ebx],eax

add ebx,4

pop ecx

loop met2

xor eax,eax

lea ebx,[Y]

mov ecx,[M]

met3:

add eax,[ebx]

add ebx,4

loop met3

cinvoke printf, spp,eax

cinvoke printf, text4

prnt2d Y,1,[M]

invoke sleep,60000

invoke exit,0

X dd 1, 2, 3, 2, 1

dd 0, 1, 18, 15, 10

dd 1, 2, 3, 2, 1

dd 1, 1, 18, 10, 10

Y dd 100 dup(?)

N dd 5

M dd 4

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

text2 db 'Enter N: ',0

text3 db '',10,13,0

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

fmat db '%d',9,0

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

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

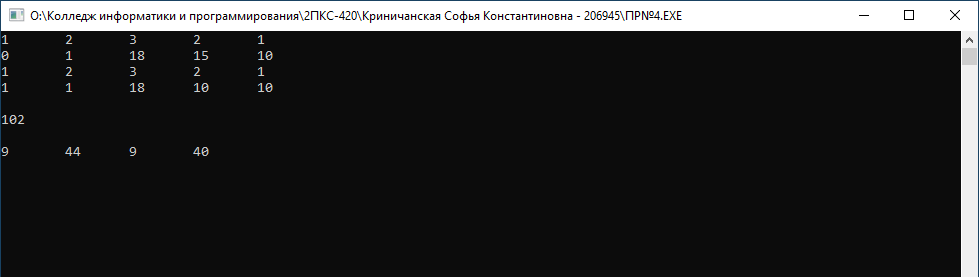


Рисунок 4. Выводимый результат

# **Практическая работа №5**

Задание:

Отношение сумм четных и нечетных элементов массивов.

Код программы:

format PE console

include 'win32a.inc'

section '.data' data readable writeable

text1 db 'Enter N: ',0

text3 db 10,13,'Summa1 = %d',10, 13, 0

text4 db 10,13,'Summa2 = %d',10, 13, 0

text6 db 10,13,'Answer = %d',10, 13, 0

text5 db '', 10, 13, 0

p db 'pause',0

fpp db ' %d',0

fmat db '%d',9,0

X dd 10, -3, 6, -6, -7, 4, 3,-23,5,7,-78,56,-4,3,-2,34,12

N dd ?

ne dd ?

c dd ?

;section '.code' code readable executable

entry start

proc sum

cycl:

add eax, [ebx]

sub ecx, 1

add ebx, 8

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

prntn X,[N]

mov eax, 0

lea ebx, [X]

mov ecx, [N]

call sum

mov [ne], eax

mov eax, 0

lea ebx, [X+4]

mov ecx, [N]

call sum

mov [c], eax

;cdq

mov eax, [ne]

mov edx, 0

mov ebx, [c]

idiv ebx

push eax

cinvoke printf, text3, [c]

cinvoke printf, text4, [ne];??????

pop eax

cinvoke printf, text6, eax

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'

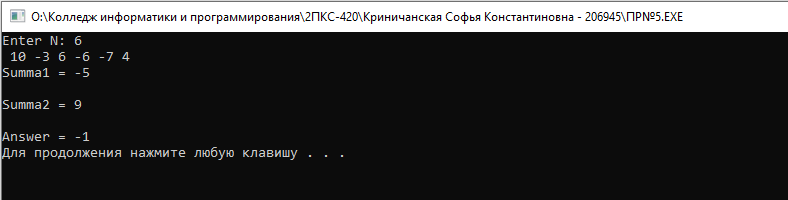


Рисунок 5. Выводимый результат

# **Практическая работа №6**

Задание:

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.

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("Криничанская Софья 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");

}

}

}

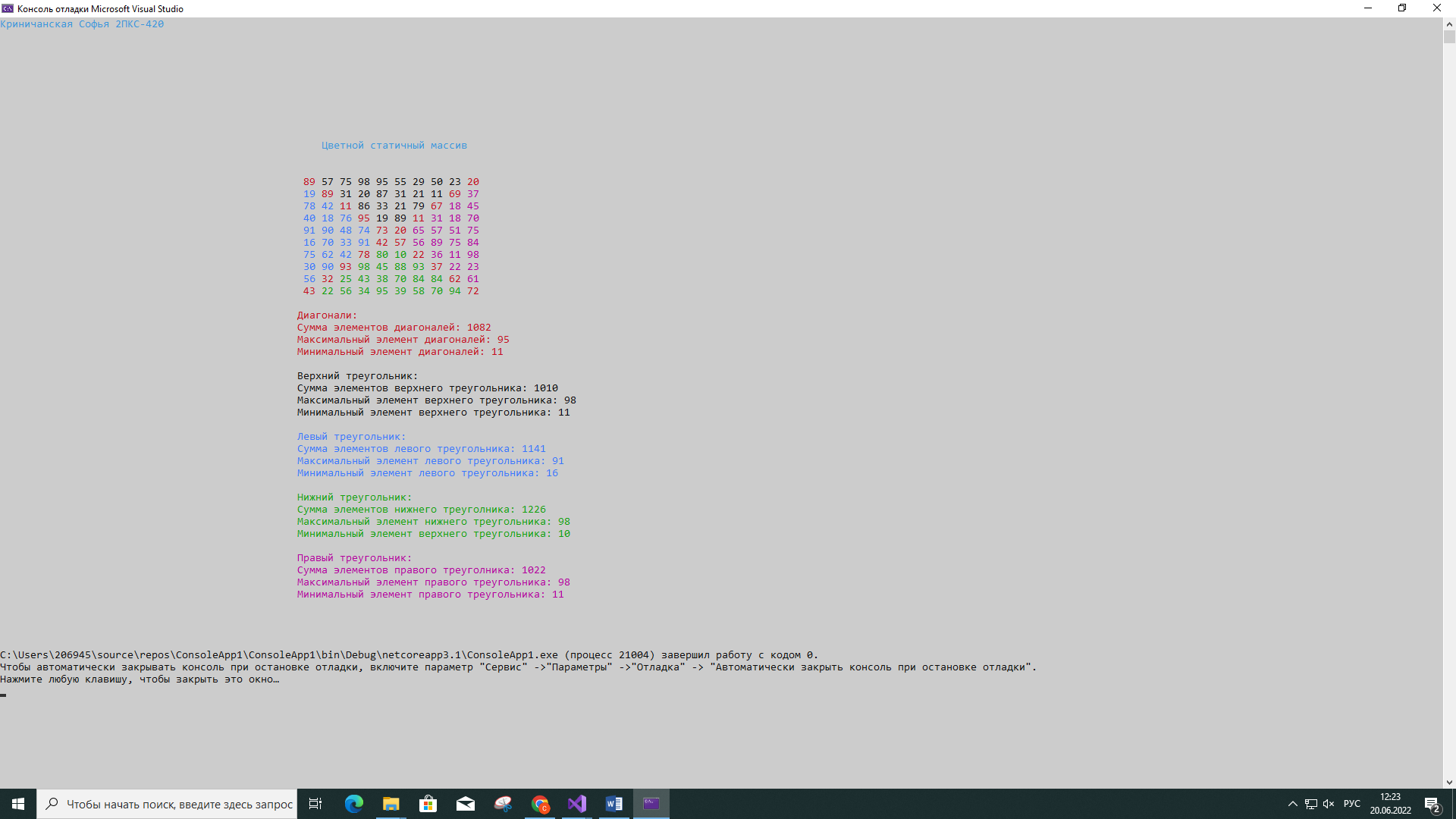


Рисунок 1. Вывод консоли

# **Практическая работа №7**

Задание:

Создать форму вычислить функцию, под ней поле с функцией:

Значение, а вводится пользователем, при нажатии на кнопку вычислить – вычислить функцию. Сетка грид под этим, в которой столбцы x | y со значениями. Также добавить кнопки «выход» - выйти из файла, «сохранить» - сохранить в файл и «вывести из файла» - вынести значения в таблицу из сохраненного ранее файла.

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 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();

}

private void Button2\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

public void TextBox\_TextChanged(object sender, RoutedEventArgs e)

{

this.Show();

}

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

{

textBox1.TabIndex = 0;

textBox1.Focus();

}

private void Button1\_click(object sender, RoutedEventArgs e)

{

int a, x;

double y;

int count = 0;

try

{

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

a = int.Parse(textBox1.Text);

for (x = -20; x < 21; x++)

{

if (textBox1.Text == "")

{

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

}

if (x + a == 1)

{

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

result.Add(new list(x, "None"));

}

if (x < 0)

{

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

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

}

else

{

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

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

}

}

DataGridView1.ColumnWidth = 97;

DataGridView1.ItemsSource = result;

}

catch (FormatException)

{

MessageBox.Show("Неправильный формат!");

textBox1.Focus();

return;

}

catch

{

MessageBox.Show("Переполнение!");

textBox1.Focus();

return;

}

button2.IsEnabled = true;

}

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

{

}

private void DataGridView1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

}

}

Код 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>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="289\*"/>

<ColumnDefinition Width="503\*"/>

</Grid.ColumnDefinitions>

<Grid.Background>

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

<GradientStop Color="#FFBF80FF" Offset="0"/>

</LinearGradientBrush>

</Grid.Background>

<Button Name="button1" Content="Вычислить" HorizontalAlignment="Left" Margin="75,255,0,0" VerticalAlignment="Top" Width="190" RenderTransformOrigin="0.5,0.5" Height="60" Click="Button1\_click" BorderBrush="#FFEEDDDD" FontWeight="Bold" FontSize="16">

<Button.Background>

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

<GradientStop Color="#FFE1D5FF" Offset="0"/>

</LinearGradientBrush>

</Button.Background>

<Button.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="-0.053"/>

<TranslateTransform/>

</TransformGroup>

</Button.RenderTransform>

</Button>

<Button Name="button2" Content="Выход" HorizontalAlignment="Left" Margin="32,357,0,0" VerticalAlignment="Top" Width="106" Height="29" Click="Button2\_Click" BorderBrush="#FFEEDBDB" FontWeight="Bold" FontSize="16">

<Button.Background>

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

<GradientStop Color="#FFE1D5FF" Offset="0"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<DataGrid Name="DataGridView1" HorizontalAlignment="Left" Height="254" Margin="180,132,0,0" VerticalAlignment="Top" Width="220" Grid.Column="1" Foreground="Black" BorderBrush="#FFD4DCE4" OpacityMask="Black"/>

<TextBlock HorizontalAlignment="Left" Margin="194,63,0,0" TextWrapping="Wrap" Text="A=" VerticalAlignment="Top" Height="23" Width="38" RenderTransformOrigin="0.862,3.5" Foreground="#FFF1E7E7" FontWeight="Bold" FontStyle="Italic" FontSize="16" Grid.Column="1"/>

<TextBox Name="textBox1" HorizontalAlignment="Left" Height="22" Margin="220,64,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="180" Grid.Column="1"/>

<TextBlock HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top" Margin="157,102,0,0" Grid.Column="1"/>

<Label Content="Введите A:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="194,24,0,0" Foreground="#FFFBECEC" FontSize="16" FontWeight="Bold" FontStyle="Italic" Grid.Column="1"/>

<Label Content="Формула:" HorizontalAlignment="Left" Margin="51,94,0,0" VerticalAlignment="Top" Width="220" Foreground="#FFF9EAEA" Height="31" FontSize="18" FontStyle="Italic" FontWeight="Bold"/>

<Button Content="Вывести из файла" Grid.ColumnSpan="2" HorizontalAlignment="Left" Height="34" Margin="161,354,0,0" VerticalAlignment="Top" Width="177" Click="Button\_Click" Background="#FFE1D5FF" FontSize="16" FontWeight="Bold"/>

<Viewbox Margin="51,134,451,217" Grid.ColumnSpan="2" Stretch="Fill">

<Image Stretch="Fill" Source="Снимок экрана 2022-05-30 206945.png" Height="137" Width="360"/>

</Viewbox>

</Grid>

</Window>

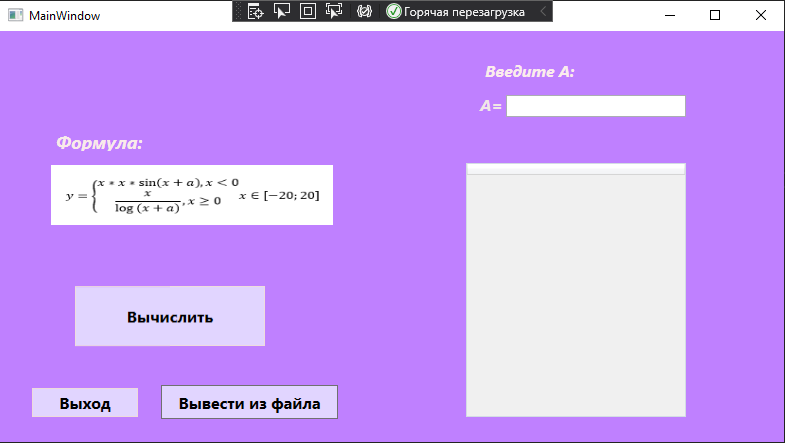


Рисунок 2. Работа с Grid

# **Практическая работа №8**

Задание:

Создать программу с использованием листбокса. Добавить окно ввода списка продуктов, обработать исключения (пробелы и тд). Добавить кнопки:

- «Добавить» - добавляет в листбокс наименование товара

- «Редактировать» - Выбрать в списке позицию мышкой и изменить введенный текст.

- «Удалить» - Выбрать в списке позицию мышкой и удалить, высветится окно с подтверждением операции.

- «Очистить всё» - очищает весь список, с предупреждением

- «Сортировать» - сортировка списка по алфавиту

- «Выход» - выйти из программы.

Также отображается счетчик записей в списке под окном ввода позиций в список.

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();

}

}

}

}

Код 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>

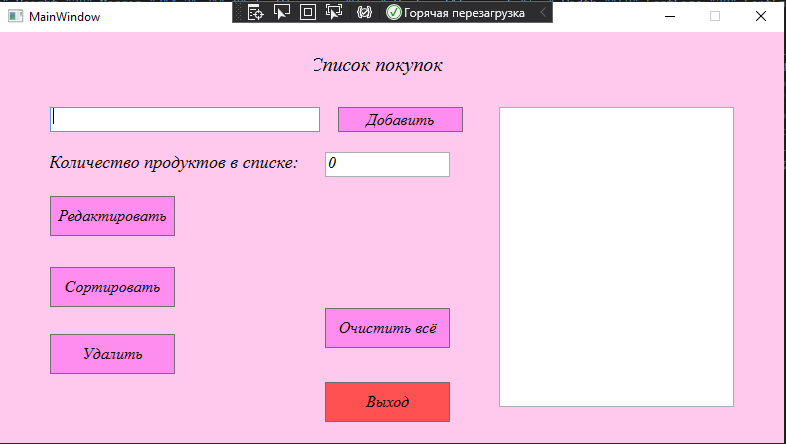
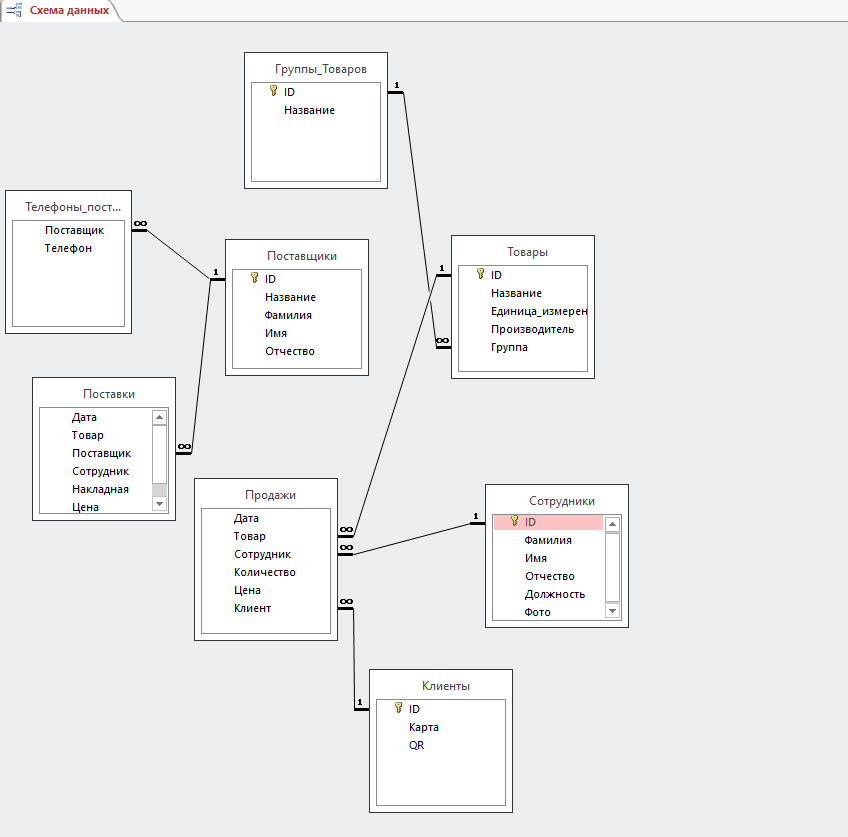


Рисунок 3. Работа с ListBox

# **Практическая работа №9**

Задание:

Создать базу данных торгового предприятия. Создать таблицы: «Товар», «Группа товаров», «Сотрудники», «Поставщики», «Клиенты», «Продажи», «Телефоны поставщиков». И создать связи между ними

Рисунок 4. Схема данных

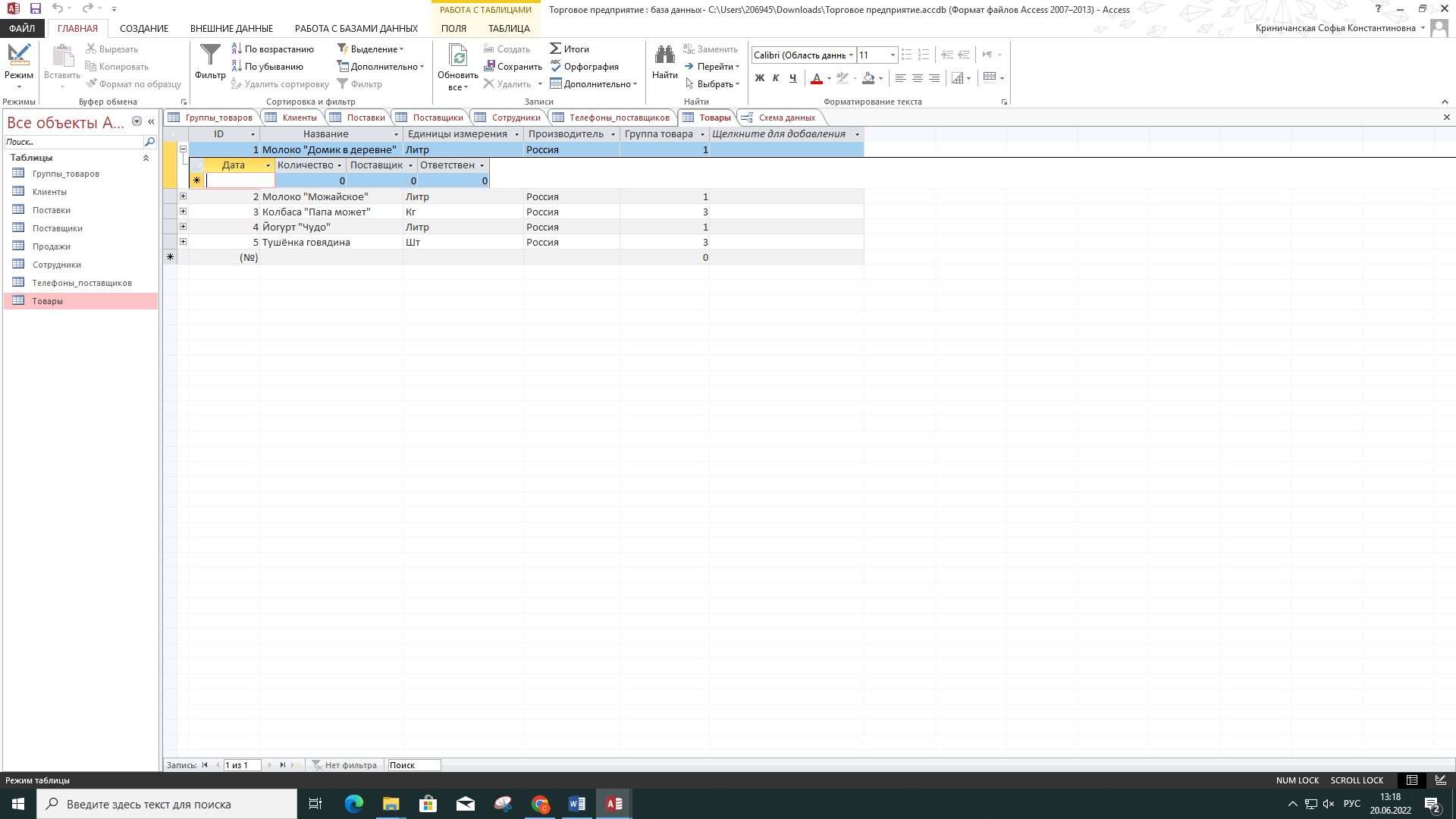


Рисунок 5. Работа с базой данных

# **Практическая работа №10**

Задание:

Разработать калькулятор на языке C#. В проекте использовать приём создания одного обработчика для нескольких событий.

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 Calculator

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void CE(object sender, RoutedEventArgs e)

{

TextBox.Clear();

}

private void C1\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 1;

}

private void C2\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 2;

}

private void C3\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 3;

}

private void C4\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 4;

}

private void C5\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 5;

}

private void C6\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 6;

}

private void C7\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 7;

}

private void C8\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 8;

}

private void C9\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 9;

}

private void C0\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 0;

}

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

{

TextBox.Text += "√";

}

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

{

TextBox.Text += "/";

}

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

{

TextBox.Text += "\*";

}

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

{

TextBox.Text += "-";

}

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

{

TextBox.Text += "+";

}

private void equality(object sender, RoutedEventArgs e)

{

TextBox.Text = Convert.ToString(Eval(TextBox.Text)).Replace(",", ".");

}

static Double Eval(String expression)

{

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

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

}

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

{

TextBox.Text += ".";

}

private void Button\_Click\_6(object sender, RoutedEventArgs e)

{

TextBox.Text = TextBox.Text.Remove(TextBox.Text.Length - 1);

}

}

}

Код XAML:

<Window x:Class="Calculator.MainWindow"

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

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

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

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

xmlns:local="clr-namespace:Calculator"

mc:Ignorable="d"

Title="Практическая работа №7 Криничанская С.К. 3ПКС-420" Height="450" Width="442">

<Grid Margin="-2,0,0,-6" HorizontalAlignment="Left" Width="444" Background="#FF8F8F8F">

<Grid.RowDefinitions>

<RowDefinition Height="83\*"/>

<RowDefinition Height="336\*"/>

<RowDefinition Height="11\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" HorizontalAlignment="Left" IsEnabled="False" Height="74" Margin="9,17,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="403" RenderTransformOrigin="0.496,0.433" Grid.RowSpan="2"/>

<Button Content="CE" HorizontalAlignment="Left" Margin="342,52,0,0" VerticalAlignment="Top" Width="72" Height="51" Click="CE" FontSize="22" Grid.Row="1" Background="#FFD4FFFA" />

<Button Name ="С1" Content="1" HorizontalAlignment="Left" Margin="11,187,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C1\_Click" FontSize="22" FontWeight="Bold" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С2" Content="2" HorizontalAlignment="Left" Margin="94,187,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C2\_Click" FontSize="22" FontWeight="Bold" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С3" Content="3" HorizontalAlignment="Left" Margin="177,187,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C3\_Click" FontSize="22" FontWeight="Bold" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С4" Content="4" HorizontalAlignment="Left" Margin="11,117,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C4\_Click" FontSize="22" FontWeight="Bold" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С5" Content="5" HorizontalAlignment="Left" Margin="94,117,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C5\_Click" FontWeight="Bold" FontSize="22" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С6" Content="6" HorizontalAlignment="Left" Margin="177,117,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C6\_Click" FontSize="22" FontWeight="Bold" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С7" Content="7" HorizontalAlignment="Left" Margin="11,52,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C7\_Click" FontSize="22" FontWeight="Bold" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С8" Content="8" HorizontalAlignment="Left" Margin="94,52,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C8\_Click" FontWeight="Bold" FontSize="22" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С9" Content="9" HorizontalAlignment="Left" Margin="177,52,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="C9\_Click" FontSize="22" FontWeight="Bold" Grid.Row="1" Background="#FF88C9C1"/>

<Button Name ="С0" Content="0" HorizontalAlignment="Left" Margin="11,260,0,0" VerticalAlignment="Top" Width="154" Height="51" Click="C0\_Click" FontWeight="Bold" FontSize="22" Grid.Row="1" Background="#FF88C9C1"/>

<Button Content="." HorizontalAlignment="Left" Margin="177,260,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Button\_Click\_5" FontSize="22" Grid.Row="1" Background="#FFD4FFFA"/>

<Button Content="/" HorizontalAlignment="Left" Margin="260,52,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_1" FontSize="22" Grid.Row="1" Background="#FFD4FFFA"/>

<Button Content="\*" HorizontalAlignment="Left" Margin="260,117,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_2" FontSize="22" Grid.Row="1" Background="#FFD4FFFA"/>

<Button Content="-" HorizontalAlignment="Left" Margin="260,187,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_3" FontSize="22" Grid.Row="1" Background="#FFD4FFFA"/>

<Button Content="+" HorizontalAlignment="Left" Margin="260,260,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_4" FontSize="22" Grid.Row="1" Background="#FFD4FFFA"/>

<Button Content="=" HorizontalAlignment="Left" Margin="342,187,0,0" VerticalAlignment="Top" Width="72" Height="124" Click="ravno" FontSize="24" Grid.Row="1" Background="#FFD4FFFA"/>

<Button Content="С" HorizontalAlignment="Left" Margin="342,117,0,0" VerticalAlignment="Top" Width="72" Height="51" Click="Button\_Click\_6" FontSize="22" Grid.Row="1" Background="#FFD4FFFA"/>

</Grid>

</Window>

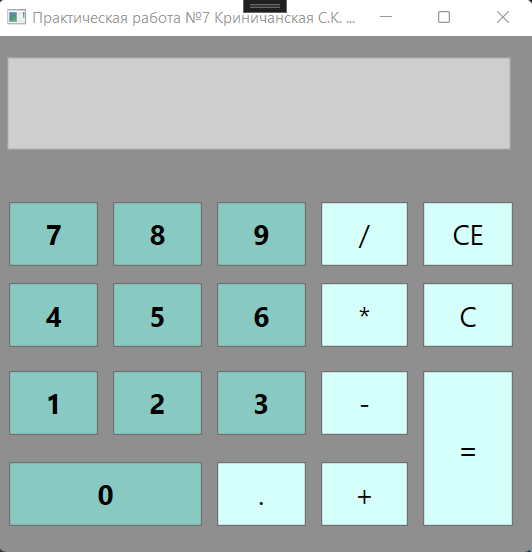


Рисунок 1 Калькулятор

# **Практическая работа №11**

Задание:

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.

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 ORFOGRAPHIA\_

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

this.Hide();

}

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

{

// Создаем новый экземпляр класса Word.Application:

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

Word1.Visible = false;

//Открываем новый документ MS Word:

var Doc = Word1.Documents.Add();

// Вводим в документ MS Word текст из текстового поля:

Doc.Words.First.InsertBefore(tb\_orfo.Text);

// Непосредственная проверка орфографии:

Doc.CheckSpelling();

// Получаем исправленный текст:

var NewText = Doc.Content.Text;

// Возвращаем в текстовое поле исправленный текст:

tb\_orfo.Text = NewText;

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

//Закрываем документ Word без сохранения

Word1.Quit();

Word1 = null;

}

}

}

Код XAML:

<Window x:Class="ORFOGRAPHIA\_.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:ORFOGRAPHIA\_"

mc:Ignorable="d"

Title="Практическая работа №8 Криничанская Софья 3ПКС-420" Height="450" Width="800" Background="{x:Null}">

<Grid Background="#FF7AC8B7">

<Button x:Name="check" Content="Проверить" HorizontalAlignment="Center" Height="56" Margin="0,245,0,0" VerticalAlignment="Top" Width="208" Click="check\_Click" Foreground="Black" Background="#FFCEFBFF" BorderBrush="#FF4B0707" FontFamily="Bell MT" FontSize="22" Opacity="0.9" FontWeight="Bold">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

<Button x:Name="back" Content="Назад" HorizontalAlignment="Center" Height="56" Margin="0,329,0,0" VerticalAlignment="Top" Width="207" Click="back\_Click" Foreground="Black" Background="#FFCEFBFF" BorderBrush="#FF4B0707" FontFamily="Bell MT" FontWeight="Bold" FontSize="22" Opacity="0.9">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

<TextBox x:Name="tb\_orfo" SpellCheck.IsEnabled="True" Language="ru-ru" HorizontalAlignment="Center" Height="151" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="697" Margin="31,63,0,0" Background="#00000000" BorderBrush="Black"/>

</Grid>

</Window>



Рисунок 2. Окно проекта “Проверка орфографии”

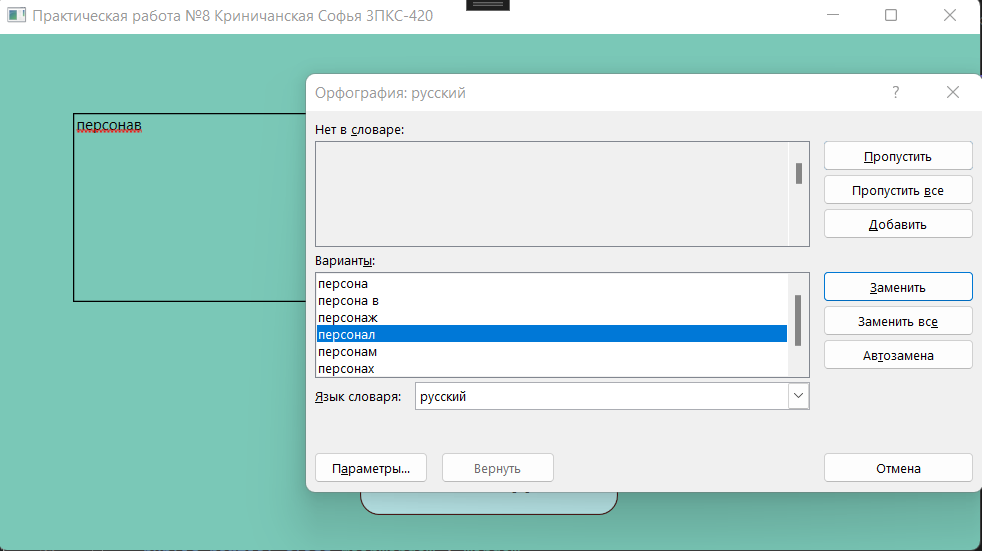


Рисунок 3. Работа проекта

# **Практическая работа №12**

Задание:

Необходимо разработать программу, которая при нажатии на кнопку создаёт таблицу в MS Word.

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 TABLE

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Exit(object sender, RoutedEventArgs e)

{

System.Windows.Application.Current.Shutdown();

}

private void Vord(object sender, RoutedEventArgs e)

{

createtable();

}

private void createtable()

{

object oMissing = System.Reflection.Missing.Value;

object oEndOfDoc = "\\endofdoc";

Microsoft.Office.Interop.Word.\_Application objWord;

Microsoft.Office.Interop.Word.\_Document objDoc;

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

objWord.Visible = true;

objDoc = objWord.Documents.Add(ref oMissing, ref oMissing,

ref oMissing, ref oMissing);

int i = 0;

int j = 0;

Microsoft.Office.Interop.Word.Table objTable;

Microsoft.Office.Interop.Word.Range wrdRng = objDoc.Bookmarks.get\_Item(ref oEndOfDoc).Range;

objTable = objDoc.Tables.Add(wrdRng, 3, 4, ref oMissing, ref oMissing);

objTable.Range.ParagraphFormat.SpaceAfter = 7;

string strText;

for (i = 1; i <= 3; i++)

for (j = 1; j <= 4; j++)

{

strText = "Строка" + i + " Столбец" + j;

objTable.Cell(i, j).Range.Text = strText;

}

objTable.Rows[1].Range.Font.Bold = 1;

objTable.Rows[1].Range.Font.Italic = 1;

objTable.Borders.Shadow = true;

this.Close();

}

}

}

Код XAML:

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

mc:Ignorable="d"

Title="Практическая работа №9 Криничанская С.К. 3ПКС-420" Height="450" Width="800">

<Grid Background="#FF72DCD0">

<Button Content="WORD" HorizontalAlignment="Left" Margin="118,103,0,0" VerticalAlignment="Top" Height="66" Width="207" FontSize="22" FontFamily="Bell MT" FontWeight="Bold" Click="Vord">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

<Button Content="Выход" HorizontalAlignment="Left" Margin="448,237,0,0" VerticalAlignment="Top" Height="66" Width="207" FontSize="22" FontFamily="Bell MT" FontWeight="Bold" Click ="Exit">

<Button.Resources>

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

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

</Grid>

</Window>



Рисунок 4. Окно проекта

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5 Создание таблица в MS WORD

**Практическая работа №11**

Задание:

Cоздать собственный проект с использованием функций MS Excel не используя функцию нахождения числа Pi.

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 FUNKCII

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void ToRad(object sender, EventArgs e)

{

// Создание экземпляра класса Excel.Application:

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

var rad = XL.WorksheetFunction.Radians(Convert.ToDouble(TextBox.Text));

XL.Quit();

}

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

{

// Создание экземпляра класса Excel.Application:

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

var rad = XL.WorksheetFunction.Radians(Convert.ToDouble(TextBox.Text));

TxtBlck.Text = Convert.ToString(rad);

XL.Quit();

}

}

}

Код XAML:

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

mc:Ignorable="d"

Title="Практическая работа №10 Криничанская С.К. 3ПКС-420" Height="450" Width="800" Background="LightSkyBlue">

<Grid Background="#FF6DFFEA">

<Grid.RowDefinitions>

<RowDefinition Height="131\*"/>

<RowDefinition Height="86\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" Background="#FFAFF3FF" HorizontalAlignment="Left" Margin="279,94,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="212" Height="37"/>

<Label Content="Введите градусы&#xD;&#xA;" HorizontalAlignment="Left" Margin="300,25,0,187" Height="50" Width="170" HorizontalContentAlignment="Center" VerticalContentAlignment="Bottom" FontSize="18" FontWeight="Bold" FontFamily="Bell MT"/>

<Button Content="Перевести в радианы" HorizontalAlignment="Left" Margin="267,165,0,0" VerticalAlignment="Top" Click="Button\_Click" Height="52" Width="236" FontFamily="Bell MT" FontSize="18" RenderTransformOrigin="-0.004,0.219" Background="#FF46B2A1" Foreground="White" FontWeight="Bold"/>

<Label Content="Ответ" HorizontalAlignment="Left" Margin="300,242,0,0" VerticalAlignment="Top" FontSize="18" FontWeight="Bold" FontFamily="Bell MT" Width="170" HorizontalContentAlignment="Center" RenderTransformOrigin="0.705,0.653" Grid.RowSpan="2"/>

<TextBox Name="TxtBlck" Background="#FFAFF3FF" HorizontalAlignment="Left" Margin="279,15,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="212" Height="37" Grid.Row="1"/>

</Grid>

</Window>

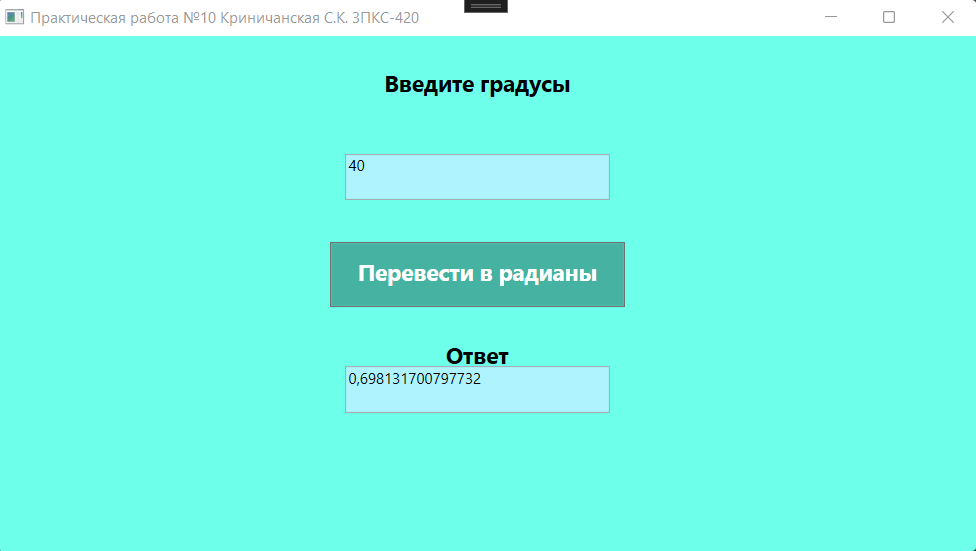


Рисунок 6. Окно проекта “Использование функций MS Excel”

# **Практическая работа №13**

Задание:

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 кодировкой Win1251
* Сохранение информации в файл с разрешением RTF
* Сохранение информации в бинарный файл
* Печать текстового документа.

Код программы:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.Serialization.Formatters.Binary;

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 uni

{

public partial class MainWindow : Window

{

Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();

Microsoft.Win32.OpenFileDialog odlg = new Microsoft.Win32.OpenFileDialog();

public MainWindow()

{

InitializeComponent();

dlg.FileName = "Документ";

odlg.FileName = "Документ";

}

private void Save\_as\_RTF(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".rtf";

dlg.Filter = "Текстовый документ (.rtf)|\*.rtf";

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

File.WriteAllText(filename, richText);

MessageBox.Show("Файл успешно сохранён!");

}

private void Save\_as\_Unicode(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".txt";

dlg.Filter = "Текстовый документ (.txt)|\*.txt";

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

File.WriteAllText(filename, richText, Encoding.Unicode);

MessageBox.Show("Файл успешно сохранён!");

}

private void Save\_as\_Win1251(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".txt";

dlg.Filter = "Текстовый документ (.txt)|\*.txt";

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

File.WriteAllText(filename, richText, Encoding.GetEncoding("windows-1251"));

MessageBox.Show("Файл успешно сохранён!");

}

private void Save\_as\_BinaryFormat(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".bin";

dlg.Filter = "Текстовый документ (.bin)|\*.bin";

BinaryFormatter bf = new BinaryFormatter();

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

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

{

writer.Write(richText);

MessageBox.Show("Файл успешно сохранён!(на рабочий стол)");

}

}

private void Open\_as\_RTF(object sender, RoutedEventArgs e)

{

rtxtbx.Document.Blocks.Clear();

odlg.DefaultExt = ".rtf";

odlg.Filter = "Текстовый документ (.rtf)|\*.rtf";

bool? dialogResult = odlg.ShowDialog();

if (dialogResult.HasValue && dialogResult.Value)

{

TextRange range = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd);

System.IO.FileStream file = new System.IO.FileStream(odlg.FileName, System.IO.FileMode.Open, System.IO.FileAccess.Read);

range.Load(file, System.Windows.DataFormats.Rtf);

file.Dispose();

MessageBox.Show("Файл открыт!");

}

}

private void Open\_as\_Unicode(object sender, RoutedEventArgs e)

{

string filename = "";

rtxtbx.Document.Blocks.Clear();

bool? result = odlg.ShowDialog();

if (result == true)

{

filename = odlg.FileName;

}

string filetext = File.ReadAllText(filename);

rtxtbx.Document.Blocks.Add(new Paragraph(new Run(filetext)));

MessageBox.Show("Файл открыт!");

}

private void Open\_as\_Win1251(object sender, RoutedEventArgs e)

{

string filename = "";

rtxtbx.Document.Blocks.Clear();

bool? result = odlg.ShowDialog();

if (result == true)

{

filename = odlg.FileName;

}

string filetext = File.ReadAllText(filename, System.Text.Encoding.GetEncoding("windows-1251"));

rtxtbx.Document.Blocks.Add(new Paragraph(new Run(filetext)));

MessageBox.Show("Файл открыт!");

}

private void Open\_as\_Binary(object sender, RoutedEventArgs e)

{

string filename = "";

rtxtbx.Document.Blocks.Clear();

bool? result = odlg.ShowDialog();

if (result == true)

{

filename = odlg.FileName;

}

string filetext = File.ReadAllText(filename);

rtxtbx.Document.Blocks.Add(new Paragraph(new Run(filetext)));

MessageBox.Show("Файл открыт!");

}

private void PrintWindow(object sender, RoutedEventArgs e)

{

PrintDialog p = new PrintDialog();

if (p.ShowDialog() == true)

{

p.PrintVisual(grid1, "Печать");

}

}

}

}

Код XAML:

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

mc:Ignorable="d"

Title="Практическая работа №10 Криничанская С.К. 3ПКС-420" Height="450" Width="800">

<Grid Name="grid1">

<Grid.RowDefinitions>

<RowDefinition Height="177\*"/>

<RowDefinition Height="40\*"/>

</Grid.RowDefinitions>

<Menu Margin="0,0,0,305">

<MenuItem Header="Файл" FontSize="20" FontFamily="Bell MT">

<MenuItem Header="Открыть" FontSize="20">

<MenuItem Header="Открыть в формате Unicode" Click="Open\_as\_Unicode" FontSize="20"></MenuItem>

<MenuItem Header="Открыть в формате Win1251" Click="Open\_as\_Win1251" FontSize="20"></MenuItem>

<MenuItem Header="Открыть в формате RTF" Click="Open\_as\_RTF" FontSize="20" ></MenuItem>

<MenuItem Header="Открыть бинарный файл" Click="Open\_as\_Binary" FontSize="20"></MenuItem>

</MenuItem>

<MenuItem Header="Сохранить как" FontSize="20">

<MenuItem Header="в формате RTF" Click="Save\_as\_RTF" FontSize="20"></MenuItem>

<MenuItem Header="в кодировке Unicode" Click="Save\_as\_Unicode" FontSize="20"></MenuItem>

<MenuItem Header="в кодировке Win1251" Click="Save\_as\_Win1251" FontSize="20"></MenuItem>

<MenuItem Header="в бинарный файл" Click="Save\_as\_BinaryFormat" FontSize="20"></MenuItem>

</MenuItem>

<MenuItem Header="Распечатать" Click="PrintWindow" FontSize="20"></MenuItem>

</MenuItem>

</Menu>

<RichTextBox Name="rtxtbx" HorizontalAlignment="Center" Background="#FF99FFE2" VerticalAlignment="Top" Width="692" Height="256" Margin="0,89,0,0">

<FlowDocument>

<Paragraph>

<Run/>

</Paragraph>

</FlowDocument>

</RichTextBox>

</Grid>

</Window>

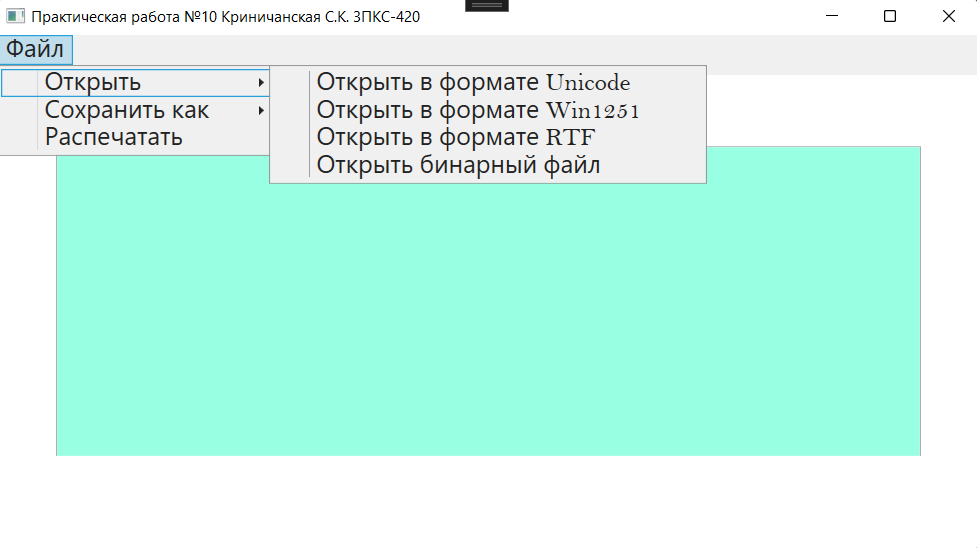


Рисунок 7. Окно проекта “Использование функций MS Excel”

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8. Окно проекта “Использование функций MS Excel”