# Содержание

Содержание 3

Введение 4

1 Математическая формулировка задачи 5

2 Описание программы 6

2.1. Структурная схема программы 6

2.2. Описание разработанного класса 7

2.3. Описание использующейся COM-технологии 7

2.4 Основные возможности программы 7

2.5 Средства использования сервисов, предоставляемых Microsoft Office 8

3 Руководство пользователя 9

4 Методика испытаний 12

Заключение 13

Литература 14

Приложение 15

# Введение

Цель данной работы продемонстрировать навыки, полученные в рамках курса «Разработка приложений в визуальных средах» на примере программы, реализующей построение фрактальной кривой – «Кривая Дракона».

Для облегчения моделирования используются программные средства наподобие данной программы. Она позволяет справиться с решением таких задач, поддерживая такие функции как построение кривой и отправка

результатов в офисные приложения.

Для создания приложения была выбрана среда разработки Microsoft Visual Studio, так как она позволяет быстро и эффективно разрабатывать сложные программные продукты.

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

В Windows Forms можно разрабатывать графически сложные приложения, которые просто развертывать, обновлять, и с которыми удобно работать как в автономном режиме, так и в сети. Приложения Windows Forms могут получать доступ к локальному оборудованию и файловой системе компьютера, на котором работает приложение.

Все разделы содержат необходимую и достаточную информацию по использованию данного программного продукта, подробное описание программы и вычислительной системы необходимой для функционирования программного продукта.

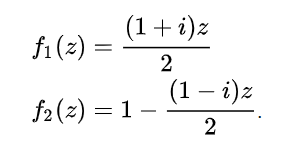
# 1 Математическая формулировка задачи

В рамках поставленной задачи должна быть разработана программа для построения «Кривой Дракона».

Кривая дракона — общее название для некоторых фрактальных кривых, которые могут быть аппроксимированы рекурсивными методами, такими как L-системы.

Фрактал может быть записан как L-система с параметрами:

* угол равен 90° или
* начальная строка — *FX*
* правила преобразования строк:
  + *X* {\displaystyle \mapsto }→ *X+YF+*
  + *Y* {\displaystyle \mapsto }→ *-FX-Y*

Кроме того, фрактал может быть описан системой итерируемых функций на комплексной плоскости:

{\displaystyle f\_{1}(z)={\frac {(1+i)z}{2}}}

{\displaystyle f\_{2}(z)=1-{\frac {(1-i)z}{2}}}.

# 2 Описание программы

# 2.1. Структурная схема программы

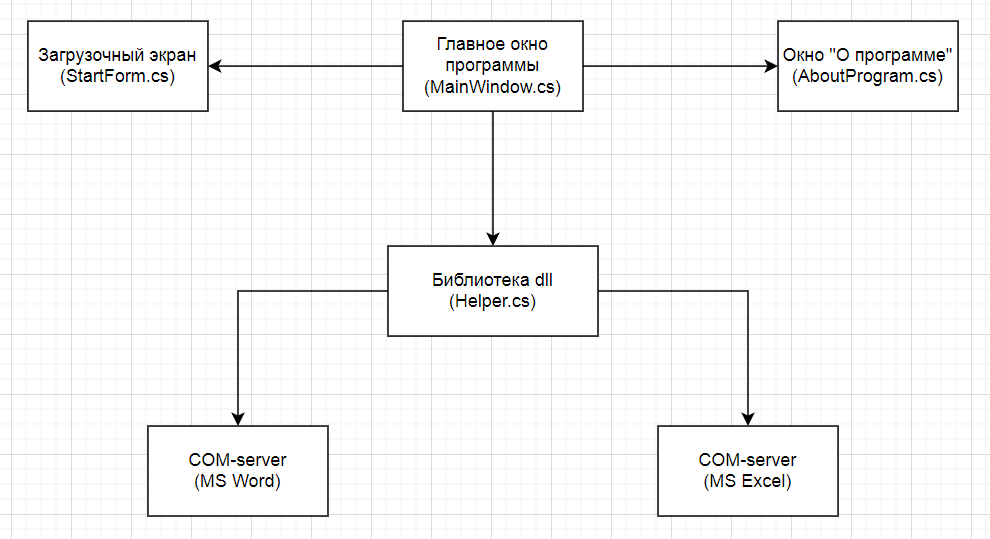
 Структурно программа состоит из главного модуля, модулей описания вспомогательных форм, динамических библиотек, файлов помощи и обращений к другим программам показана на рисунке 2.1

Рисунок 2.1 - Связь модулей программы

# 2.2. Описание разработанного класса

Был разработан базовый класс для общего функционирования программы.

**private void exitToolScripMenuItem\_Click()** /функция выхода из программы

**private void Draw()** /функция отрисовки фрактала

**private void Dragon()** /рекурсивная функция расчетов

**private void saveToolScripMenuItem\_Click()** /функция сохранения результатов

**private void wordToolStripMenuItem1\_Click()** /функция для открытия MS Word

**private void excelToolStripMenuItem1\_Click()** /функция для открытия MS Excel

# 2.3 Основные возможности программы

Программа начинается с отображения оригинальной заставки, а затем на экране появляется главное окно программы (MainWindow.cs). Главный модуль управляет работой всех других модулей в соответствии с запросами пользователя.

Он содержит:

* Вызов окна «О программе»;
* Передачу данных в Word и Excel;
* Открытие полученных отчётов;
* Вызов функции отрисовки кривой;
* Вызов проверки актуальности версий;
* Вызов окна помощи;

Диалог с пользователем поддерживается с помощью главного и выпадающего меню и кнопок. Окно «О программе» содержит краткую информацию о приложении. При выполнении расчетов пользователю предоставляется возможность выбора сложности отрисовки фрактала.

Процесс логического взаимодействия пользователя c программой, назначение элементов главного окна описаны в мануале, который можно вызывать при нажатии на кнопки «Help».

Вопросы непосредственной программной реализации конкретных модулей приведены в приложении.

# 2.4 Средства использования сервисов, предоставляемых Microsoft Office

Функция главной программы позволяет сохранить полученное изображение кривой, а также сложность отрисовки.

**private void saveToolStripMenuItem\_Click()**

Для автоматического запуска MS Word и MS Excel с готовыми отчетами, используются функции библиотеки dll:

**public void openWord()**

**public void openExcel()**

# 3 Руководство пользователя

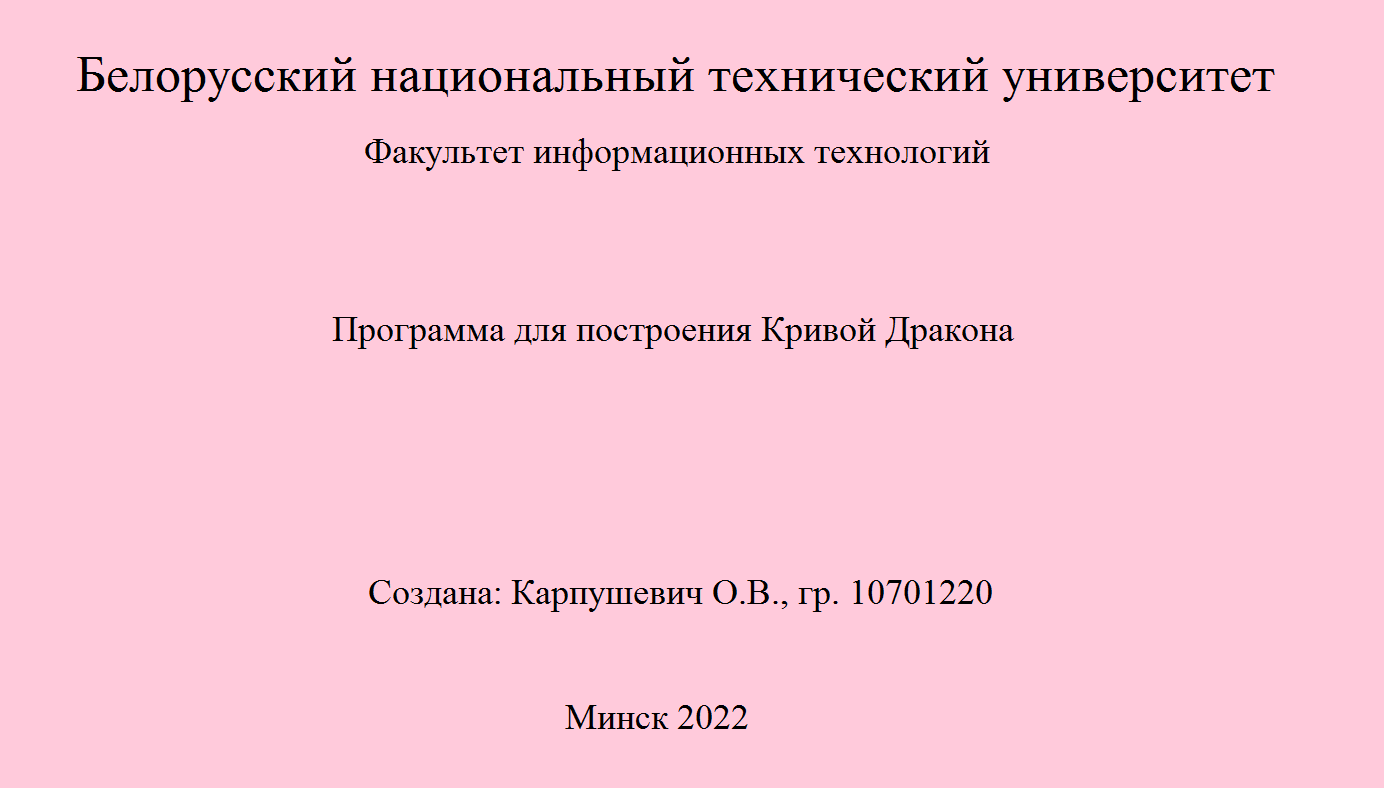
Для выполнения программы необходимо вызвать файл DragonCurve.exe. Появится окно-заставка, приведённое на рисунке 3.1

Рисунок 3.1 - Окно-заставка

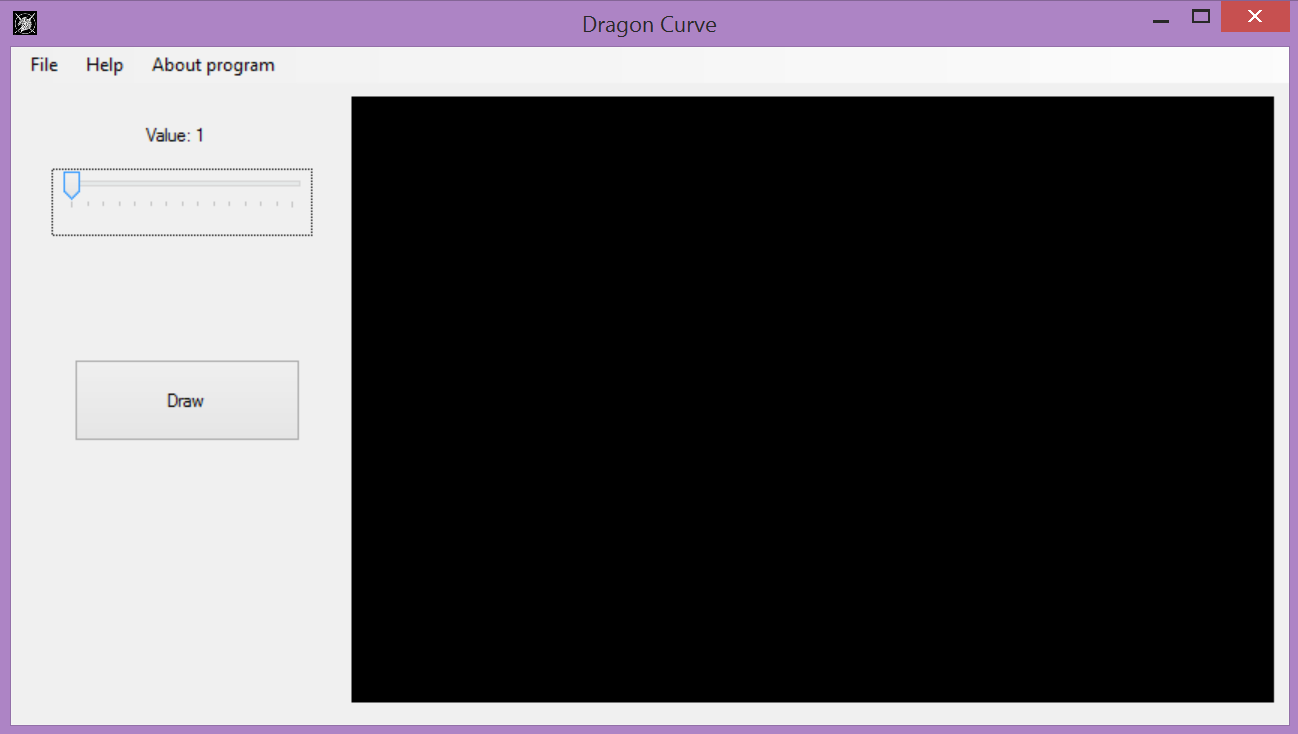
Через 3 секунды появится основное окно программы, рисунок 3.2

Рисунок 3.2 - Главное окно программы

Как видно, данное окно состоит из следующих частей:

* Главного меню;
* Области выбора сложности;
* Области вывода информации;
* Управляющая кнопка;

Пункты главного меню показаны на рисунке 3.3



Рисунок 3.3 - Пункты главного меню

Некоторые из пунктов содержат подпункты. Так, пункт «File», показанный на рисунке 3.4, раскладывается на подпункты «Save», «Open», «Exit». «Open» в свою очередь раскладывается на «Word» и «Excel». Пункт «Save» выполняет сохранение результатов в MS Word и MS Excel.

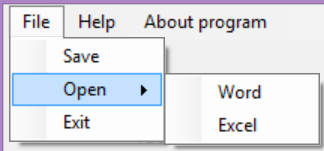


Рисунок 3.4 - Общий вид подпунктов «Файл»

Кнопка «Open» производит открытие полученных документов, а пункт «Exit» завершает работу программы.

Пункт «About Program», рисунок 3.5, подразделяется на «New Version», «This version» и «About program Dragon Curve». Пункт «New Version» выводит окно с информацией о новых версиях программы. «This version» вызывает окно с номером данной версией.

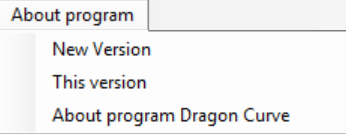


Рисунок 3.5 – Выпадающее меню

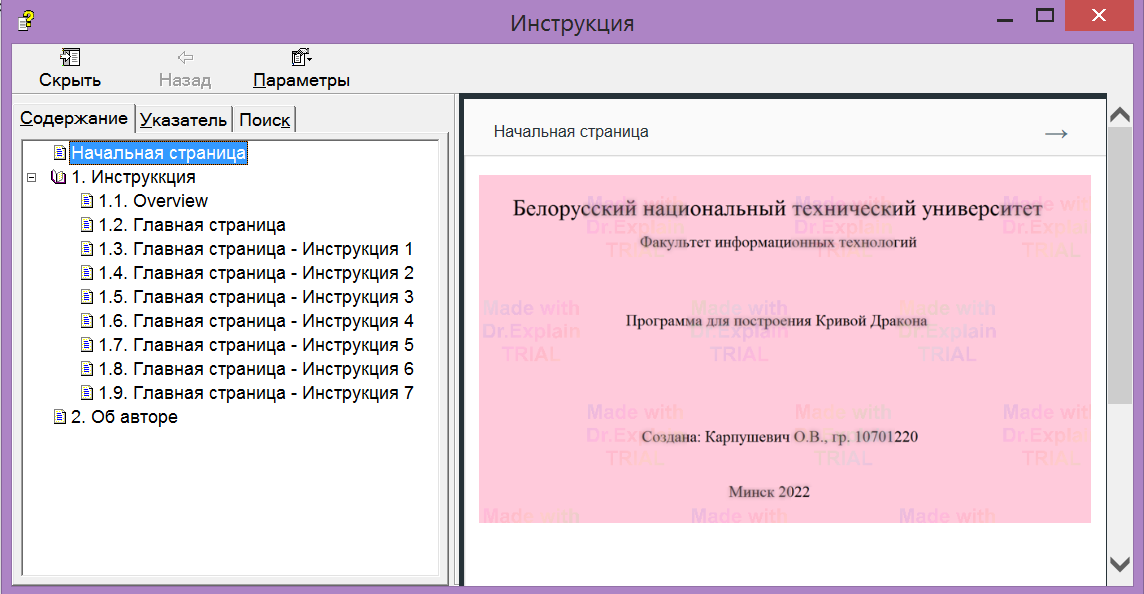
Пункт «Help», демонстрируемый на рисунке 3.7, обеспечивает вызов справки для пользователя.

Рисунок 3.7 - Справка

# 4 Методика испытаний

Целью проведения испытаний является проверка работоспособности (надежности) программы при различных условиях ее функционирования. Программа должна обеспечивать корректность отрисовки при заданной сложности данных, а также корректный вывод в MS Word и MS Excel.

Для демонстрации работоспособности программы необходимо провести ряд испытаний с различными начальными условиями. Для обеспечения нормальной работы программы требуется наличие соответствующих DLL, а также приложений Word, Excel. Тестовые примеры выполнялись в среде операционной системы Windows 8.1. Professional при использовании процессора Intel Core i5-4310U 2 2.6 GHz и разрешении экрана 1366\*768. Для работы с программой необходимо также наличие клавиатуры и манипулятора типа «мышь».

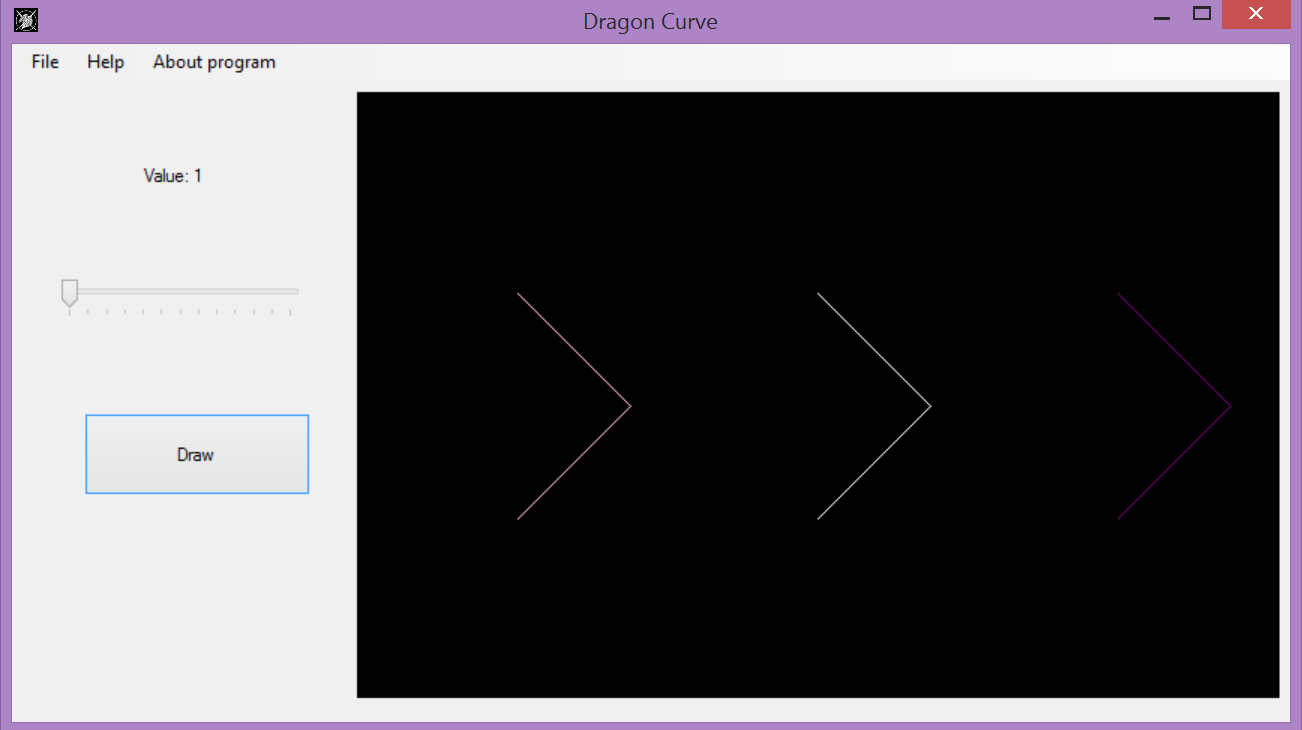
При начальных условиях, представленных на форме, получаем результат как на рисунке 4.1.

Рисунок 4.1 – Начальные результаты

Таким образом, проведенное тестирование программы не выявило сбойных ситуаций и некорректностей в ее работе. Следует считать, что в целом программа протестирована, отвечает поставленным требованиям и вполне работоспособна.

# Заключение

В ходе выполнения курсовой работы были закреплены знания по курсу «Разработка приложений в визуальных средах»; изучены принципы создания динамических библиотек; освоены методы передачи данных между приложениями; изучены возможности создания справочной системой высокой степени сложности и различных форматов.

Программа имеет достаточно ясный и понятый дружелюбный пользовательский интерфейс, обеспечивающий удобство в работе и обеспечивающий получение необходимой пользователю информации с наименьшими затратами времени.

# Литература

1. Гурский Н.Н. Курс лекций по предмету «Разработка приложений в визуальных средах»/ Н.Н. Гурский
2. Руководство по программированию в Window Forms [Электронный ресурс]: Сборник тематических уроков - Режим доступа: <https://metanit.com/sharp/windowsforms/>
3. Классы, структуры и пространства имён [Электронный ресурс]: Создание библиотек классов - Режим доступа: https://metanit.com/sharp/tutorial/3.46.php

# Приложение

***Файл проекта DragonCurve MainWindow.cs***

**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** Xceed.Document.NET;

**using** Xceed.Words.NET;

**using** Image = Xceed.Document.NET.Image;

**using** Excel = Microsoft.Office.Interop.Excel;

**using** MyLib;

**namespace** DragonCurve

{

**public** **partial** **class** MainWindow : Form

{

**private** **string** path = @"C:\univer\2 курс\2 сем\рпвс\курсовая\DragonCurve\DragonCurve\Resources\";

Graphics gr;

Helper hp = **new** Helper();

**public** MainWindow()

{

InitializeComponent();

**this**.KeyPreview = **true**;

**this**.KeyUp += **new** System.Windows.Forms.KeyEventHandler(**this**.MainWindow\_KeyUp);

}

**private** **void** exitToolStripMenuItem\_Click(**object** sender, EventArgs e)

{

MessageBoxButtons msb = MessageBoxButtons.YesNo;

String message = "Вы действительно хотите выйти?";

String caption = "Выход";

**if** (MessageBox.Show(message, caption, msb) == DialogResult.Yes)

**this**.Close();

}

**private** **void** valueBar\_Scroll(**object** sender, EventArgs e)

{

valueLable.Text = "Value: " + valueBar.Value.ToString();

}

**private** **void** drawBt\_Click(**object** sender, EventArgs e)

{

Pen Pen1 = **new** Pen(Color.White, 1);

Pen Pen2 = **new** Pen(Color.Pink, 1);

Pen Pen3 = **new** Pen(Color.Purple, 1);

gr = pictureBox.CreateGraphics();

gr.Clear(Color.Black);

Draw(pictureBox.Width, pictureBox.Height, gr, Pen1);

Draw(pictureBox.Width - 400, pictureBox.Height, gr, Pen2);

Draw(pictureBox.Width + 400, pictureBox.Height, gr, Pen3);

}

**private** **void** Draw(**int** w, **int** h, Graphics gr, Pen pen)

{

**int** iterarions = (**int**)valueBar.Value;

**int** size = 150;

PointF[] points =

{

**new** PointF(w/2, h/3),

**new** PointF(w/2, h/3 + size)

};

Dragon(iterarions, points, gr, pen);

}

**private** **void** Dragon(**int** iter, PointF[] points, Graphics g, Pen pen)

{

**float** dx, dy;

**if** (iter == 0)

g.DrawLine(pen, points[0], points[1]);

**else**

{

dx = (points[1].X + points[0].X) / 2 + (points[1].Y - points[0].Y) / 2;

dy = (points[1].Y + points[0].Y) / 2 - (points[1].X - points[0].X) / 2;

PointF[] point1 =

{

**new** PointF(points[0].X, points[0].Y),

**new** PointF(dx, dy)

};

PointF[] point2 =

{

**new** PointF(points[1].X, points[1].Y),

**new** PointF(dx, dy)

};

Dragon(iter - 1, point1, g, pen);

Dragon(iter - 1, point2, g, pen);

}

}

**private** **void** wordToolStripMenuItem1\_Click(**object** sender, EventArgs e)

{

**try**

{

hp.openWord();

}

**catch** (Exception exp)

{

MessageBox.Show(exp.Message);

}

}

**private** **void** excelToolStripMenuItem1\_Click(**object** sender, EventArgs e)

{

**try**

{

hp.openExcel();

}

**catch** (Exception exp)

{

MessageBox.Show(exp.Message);

}

}

**private** **void** MainWindow\_KeyUp(**object** sender, KeyEventArgs e)

{

**if** (e.KeyCode == Keys.Enter) drawBt.PerformClick();

}

**private** **void** helpToolStripMenuItem\_Click(**object** sender, EventArgs e)

{

**try**

{

System.Diagnostics.Process.Start(path + "HelpManual.chm");

} **catch** (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

**private** **void** newVersionToolStripMenuItem\_Click(**object** sender, EventArgs e)

{

MessageBox.Show("You have latest version");

}

**private** **void** thisVersionToolStripMenuItem\_Click(**object** sender, EventArgs e)

{

MessageBox.Show("version v.1.2.0");

}

**private** **void** saveToolStripMenuItem\_Click(**object** sender, EventArgs e)

{

Rectangle r = **new** Rectangle(670, 240, 910, 600);

Bitmap image = **new** Bitmap(910, 600);

Graphics g = Graphics.FromImage(image);

g.CopyFromScreen(r.Location, **new** Point(0, 0), r.Size);

image.Save(path + "DragonCurve.jpg", System.Drawing.Imaging.ImageFormat.Jpeg);

*//сохранение в Word*

**string** imageDoc = path + "DragonCurve.jpg";

**string** doc = path + "DragonCurveWord.docx";

DocX document = DocX.Create(doc);*//создаем документ*

document.InsertParagraph("Сложность = " + (**int**)valueBar.Value).Font("Times New Roman").

FontSize(16).

Color(Color.Purple);

Paragraph paragraph = document.InsertParagraph();

Image imageSave = document.AddImage(imageDoc);

**var** t = (**int**)valueBar.Value;

**var** p = path;

Paragraph paragraph2 = document.InsertParagraph();

paragraph2.AppendPicture(imageSave.CreatePicture()); *// вставка изображения в параграф*

paragraph2.Alignment = Alignment.left; *// по центру*

document.Save();

**string** excelSave = path + "Excel.xls";

Excel.Application excel = **new** Excel.Application(); *//объявляем приложение*

Excel.Workbook workBook; *// Создаём экземпляр рабочий книги Excel*

Excel.Worksheet workSheet;*// Создаём экземпляр листа Excel*

workBook = excel.Workbooks.Add();

workSheet = (Excel.Worksheet)workBook.Worksheets.get\_Item(1);

workSheet.Cells[1, 1] = "Cложность";

workSheet.Cells[2, 1] = (**int**)valueBar.Value;

workSheet.Shapes.AddPicture(imageDoc, Microsoft.Office.Core.MsoTriState.msoFalse,

Microsoft.Office.Core.MsoTriState.msoCTrue, 0, 30, 350, 250);

excel.DefaultFilePath = excelSave;

excel.DisplayAlerts = **false**;

workBook.SaveAs(excelSave, Excel.XlFileFormat.xlWorkbookNormal,

Type.Missing,

Type.Missing,

Type.Missing,

Type.Missing,

Excel.XlSaveAsAccessMode.xlExclusive,

Type.Missing,

Type.Missing,

Type.Missing,

Type.Missing,

Type.Missing);

excel.Quit();

}

**private** **void** aboutProgramDragonCurveToolStripMenuItem\_Click(**object** sender, EventArgs e)

{

AboutProgram pr = **new** AboutProgram();

pr.Show();

}

}

}

***Файл проекта Program.cs***

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Threading.Tasks;

**using** System.Windows.Forms;

**namespace** DragonCurve

{

**static** **class** Program

{

*/// <summary>*

*/// Главная точка входа для приложения.*

*/// </summary>*

[STAThread]

**static** **void** Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(**false**);

StartForm first = **new** StartForm();

DateTime end = DateTime.Now + TimeSpan.FromSeconds(3);

first.Show();

**while**(end > DateTime.Now)

{

Application.DoEvents();

}

first.Close();

first.Dispose();

Application.Run(**new** MainWindow());

}

}

}

***Класс dll библиотеки Helper.cs***

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**namespace** MyLib

{

**public** **class** Helper

{

**private** **string** path = @"C:\univer\2 курс\2 сем\рпвс\курсовая\DragonCurve\DragonCurve\Resources\";

**public** **void** openWord()

{

System.Diagnostics.Process.Start(path + "DragonCurveWord.docx");

}

**public** **void** openExcel()

{

System.Diagnostics.Process.Start(path + "Excel.xls");

}

}