**Лабораторная работа №6**

**“Создание приложений, использующих итеративные циклические структуры”**

1. **Постановка задачи**

Вычислить приближенное значение функции 1 с точность **ε = 0.0001** с помощью рекуррентной формулы 2:

(1)

(2)

(3)

(4)

Вычисление с точностью **ε** значит, что точность числа должна быть не ниже точности числа **ε.**

**2. Формализация и уточнение задачи**

В нашем случае будет удобнее использовать рекуррентную формулу для подсчёта корня с определённой точностью. Она позволяет вычислить более точное значение функции, опираясь на предыдущий шаг.

Для выполнения задания будем использовать следующие методы:

* Метод ***Vvod()***, предназначенный для ввода данных;
* Метод ***Vivod()***, предназначенный для вывода данных;
* Метод ***Zapis\_v\_PDF()*** для выгрузки данных из таблицы в отдельный pdf-файл;
* Метод ***Func()***,который будет вычислять значение переменной с большей точностью, опираясь на предыдущий шаг;
* Событийный метод, в котором осуществляется вызов метода ввода ***Vvod()***, вызов метода ***Tub()***, вызов метода вывода ***Vivod()***;

**3. Разработка проекта**

3.1. Разработка графического интерфейса

Графический интерфейс представлен на 1-м рисунке.

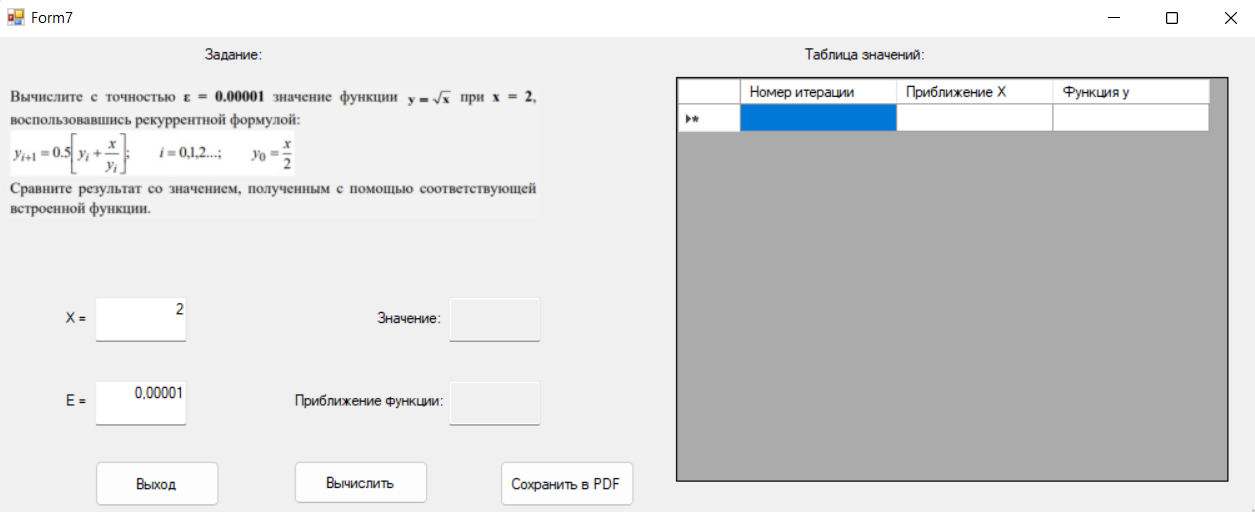


Рисунок 1 – Графический интерфейс

3.2. Разработка схемы алгоритмов

Блок-схемы событийной части проекта представлен на рисунках 2 - 3.

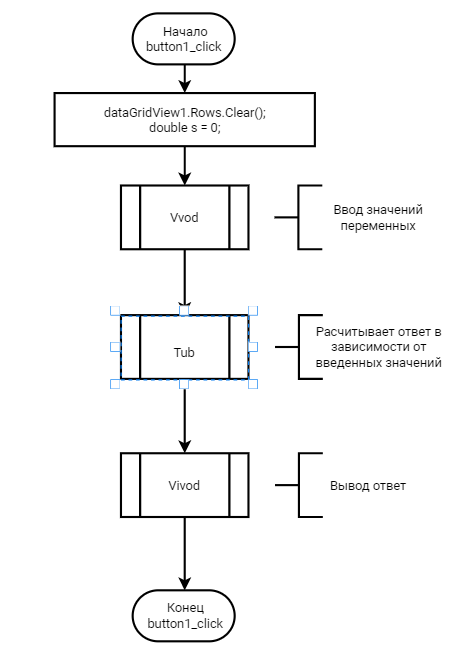


Рисунок 2 - Схема алгоритмов процедур **button\_Click1**

Проверка вводимых данных будет осуществляться по следующему алгоритму

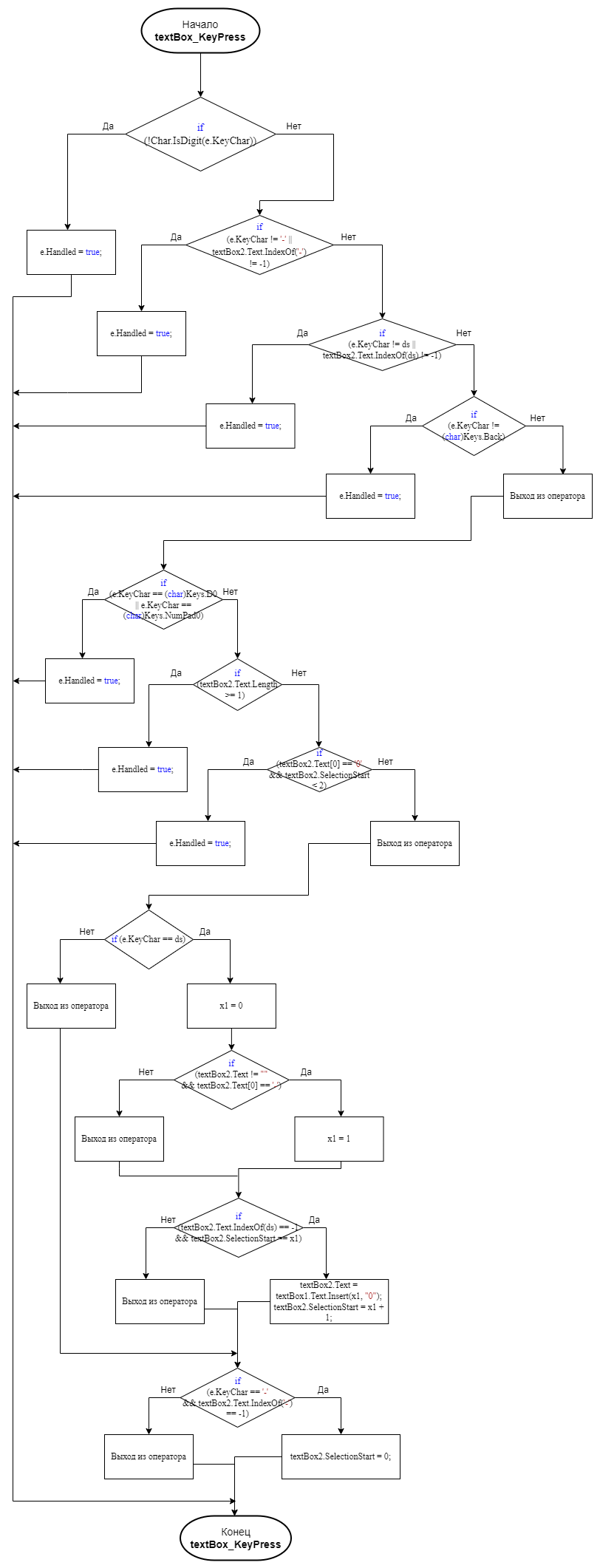


Рисунок 3 - Схема алгоритмов процедур **textbox\_KeyPress**

Блок схемы из библиотеки dll представлены ниже на рисунках 4-7

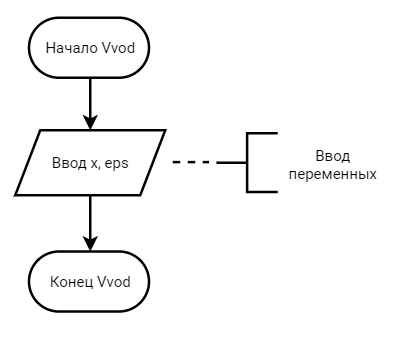


Рисунок 4 - Схема алгоритмов процедур **Vvod**

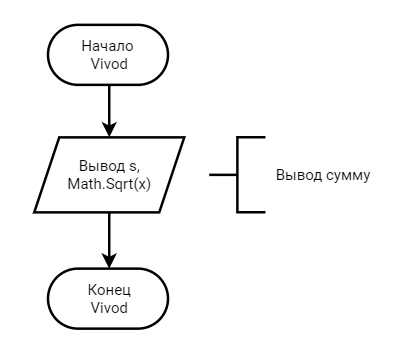


Рисунок 5 - Схема алгоритмов процедур **Vivod**

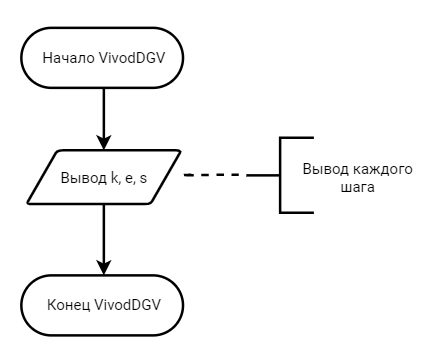


Рисунок 6 - Схема алгоритмов процедур **Vivod\_DGV**

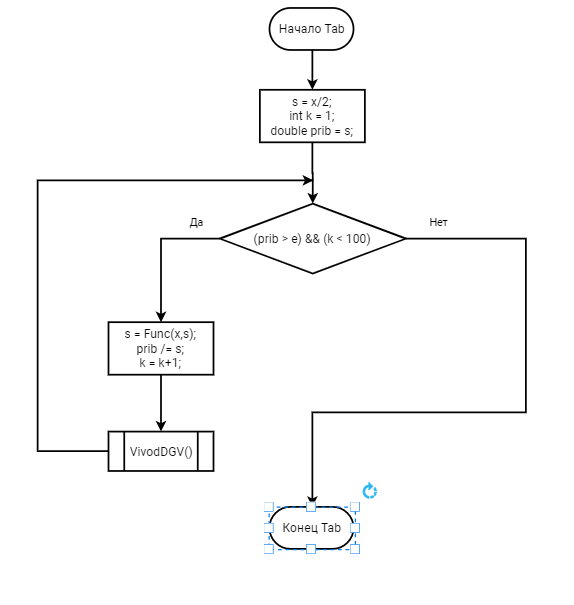


Рисунок 7 - Схема алгоритмов процедур **Tab**

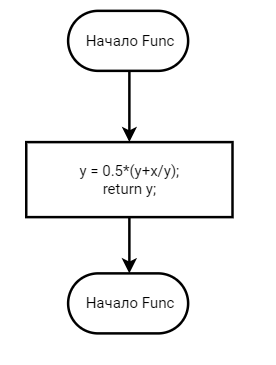


Рисунок 8 - Схема алгоритмов процедур **Func**

3.3. Написание программного кода

Основная часть кода

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Globalization;

namespace AnP\_proj

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

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

{

dataGridView1.Rows.Clear();

double s = 0;

double x = Laba\_6.Class1.Vvod(textBox1);

double eps = Laba\_6.Class1.Vvod(textBox2);

Laba\_6.Class1.Tab(x, eps, dataGridView1, ref s);

Laba\_6.Class1.Vivod(textBox3, s);

Laba\_6.Class1.Vivod(textBox4, Math.Sqrt(x));

}

private void Form7\_Load(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

Laba\_6.Class1.Zapis\_v\_PDF(dataGridView1, textBox1, textBox2, textBox3);

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

Laba\_6.Class1.Prof(textBox1, e);

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

Laba\_6.Class1.Prof(textBox2, e);

}

}

}

Часть кода, вызываемая через библиотеку Lab\_6

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Globalization;

using Microsoft.VisualBasic;

using System.Diagnostics;

using iTextSharp.text;

using iTextSharp.text.pdf;

namespace Laba\_6

{

public class Class1

{

public static double Vvod(TextBox textbox)

{

return Convert.ToDouble(textbox.Text);

}

public static void Vivod(TextBox t, double c)

{

t.Text = c.ToString("F");

}

public static double Func(double x, double y)

{

y = 0.5 \* (y + x/y);

return y;

}

public static void Tab(double x, double e, DataGridView dgv, ref double s)

{

int k = 1;

s = x / 2;

double prib = s;

//for(int i = 1; i <= 100; i++)

while ((prib >= e) && (k < 100))

{

s = Func(x, s);

prib /= s;

VivodDGV(k, prib, s, dgv);

k = k + 1;

}

}

public static void VivodDGV(int k, double e, double s, DataGridView DGV1)

{

DGV1.Rows.Add(k.ToString("F1"), e.ToString("F5"), s.ToString("F10"));

}

public static void Zapis\_v\_PDF(DataGridView grid, TextBox t1, TextBox t2, TextBox t3)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.ShowDialog();

string filename = sfd.FileName + ".pdf";

try

{

var Document = new Document();

var Zap = PdfWriter.GetInstance(Document, new FileStream(filename, FileMode.Create));

Document.Open();

string ttf = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Fonts), "ARIAL.TTF");

var baseFont = BaseFont.CreateFont(ttf, BaseFont.IDENTITY\_H, BaseFont.NOT\_EMBEDDED);

var font = new Font(baseFont, Font.DEFAULTSIZE, Font.NORMAL);

var tabl = new PdfPTable(2);

var cell = new PdfPCell();

cell.FixedHeight = 16.0F;

cell.BackgroundColor = Color.WHITE;

cell.Colspan = 2;

cell.Border = 10;

cell.HorizontalAlignment = Element.ALIGN\_LEFT;

cell.Phrase = new Phrase("Исходные данные:", font);

tabl.AddCell(cell);

cell.Phrase = new Phrase("X=" + t1.Text + " E=" + t2.Text, font);

tabl.AddCell(cell);

cell.Phrase = new Phrase("Результаты:", font);

tabl.AddCell(cell);

cell.BackgroundColor = Color.LIGHT\_GRAY;

cell.Colspan = 1;

cell.Border = 10;

cell.Phrase = new Phrase("Номер итерации", font);

tabl.AddCell(cell);

cell.Phrase = new Phrase("Элемент ряда", font);

tabl.AddCell(cell);

for (int i = 0; i < grid.RowCount; i++)

{

cell.HorizontalAlignment = Element.ALIGN\_RIGHT;

cell.Phrase = new Phrase((string)grid[0, i].Value, font);

tabl.AddCell(cell);

cell.Phrase = new Phrase((string)grid[1, i].Value, font);

tabl.AddCell(cell);

}

cell.BackgroundColor = Color.WHITE;

cell.Colspan = 2;

cell.Border = 10;

cell.HorizontalAlignment = Element.ALIGN\_LEFT;

cell.Phrase = new Phrase("Сумма ряда:" + t3.Text, font);

tabl.AddCell(cell);

tabl.TotalWidth = Document.PageSize.Width - 200;

tabl.WriteSelectedRows(0, -1, 40, Document.PageSize.Height - 30, Zap.DirectContent);

Document.Close();

Zap.Close();

DialogResult result =

MessageBox.Show("Запись успешна." + "\nОткрыть файл?", "Запись", MessageBoxButtons.YesNo,

MessageBoxIcon.Information);

if (result == DialogResult.Yes)

Process.Start(filename);

}

catch (Exception e)

{

MessageBox.Show("Что-то пошло не так " + e.Message, "Ошибка", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

}

}

public static void Prof(TextBox textBox1, KeyPressEventArgs e)

{

int x1;

char ds = (char)NumberFormatInfo.CurrentInfo.NumberDecimalSeparator[0];

// Если нажатая клавиша не цифровая.

if (!Char.IsDigit(e.KeyChar))

// Запрет на ввод более одного знака минуса.

if (e.KeyChar != '-' || textBox1.Text.IndexOf('-') != -1)

// Запрет на ввод более одного десятичного разделителя.

if (e.KeyChar != ds || textBox1.Text.IndexOf(ds) != -1)

// Если нажатая клавиша не является клавишей BackSpace.

if (e.KeyChar != (char)Keys.Back)

e.Handled = true; // Запрет ввода

// Запрет ввода повторных нулей в начале числа.

if (e.KeyChar == (char)Keys.D0 || e.KeyChar == (char)Keys.NumPad0)

if (textBox1.Text.Length >= 1) // Если строка не пустая

if (textBox1.Text[0] == '0' && textBox1.SelectionStart < 2)

e.Handled = true; // Запрет ввода

//-незначащий ноль

if (Char.IsDigit(e.KeyChar))

if (textBox1.Text.Length == 1 && textBox1.Text[0] == '0')

textBox1.Paste(",");

// Замена десятичного разделителя в начале числа на "0,".

if (e.KeyChar == ds)

{ // Если введен десятичный разделитель

x1 = 0;

if (textBox1.Text != "" && textBox1.Text[0] == '-') { x1 = 1; }

if (textBox1.Text.IndexOf(ds) == -1 && textBox1.SelectionStart == x1)

{

// Если десятичный разделитель уже есть, то 0 не вставлять.

textBox1.Text = textBox1.Text.Insert(x1, "0"); // Вставка 0

textBox1.SelectionStart = x1 + 1; // Текстовый курсор в конец

}

// Перенос точки ввода минуса в начало (ввод минуса только спереди)

}

if (e.KeyChar == '-' && textBox1.Text.IndexOf('-') == -1)

textBox1.SelectionStart = 0; // Текстовый курсор в начало

//-незначащий ноль

}

}

}

**4. Результат выполнения проекта**

Результат выполнения программы продемонстрирован на рисунке 9.

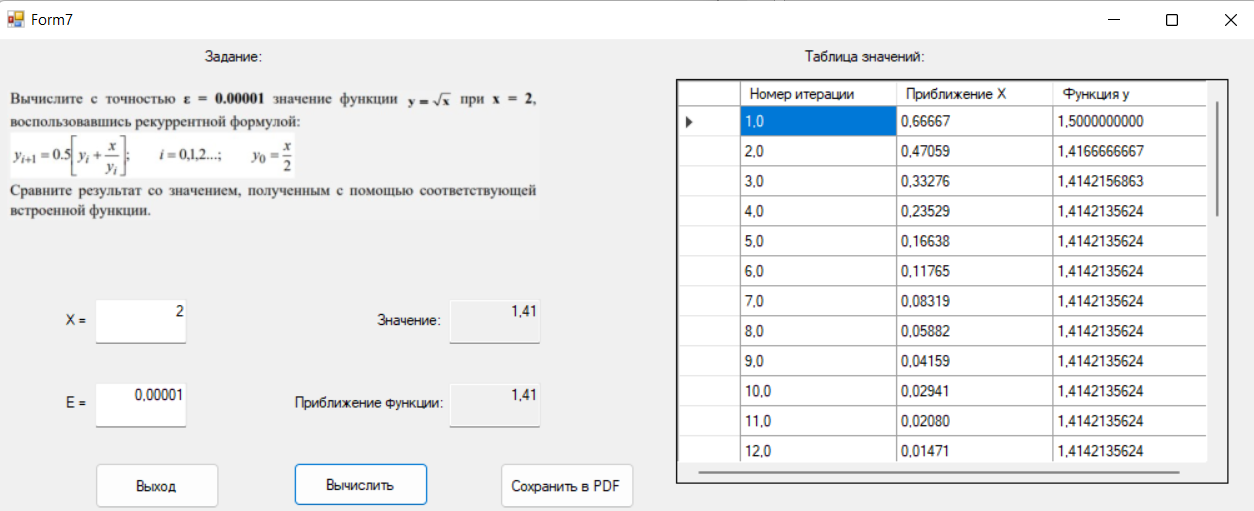


Рисунок 9 - Результат выполнения проекта

Также по заданию требовалось выгрузить данные из таблицы в pdf-файл. Результат представлен на рисунке 10.

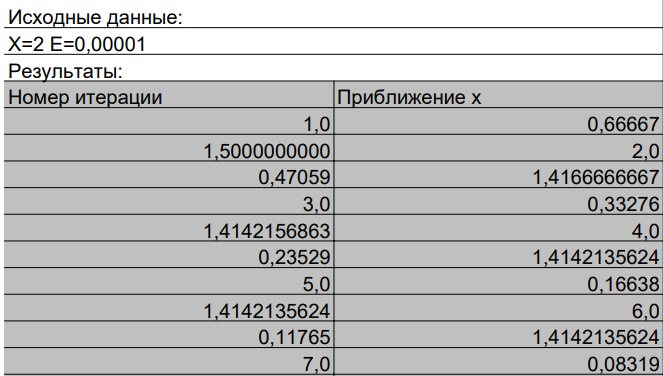


Рисунок 10 – Таблица значений