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

Дронникова А.М.

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

И. В. Сибирев

*(И.О. Фамилия)*

Е. Л. Альшакова

*(И.О. Фамилия)*

**Москва**

**20­­­­22**

Оглавление

[Перечень заданий/работ, выполненных в ходе учебной практики 3](#_Toc121241923)

[Практическая работа №1 (от 19.06.2022) 8](#_Toc121241924)

[*Задание 1* 8](#_Toc121241925)

[*Задание 2* 9](#_Toc121241926)

[*Задание 3* 10](#_Toc121241927)

[*Задание 4* 10](#_Toc121241928)

[*Задание 5* 11](#_Toc121241929)

[Практическая работа №2 (от 21.06.2022) 13](#_Toc121241930)

[*Задание 1* 13](#_Toc121241931)

[*Задание 2* 16](#_Toc121241932)

[*Задание 3* 19](#_Toc121241933)

[*Задание 4* 22](#_Toc121241934)

[Практическая работа №3. 25](#_Toc121241935)

[Практическая работа №4 33](#_Toc121241936)

[Практическая работа №5 35](#_Toc121241937)

[Практическая работа №6 37](#_Toc121241938)

[Практическая работа №7 39](#_Toc121241939)

# **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 *(от 19.06.2022)***

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Темы учебной практики** | **Выполненные задания/работы** |
| **1** | **Локальные данные процедур** | Дано 50 чисел, среди которых есть по крайней мере одно отрицательное. Найти наибольшее среди отрицательных чисел |
| **2** | **Процедуры** | Задание 5. Ввести одномерный массив m = {-1, 0, 10, -3, -5, 6, -2, 3, 4}. Вычислить сумму элементов с нечетными номерами. Сформировать и вывести массив n, элементами которого являются индексы положительных элементов массива m. Программу разработать на основе пользовательских функций. |
| **3** | **Рекурсивные процедуры** | Рекурсивно описать процедуру RevPrint(N), которая печатает в обратном порядке цифры десятичной записи целого неотрицательного числа N. Например, RevPrint(12345) должна вывести текст 54321. |
| **4** | **Обработка двумерных массивов** | Разместить в памяти компьютера матрицу 4\*3 целых беззнаковых чисел. Написать программу, позволяющую найти сумму всех строк массива. Исходную матрицу и полученные суммы вывести на экран. |
| **5** | **Обработка одномерных массивов** | Вариант 8. Написать программу для последовательности чисел C1, C2, ..., Cn, которая выводит номер первого положительного элемента. |

## *Задание 1*

|  |  |
| --- | --- |
| proc func  mov ECX,[N]  lea ESI, [X]  mov EBX,0  mov DL,0  SEARCH:  lodsd  or EAX,EAX  jns NEXT  or DL,DL  jnz COMPARE  mov EBX,EAX  mov DL,1  COMPARE:  cmp EAX,EBX  jle NEXT  mov BX,AX  NEXT: loop SEARCH  cinvoke printf, text1,ebx  invoke sleep, 9000  invoke exit, 0  mov sp,bp  pop bp  ret  endp  call func  invoke sleep, 9000  invoke exit, 0  text1 db 'Max negative= %d',0  fpp db '%d',9, 0  N dd 50  X dd -3, 0, 2, 5, 10, -5, 0, 77, 69, -13, -14, 9, 1, 4, 1, 8, 9, 0, -3, 10, 19, -42, 5, 90,71  dd 9, 5, -9, 45, -90, -10, -1, 6, 14, 456, 122, 34, 87, 99, 12, 11, 321, 67, 81, 44, 92, 55, 2, -89, 100 |  |

## *Задание 2*

|  |  |
| --- | --- |
| macro prnt op1, op2 {  local metka  mov ebx, op1  mov ecx,op2  mov eax, 0  metka :  mov eax,[ebx]  push ecx  cinvoke printf, fpp, eax  pop ecx  add ebx, 4  loop metka  }  prnt m,9  lea ebx, [m]  mov ecx,5  mov eax,0  cycl:  add eax, [ebx]  add ebx,8  loop cycl  cinvoke printf, text1, eax  lea ebx, [m]  mov ecx,9  mov eax,0  mov edx,0  mov esi,n  cycl2:  mov eax,[ebx]  cmp eax,0  jg met  add ebx,4  inc edx  loop cycl2  met:  inc [amount]  mov [esi], edx  add esi, 4  add ebx,4  inc edx  loop cycl2  prnt n,[amount] | **Изображение выглядит как текст  Автоматически созданное описание** |

## *Задание 3*

|  |  |
| --- | --- |
| proc RevPrint  mov eax, [n]  mov ecx, 10  idiv ecx  mov [n], eax  push eax  cinvoke printf,text2,edx  pop eax  cmp eax,0  ja met  ret  met:  call RevPrint  ret  endp  cinvoke printf,text1  cinvoke scanf, fmat, n  mov edx, 0  call RevPrint | Изображение выглядит как текст  Автоматически созданное описание |

## *Задание 4*

|  |  |
| --- | --- |
| format PE console  include 'win32a.inc'  start:  macro prnt op1, op2,op3 {  local metka  mov esi,op1  mov ecx,op2  met2:  push ecx  mov ecx,op3  met1:  push ecx  mov eax,[esi]  add [sum],eax  push ecx  cinvoke printf, fpp, eax  pop ecx  add esi, 4  pop ecx  loop met1  ;cinvoke printf,text1  cinvoke printf,text3,[sum]  mov [sum],0  pop ecx  loop met2  }  cinvoke printf,text2  prnt m,4,3  cinvoke system,p  invoke exit,0  text1 db 10, 13, '', 10,0  text2 db 10, 13, 'Matrix 4x3', 10,0  text3 db 10, 13, 'Sum=%d',10,0  fpp db '%d ', 0  spp db '%d',0  p db 'pause',0  amount dd 0  m dd 1,0,10; 11  dd 3,5,6 ; 14 ;45  dd 5,3,4 ;12  dd 3,2,3 ;8  sum dd 0  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 | Изображение выглядит как текст  Автоматически созданное описание |

## *Задание 5*

|  |  |
| --- | --- |
| format PE console  include 'win32a.inc'  start:  macro prnt op1, op2 {  local metka  mov ebx, op1  mov ecx,op2  mov eax, 0  metka :  mov eax,[ebx]  push ecx  cinvoke printf, fpp, eax  pop ecx  add ebx, 4  loop metka  }  prnt m,10  lea ebx, [m]  mov ecx,10  mov eax,[ebx]  add ebx,4  cycl:  mov edx, [ebx]  cmp edx,0  jg metka  add ebx,4  loop cycl  metka:  mov eax,11  mov ebx, ecx  sub eax,ebx  cinvoke printf, text1,eax  cinvoke system,p  invoke exit,0  text1 db 10, 13, 'Result= %d', 13, 10  fpp db '%d ', 0  spp db '%d',0  p db 'pause',0  amount dd 0  m dd 0,-3,4,3,5,7,0,4,89,10  count dd ?  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 |  |

# **Практическая работа №2** *(от 21.06.2022)*

## *Задание 1*

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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace УП\_01.\_01\_задание\_1

{

class Program

{

static void Main(string[] args)

{

void matrix(bool answer)

{

Console.Write("Введите порядок матрицы: ");

int N = Convert.ToInt32(Console.ReadLine());

if (N <= 0)

{

Console.WriteLine("Размер массива не может быть равен нулю!");

return;

}

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

int mainD = 0, pobochD = 0, upTr = 0, dwTr = 0, rtTr = 0, lfTr = 0;

Random rnd = new Random();

Console.CursorTop = 7;

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

{

Console.WriteLine();

Console.CursorLeft = 10;

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

{

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

if (i == j) // главная диагональ

{

Console.ForegroundColor = ConsoleColor.DarkCyan;

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

mainD += arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.DarkMagenta;

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

pobochD += arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.DarkGray;

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

upTr += arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.DarkGreen;

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

rtTr += arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.DarkRed;

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

dwTr += arr[i, j];

}

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

{

Console.ForegroundColor = ConsoleColor.DarkYellow;

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

Console.ResetColor();

lfTr += arr[i, j];

}

}

}

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

if (answer)

{

Console.CursorLeft = 10;

Console.ForegroundColor = ConsoleColor.DarkCyan;

Console.WriteLine("Сумма элементов главной диагонали: " + mainD);

Console.CursorLeft = 10;

Console.ForegroundColor = ConsoleColor.DarkMagenta;

Console.WriteLine("Сумма элементов побочной диагонали: " + pobochD);

Console.CursorLeft = 10;

Console.ForegroundColor = ConsoleColor.DarkGray;

Console.WriteLine("Сумма элементов верхнего треугольника: " + upTr);

Console.CursorLeft = 10;

Console.ForegroundColor = ConsoleColor.DarkGreen;

Console.WriteLine("Сумма элементов правого треугольника: " + rtTr);

Console.CursorLeft = 10;

Console.ForegroundColor = ConsoleColor.DarkRed;

Console.WriteLine("Сумма элементов нижнего треугольника: " + dwTr);

Console.CursorLeft = 10;

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.WriteLine("Сумма элементов левого треугольника: " + lfTr);

Console.ResetColor();

}

}

Console.WriteLine(" Дронникова 2ПКС-420");

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

Console.WriteLine("1 - да");

Console.WriteLine("2 - нет");

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

if (A == "1") matrix(true);

else if (A == "2") matrix(false);

else

{

Console.ForegroundColor = ConsoleColor.Red;

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

Console.ResetColor();

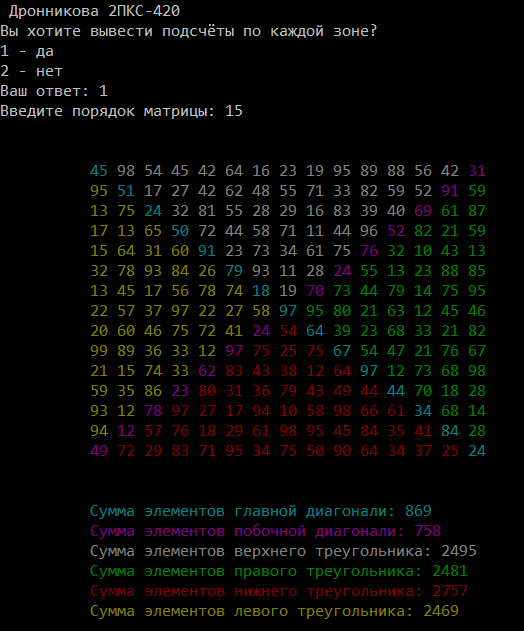
}

string n = Console.ReadLine();

}

}

}



## *Задание 2*

Создать форму вычисления функции.

Переменная а вводится пользователем.

Составляющие формы:

- кнопка «вычислить» функцию

- datagrid со значениями х (от -20 до 20) и у

- кнопка «записать в файл» данные из datagrid

- кнопка «считать из файла»

- кнопка «выход»

Код С#:

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 Zadanie2

{

public class XY

{

public int X { get; set; }

public string Y { get; set; }

public XY(int x, string y)

{

X = x;

Y = y;

}

}

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void TextBox\_TextChanged\_1(object sender, TextChangedEventArgs e)

{

}

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

{

if (A.Text.Length == 0) return;

List<XY> rezult = new List<XY>(40);

int a = Convert.ToInt32(A.Text);

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

{

if (i < 0) rezult.Add(new XY(i, Convert.ToString(Math.Round(i \* i \* Math.Sin(i + a), 2))));

else if (i >= 0)

{

if (a + i <= 0) rezult.Add(new XY(i, "Решения нет, аргумент логарифма не может быть <=0"));

else if (Math.Log10(i + a) == 0) rezult.Add(new XY(i, "Решения нет, деление на 0 невозможно"));

else rezult.Add(new XY(i, Convert.ToString(Math.Round(i / Math.Log(i + a), 2))));

}

}

rez.ColumnWidth = 97;

rez.ItemsSource = rezult;

}

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

{

}

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

{

}

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

{

string line;

StreamReader sr = new StreamReader("file.txt");

List<XY> rezult = new List<XY>();

line = sr.ReadLine();

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

{

rezult.Add(new XY(i, line));

//Read the next line

line = sr.ReadLine();

}

//close the file

sr.Close();

rez.ColumnWidth = 97;

rez.ItemsSource = rezult;

}

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

{

this.Close();

}

}

}

Код XAML:

<Window x:Class="Zadanie2.MainWindow"

WindowStartupLocation="CenterScreen"

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

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

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

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

xmlns:local="clr-namespace:Zadanie2"

mc:Ignorable="d"

Title="Дронникова 420" Height="450" Width="600" Background="#FFFF006E">

<Grid Background="#FFFFC1DE" >

<Image Source="Resources/formula.png" Margin="14,23,274.6,298" RenderTransformOrigin="0.524,0.475"></Image>

<TextBlock HorizontalAlignment="Left" Margin="14,10,0,0" TextWrapping="Wrap" Text="Вычислить функцию" VerticalAlignment="Top" Height="23" Width="226" FontSize="25" Foreground="#FFFF006E" FontFamily="Monotype Corsiva"/>

<TextBox Name="A" HorizontalAlignment="Left" Height="23" Margin="77,117,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="68" TextChanged="TextBox\_TextChanged\_1" FontSize="18" FontFamily="Monotype Corsiva"/>

<TextBlock HorizontalAlignment="Left" Margin="55,117,0,0" TextWrapping="Wrap" Text="a=" VerticalAlignment="Top" Height="23" Width="30" FontSize="20" Foreground="#FFFF006E" FontFamily="Monotype Corsiva"/>

<Button Name="Calculate" Content="Вычислить" HorizontalAlignment="Left" Margin="363,117,0,0" VerticalAlignment="Top" Width="183" Height="50" FontSize="20" Click="Calculate\_Click" FontFamily="Monotype Corsiva" Background="#FFFF4394" Foreground="White"/>

<Button Name="Write" Content="Записать в файл" HorizontalAlignment="Left" Margin="363,217,0,0" VerticalAlignment="Top" Width="183" Height="50" FontSize="20" Click="Write\_Click" FontFamily="Monotype Corsiva" Background="#FFFF4394" Foreground="White"/>

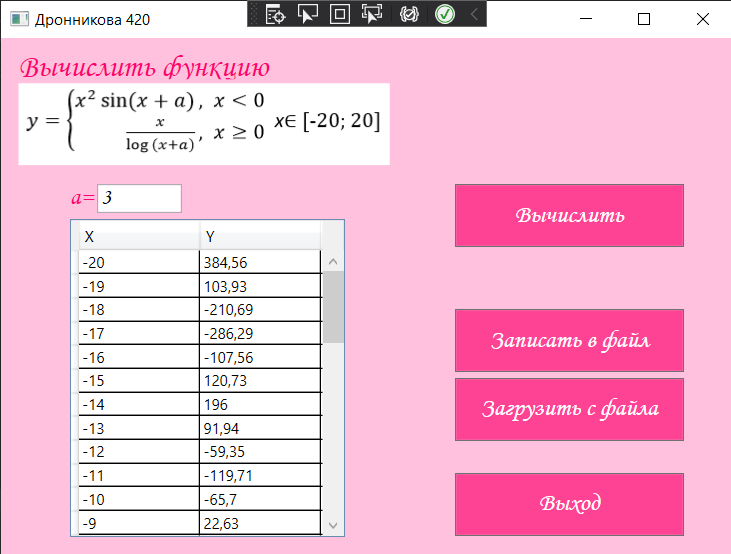
<Button Name="Read" Content="Загрузить с файла" HorizontalAlignment="Left" Margin="363,272,0,0" VerticalAlignment="Top" Width="183" Height="50" FontSize="20" Click="Read\_Click" FontFamily="Monotype Corsiva" Background="#FFFF4394" Foreground="White"/>

<Button Name="Exit" Content="Выход" HorizontalAlignment="Left" Margin="363,348,0,0" VerticalAlignment="Top" Width="183" Height="50" FontSize="20" Click="Exit\_Click" FontFamily="Monotype Corsiva" Background="#FFFF4394" Foreground="White"/>

<DataGrid Name="rez" HorizontalAlignment="Left" Height="254" Margin="55,145,0,0" VerticalAlignment="Top" Width="220"/>

</Grid>

</Window>



## *Задание 3*

Работа с ListBox

Составляющие формы:

- окно для ввода страны

- кнопка «добавить» (добавляет в ListBox введённое слово)

- кнопка «изменить» (выбрать элемент в ListBox, при нажатии на кнопку открывается изменение в TextBox с курсором в конце)

- кнопка «удалить» (удаляет один элемент) (предупреждение)

- кнопка «очистить» весь список (предупреждение)

- кнопка «сортировать» по алфавиту

- кнопка «выход»

Код С#:

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 Zadanie3

{

public partial class MainWindow : Window

{

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

public MainWindow()

{

InitializeComponent();

Show();

TB.Focus();

}

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

{

this.Close();

}

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

{

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

{

MessageBox.Show("Введите страну!");

TB.Clear();

TB.Focus();

return;

}

if (LB.Items.Contains(TB.Text))

{

MessageBox.Show("Эта страна уже есть в списке.");

TB.Clear();

TB.Focus();

return;

}

foreach(var i in TB.Text)

{

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

{

MessageBox.Show("Введите название страны без лишних знаков.");

TB.Focus();

return;

}

}

LB.Items.Add(TB.Text.Trim());

TB.Clear();

TB.Focus();

}

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

{

if (LB.SelectedItems.Count == 0)

{

MessageBox.Show("Выберите элемент из списка.", "Внимание", MessageBoxButton.OK, MessageBoxImage.Exclamation);

return;

}

TB.Text = (string)LB.SelectedItem;

LB.Items.Remove(LB.SelectedItem);

TB.Focus();

TB.SelectionStart = TB.Text.Length;

}

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

{

if (LB.SelectedItems.Count == 0)

{

MessageBox.Show("Выберите элемент из списка.", "Внимание", MessageBoxButton.OK, MessageBoxImage.Exclamation);

return;

}

LB.Items.RemoveAt(LB.SelectedIndex);

}

private void Del\_all\_Click(object sender, RoutedEventArgs e)

{

if (LB.Items.Count == 0)

{

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

return;

}

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

{

LB.Items.Clear();

LB.Focus();

}

}

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

{

if (LB.Items.Count == 0)

{

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

return;

}

LB.Items.SortDescriptions.Add(

new System.ComponentModel.SortDescription("",

System.ComponentModel.ListSortDirection.Ascending));

TB.Focus();

}

}

}

Код XAML:

<Window x:Class="Zadanie3.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:Zadanie3"

mc:Ignorable="d"

WindowStartupLocation="CenterScreen"

ResizeMode="CanMinimize"

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

<Grid Background="#FFFFC1DE">

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

<Label Content="Работа с ListBox" HorizontalAlignment="Left" Margin="62,31,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Sitka Banner" Foreground="#FFFF4394"/>

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

<Label Content="Введите страну" HorizontalAlignment="Left" Margin="62,286,0,0" VerticalAlignment="Top" FontSize="20" Height="36" FontFamily="Sitka Text" Foreground="#FFFF4394"/>

<Button Name="Dobav" IsDefault="True" Content="Добавить" HorizontalAlignment="Left" Margin="457,50,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Dobav\_Click" Background="#FFFF4394" Foreground="White"/>

<Button Name="Red" Content="Изменить" HorizontalAlignment="Left" Margin="457,109,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Red\_Click" Background="#FFFF4394" Foreground="White"/>

<Button Name="Del" Content="Удалить" HorizontalAlignment="Left" Margin="457,172,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Del\_Click" Background="#FFFF4394" Foreground="White"/>

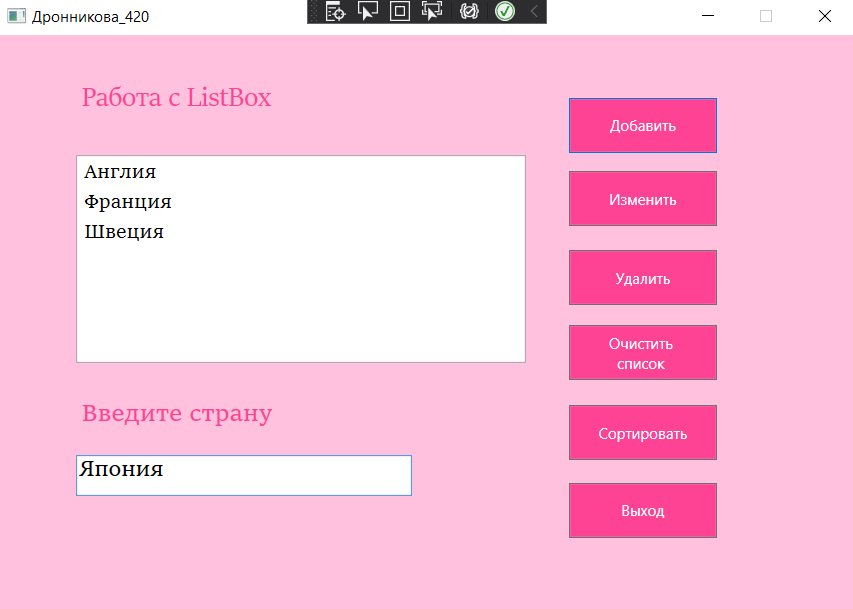
<Button Name="Del\_all" Content="Очистить &#xD;&#xA; список" HorizontalAlignment="Left" Margin="457,232,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Del\_all\_Click" Background="#FFFF4394" Foreground="White"/>

<Button Name="Sort" Content="Сортировать" HorizontalAlignment="Left" Margin="457,296,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Sort\_Click" Background="#FFFF4394" Foreground="White"/>

<Button Content="Выход" HorizontalAlignment="Left" Margin="457,358,0,0" VerticalAlignment="Top" Width="118" Height="44" Click="Button\_Click" Background="#FFFF4394" Foreground="White"/>

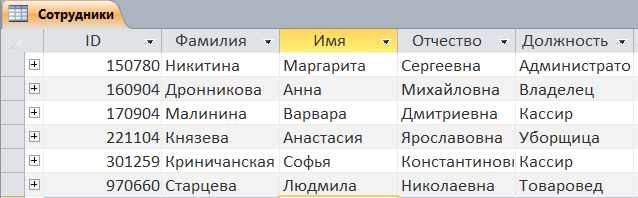
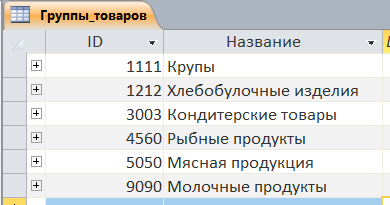
</Grid>

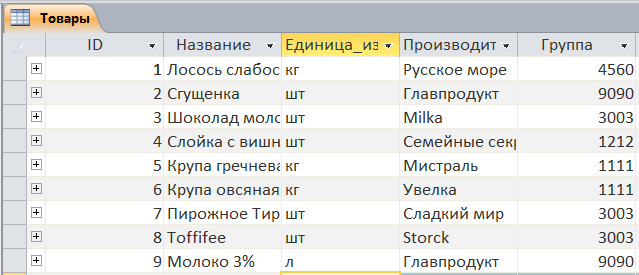
</Window>

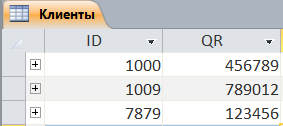


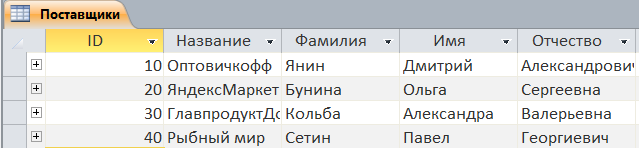
## *Задание 4*

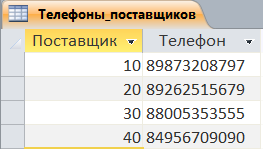
Создать базу данных «Торговое предприятие». В ней создать таблицы, которые относятся к этой базе. Установить связи между таблицами этой базы, заполнить их некоторыми данными. В отчёт вставить схемы данных и данные из таблиц.

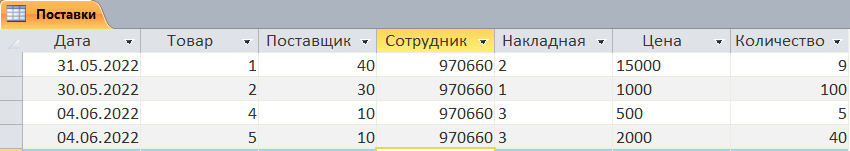


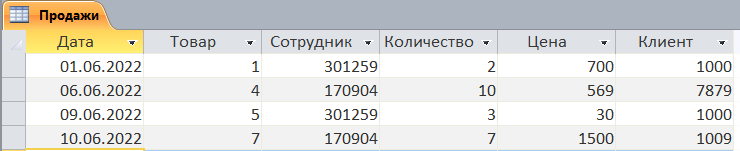


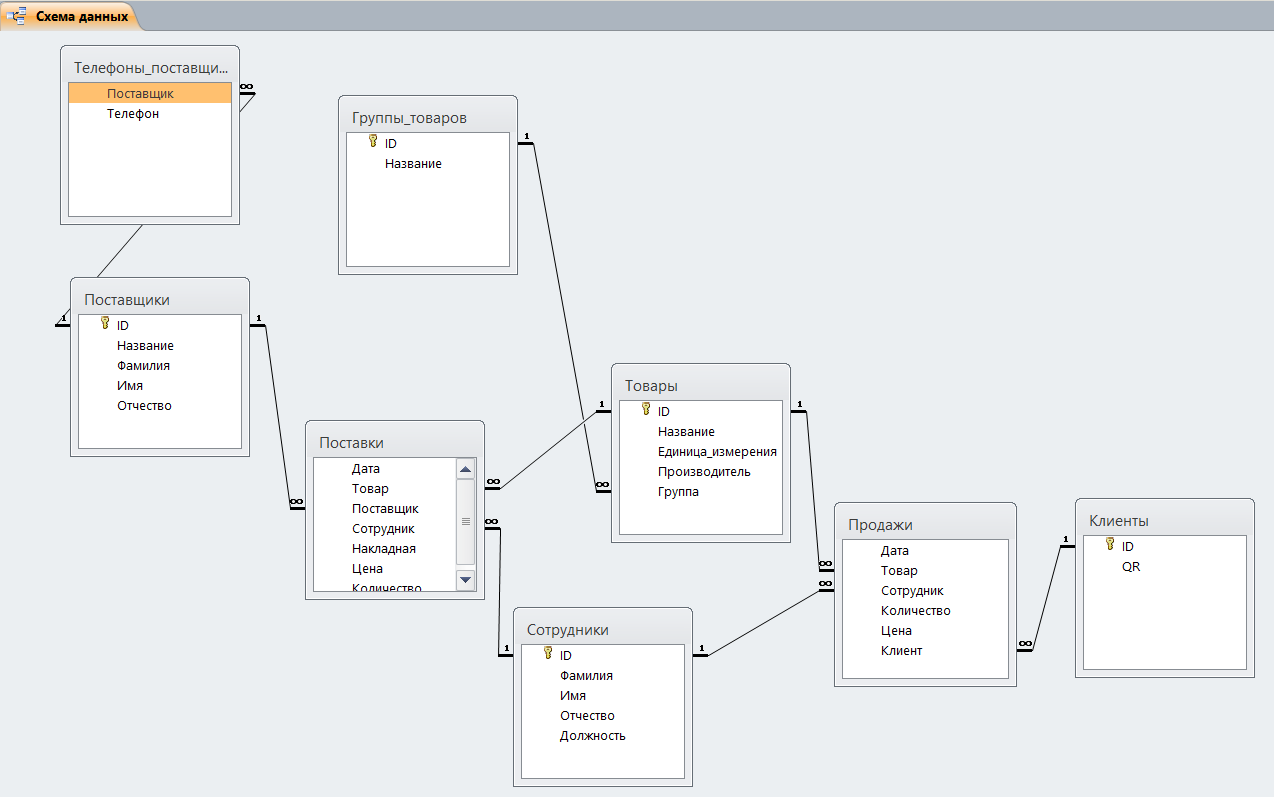












# **Практическая работа №3.**

Разработать калькулятор на языке С#. В проекте использовать прием

создания одного обработчика для нескольких событий.

Код XAML:

<Window x:Class="Anya\_help.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:Anya\_help"

mc:Ignorable="d"

Title="Дронникова ПКС420" Height="606.366" Width="482.525">

<Grid Height="573" VerticalAlignment="Top">

<Grid.Background>

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

<GradientStop Color="HotPink" Offset="0"/>

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

</LinearGradientBrush>

</Grid.Background>

<TextBox 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="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="±" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,163,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="24">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="СE" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,163,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="х²" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,223,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="x³" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,281,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="7" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,339,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="4" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,397,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="1" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,460,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="," HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="10,520,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="+" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,223,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="32">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="x" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,281,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="8" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,339,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="5" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,397,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="2" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,460,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="0" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="174,520,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="24">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="-" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,223,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="36">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="/" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,281,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="9" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,339,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26" >

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="6" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,397,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26" RenderTransformOrigin="0.492,0.521">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="3" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,460,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="26">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

<Button Content="=" HorizontalAlignment="Left" Click="Button\_Click" Height="43" Margin="337,520,0,0" VerticalAlignment="Top" Width="122" BorderBrush="White" Background="#00000000" Foreground="White" FontFamily="MS Reference Sans Serif" FontSize="26" FontWeight="Bold">

<Button.Resources>

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

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

</Style>

</Button.Resources>

</Button>

</Grid>

</Window>

Код C#:

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 Anya\_help

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private string key = "";

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

private string number = " ";

public MainWindow()

{

InitializeComponent();

calc.Foreground = Brushes.DeepPink;

}

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);

}

Изображение выглядит как текст

Автоматически созданное описание 

# **Практическая работа №4**

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 XAML:

<Window x:Class="Anya\_help.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:Anya\_help"

mc:Ignorable="d"

Title="Дронникова ПКС420" Height="450" Width="800" ResizeMode="NoResize">

<Grid Background="HotPink">

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

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

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

</TextBlock>

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

</Grid>

</Window>

Код C#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Word = Microsoft.Office.Interop.Word;

using System.Reflection;

namespace Anya\_help

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

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

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

if (textIsCorrect)

{

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

}

else

{

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

}

}

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

{

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

}

}

}

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

# **Практическая работа №5**

Необходимо разработать программу, которая при нажатии на кнопку создает

таблицу в MS Word.

<Window x:Class="Anya\_help.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:Anya\_help"

mc:Ignorable="d"

Title="Дронникова ПКС420" Height="450" Width="800" ResizeMode="NoResize" Background="HotPink" WindowStartupLocation="CenterScreen">

<Grid>

<Label Foreground="White" Content="Создание таблиц в MS Word" HorizontalAlignment="Center" Margin="0,88,0,0" VerticalAlignment="Top" FontSize="40" FontWeight="Bold" FontFamily="Times New Roman" HorizontalContentAlignment="Center" Width="800"/>

<Button Content="Создать" HorizontalAlignment="Center" VerticalAlignment="Top" Height="169" Width="468" FontSize="60" FontWeight="Bold" FontFamily="Times New Roman" Background="DeepPink" Foreground="White" BorderThickness="3" BorderBrush="Crimson" Cursor="Hand" Click="Button\_Click" Margin="0,199,0,0"/>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Word = Microsoft.Office.Interop.Word;

namespace Anya\_help

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

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

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

object objMissing = System.Reflection.Missing.Value;

object endOfDoc = "\\endofdoc";

Word.Visible = true;

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

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

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

Microsoft.Office.Interop.Word.Table table;

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

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

table.Range.ParagraphFormat.SpaceAfter = 8;

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

{

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

{

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

}

}

try

{

table.Borders.Shadow = true;

}

catch (Exception) { }

Word.Quit();

}

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

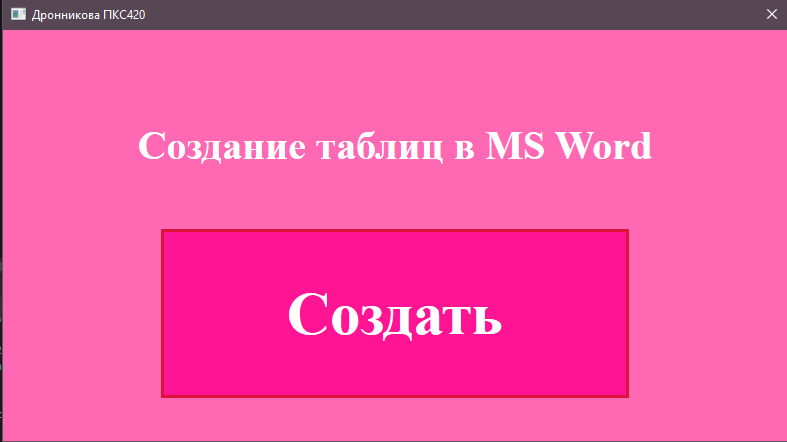
{

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

}

}

}



Изображение выглядит как текст, экран, снимок экрана

Автоматически созданное описание

# **Практическая работа №6**

Cоздайте собственные проекты с использованием функций MS

Excel не используя функцию нахождения числа Pi.

Код на XAML:

<Window x:Class="Anya\_help.func"

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:Anya\_help"

mc:Ignorable="d"

Title="Дроннкиова ПКС420" Height="316.535" Width="581.496" Background="HotPink">

<Grid Margin="0,0,2,0">

<TextBox Foreground="White" Name="ch\_func" HorizontalAlignment="Left" Height="80" Margin="55,91,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="25" Background="#00000000" BorderBrush="Black"/>

<Button x:Name="Btn\_ch\_func" Content="Корень" HorizontalAlignment="Left" Margin="223,137,0,0" VerticalAlignment="Top" Width="125" Height="54" Click="Btn\_ch\_func\_Click" Foreground="White" Background="DeepPink" BorderBrush="#FF4B0707" FontFamily="MS Reference Sans Serif" FontWeight="Bold" FontSize="20" Opacity="0.9">

<Button.Resources>

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

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

</Style>

</Button.Resources>

<Button.Effect>

<DropShadowEffect BlurRadius="15" Opacity="0.8" RenderingBias="Quality" ShadowDepth="6"/>

</Button.Effect>

</Button>

<TextBox Foreground="White" x:Name="chi\_func" HorizontalAlignment="Left" Height="80" Margin="371,177,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="147" FontSize="25" Background="#00000000" BorderBrush="Black"/>

<Border CornerRadius="10 10 10 10"

BorderThickness="5"

Margin="42,10,35,211"

Background="DeepPink"

Padding="20">

<TextBlock Text="Введите число, из которого нужно вычислить корень: "

Foreground="White"

TextAlignment="Center" FontFamily="MS Reference Sans Serif" Margin="-22,-2,-22,-22" FontSize="16" LineHeight="20" FontWeight="Bold"/>

<Border.Effect>

<DropShadowEffect/>

</Border.Effect>

</Border>

</Grid>

</Window>

Код C#:

using System.Windows;

using excel = Microsoft.Office.Interop.Excel;

namespace Anya\_help

{

/// <summary>

/// Логика взаимодействия для func.xaml

/// </summary>

public partial class func : Window

{

public func()

{

InitializeComponent();

}

private void Btn\_ch\_func\_Click(object sender, RoutedEventArgs e)

{

chi\_func.Text = "";

double num = 0;

if (!double.TryParse(ch\_func.Text.Replace('.', ','), out num))

{

ch\_func.Text = "";

MessageBox.Show("Введено не число");

return;

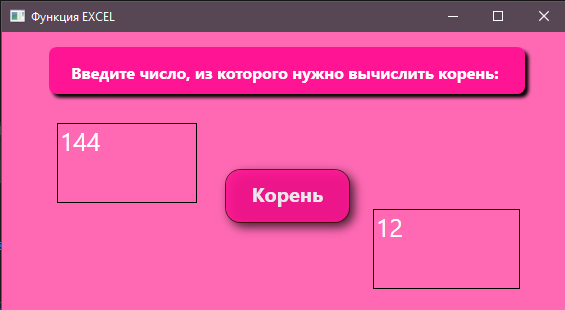
}

chi\_func.Text += new excel.Application().WorksheetFunction.ImSqrt(num);

}

}

}



# **Практическая работа №7**

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.

Код на XAML:

<Window x:Class="Anya\_help.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:Anya\_help"

mc:Ignorable="d"

Title="Дронникова ПКС420" MinHeight="350" MinWidth="715" Width="715.573" Height="350.049" MaxWidth="715" MaxHeight="350" Background="HotPink">

<Grid Margin="0,0,-9,0">

<Menu Height="25" VerticalAlignment="Top" Background="Black" Foreground="White">

<MenuItem Header="Файл" Height="25" Width="67" FontSize="18" FontFamily="MS Reference Sans Serif" Foreground="White">

<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="175" Foreground="White" FontFamily="MS Reference Sans Serif" FontSize="18"/>

<TextBox Name="input" HorizontalAlignment="Left" Height="200" Margin="10,85,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="671" FontFamily="MS Reference Sans Serif" FontSize="18"/>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.IO;

using System.Drawing;

Код C#:

namespace Anya\_help

{

/// <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();

}

}

}

}

Изображение выглядит как текст

Автоматически созданное описание