МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования

**«Северный (Арктический) федеральный университет имени М.В. Ломоносова»**

Высшая школа информационных технологий и автоматизированных систем

(наименование высшей школы / филиала / института / колледжа)

привет от автора)

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

|  |  |  |
| --- | --- | --- |
| По дисциплине | | Технология разработки программного обеспечения |
|  | |  |
|  | |  |
| По теме | Разработка приложения для построения фракталов | |
|  | | |

|  |
| --- |
| Выполнил обучающийся:  Анащенко Артем Альбертович |
| (Ф.И.О.) |
| Направление подготовки:  09.03.02 Информационные системы и технологии |
| (код и наименование) |
| Курс: 3 |
| Группа: 351818 |
| Руководитель:  Маломан Ю.С., ст. преподаватель |
| (Ф.И.О. руководителя, должность / уч. степень / звание) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отметка о зачете |  |  |  |  |
|  |  | (отметка прописью) |  | (дата) |
| Руководитель |  |  |  | Ю.С. Маломан |
|  |  | (подпись руководителя) |  | (инициалы, фамилия) |

Архангельск 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное автономное образовательное учреждение высшего образования   
**«Северный (Арктический) федеральный университет имени М.В. Ломоносова»**

Высшая школа информационных технологий и автоматизированных систем

**ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЫ**

**По дисциплине Технология разработки программного обеспечения**

Студенту высшей школы информационных технологий и автоматизированных систем 3 курса 351818 группы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Анащенко Артему Альбертовичу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О. студента)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_09.03.02 Информационные системы и технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки)

**На тему: Разработка приложения для построения фракталов**

**Задание**

1. По выбранной теме разработать приложение, провести комментирование, форматирование и рефакторинг кода
2. Оформить пояснительную записку по стандарту СТО.

**Структура пояснительной записки**Титульный лист   
Задание   
Лист для замечаний   
ОглавлениеВведение

Сбор и анализ требований

Проектирование

Разработка

Тестирование и отладка

Документирование

Заключение

Приложение (при необходимости)

**Срок выполнения:** с 1 ноября 20 20 г. по 1 декабря 20 20 г.

**Руководитель работы** ст.преподаватель Ю.С. Маломан

Архангельск 2020

## ЛИСТ ДЛЯ ЗАМЕЧАНИЙ

## ОГЛАВЛЕНИЕ

[Нормативные ссылки 5](#_Toc57598549)

[Введение 6](#_Toc57598550)

[1 Сбор и анализ требований 7](#_Toc57598551)

[2 Проектирование 8](#_Toc57598552)

[3 Разработка 10](#_Toc57598553)

[4 Тестирование и отладка 12](#_Toc57598554)

[5 Документирование 14](#_Toc57598555)

[5.1 Руководство пользователя. 14](#_Toc57598556)

[5.1 Документирование кода 17](#_Toc57598557)

[Заключение 18](#_Toc57598558)

[Приложение А (обязательное) Файл «MainForm.cs» 19](#_Toc57598559)

[Приложение Б (обязательное) Файл «FractalsFunctions.cs» 22](#_Toc57598560)

## НОРМАТИВНЫЕ ССЫЛКИ

Настоящий текстовый документ оформлен в соответствии со следующим нормативным документом:

СТО 60–02.2.3–2018 федерального государственного автономного образовательного учреждения высшего образования «Северный (Арктический) федеральный университет имени М.В. Ломоносова».

## Введение

Было необходимо разработать приложение, которое позволяло бы строить и получать графические изображения фракталов. Изображение некоторых типов фракталов, например, фрактальных деревьев, могут использоваться в качестве текстур растительности в различных областях, где применяется графика, например, в играх, тем самым автоматизируя часть труда художника, освобождая его от обязанности рисовать большое количество непохожих друг на друга деревьев и кустов.

# 1 СБОР И АНАЛИЗ ТРЕБОВАНИЙ

Программное обеспечение должно быть простым в использовании, интуитивно понятным, а также использовать минимальное время пользователя для обучения использованию программы. На рынке имеется некоторое количество программ, позволяющих создавать фрактальные изображения, однако большинство из них требуют большого количества времени на обучение использованию, как, например, Ultra Fractal (рисунок 1). Некоторые программы не обладают функционалом для сохранения получившегося изображения, а также изменения каких-либо параметров при построении фрактала, например BinaryTree (рисунок 2).

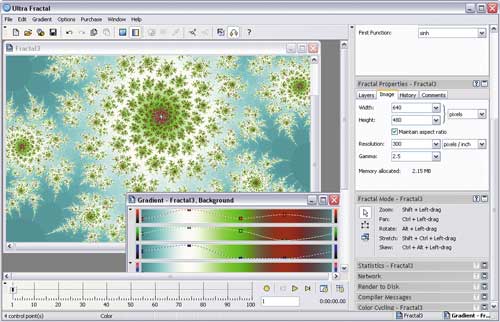


Рисунок 1 – Программа Ultra Fracrtal

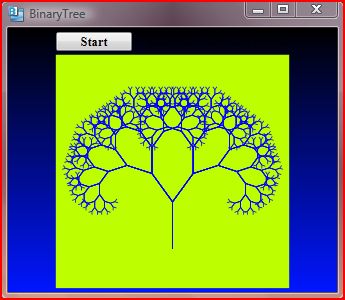


Рисунок 2 – Программа BinaryTree

# 2 ПРОЕКТИРОВАНИЕ

Формат программного обеспечения был выбран как десктопное приложение. Программе не требуется подключение и наличие базы данных, а также подключение к интернету. После выхода программы в эксплуатацию планируется выпускать обновления, где будет расширяться список доступных для построения типов фракталов. Файл обновления будет загружаться пользователем с сайта разработчика при необходимости.

Была создана диаграмма вариантов использования (рисунок 3). Действующим лицом является пользователь, который в свою очередь может выбрать тип создаваемого фрактала, задать параметры, сгенерировать фрактал и сохранить получившееся изображение.

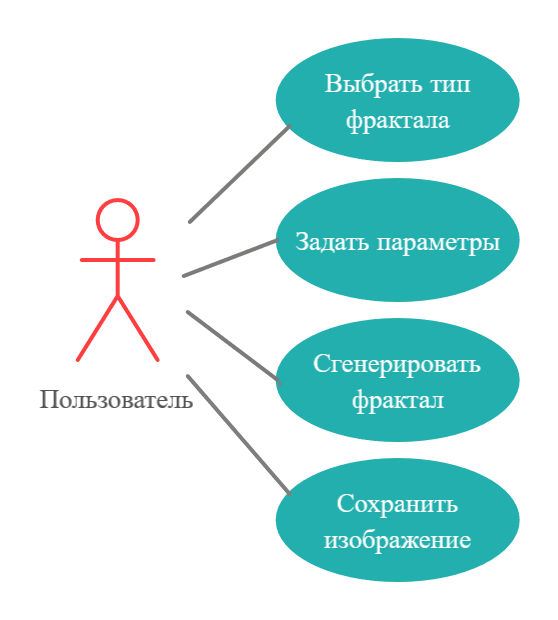


Рисунок 3 – Диаграмма вариантов использования

Название программы должно отражать суть ее работы, а также быть уникальным. Был разработан дизайн логотипа программы, а также ее иконка (рисунок 4 и 5 соответственно).



Рисунок 4 – Логотип программы Fractal Builder 2020



Рисунок 5 – Иконка программы Fractal Builder 2020

Пользовательский интерфейс должен быть интуитивно понятен, не перегружен функционалом и простым в использовании. Была создана главная форма программы, представляющая собой окно взаимодействия пользователя с программными компонентами. Эта форма представлена на рисунке 6.

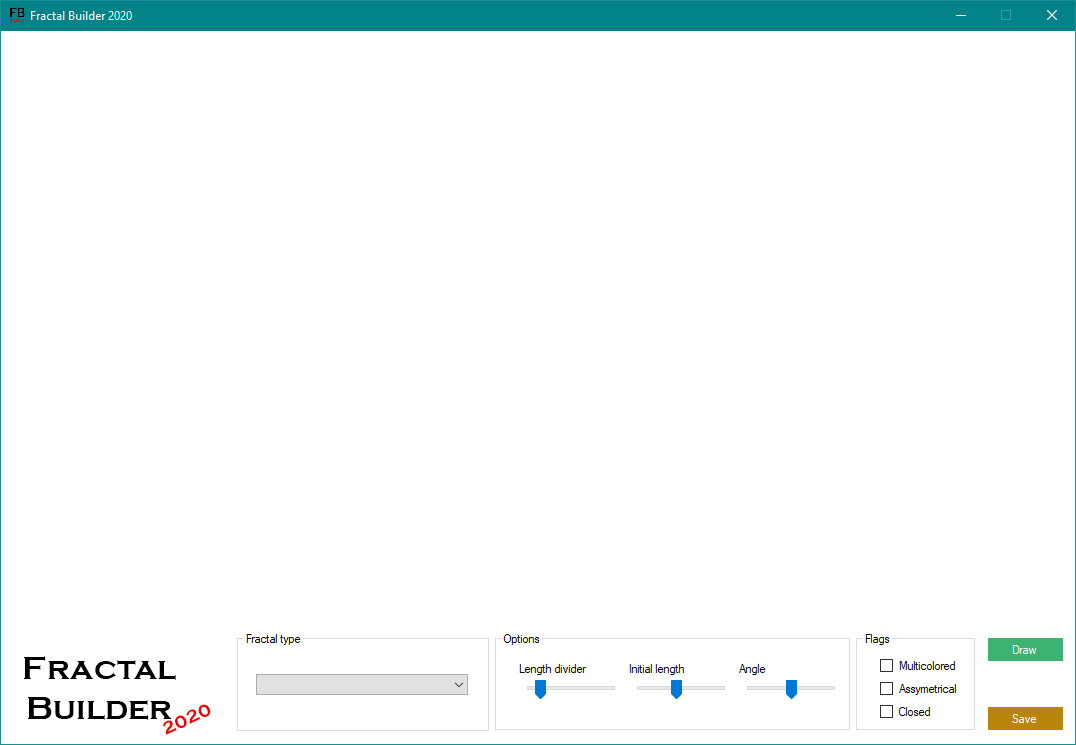


Рисунок 6 – Главная форма программы Fractal Builder

# 3 РАЗРАБОТКА

Для разработки был выбран язык программирования C# и «.NET Framework». Был создан частичный класс «MainForm». Для удобства понимания и написания кода, был создан отдельный файл «FractalsFunctions.cs», куда была вынесена реализация работы алгоритмов для построения фракталов. Код обработки событий элементов форм при пользовательском взаимодействии с программой, прописан в файле «MainForm.cs».

Пользователь перед построением фрактала выбирает его тип в выпадающем списке (рисунок 7), далее срабатывает функция обработки события на смену значения выбранного элемента «ComboBox», в связи с чем происходит изменение элементов формы, отвечающих за параметры, например, минимальные и максимальные значения ползунка (элемента «TrackBar») меняются в зависимости от типа выбранного элемента в «ComboBox». Функция обработки события на смену элемента в форме имеет название «ComboBoxSelectedIndexChanged».

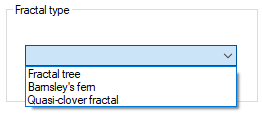


Рисунок 7 – Выпадающий список «ComboBox»

После выбора типа фрактала в выпадающем списке, пользователь задает параметры для генерируемого изображения. В зависимости от типа выбранного фрактала, некоторые элементы параметров могут быть недоступны, то есть иметь статус «disabled». Например, фрактал Квазиклевера не может быть асимметричным, поэтому пользователю не дана возможность его таким сделать (рисунок 8).

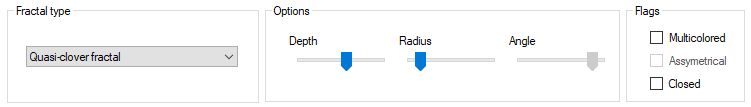


Рисунок 8 – Отключение элемента «CheckBox»

После указания значения параметров пользователь может нажать на кнопку «Draw» и сгенерировать фрактал. При нажатии на кнопку срабатывает функция «DrawButtonClick», которая в зависимости от типа выбранного элемента в «ComboBox» вызывает функцию фонового заполнения элемента «PictureBox», на котором выполняется графическое построение фрактала, а затем соответствующую функцию построения фрактала.

Функция «FillPictureBox», которая вызывается при нажатии на кнопку «Draw» перед функцией отрисовки фрактала, нужна для возможности сохранения получившегося изображения. Она создает элемент типа «Bitmap», присваивает его параметру «pictureBox.Image», а затем заполняет фон белым цветом.

Функции отрисовки фракталов являются рекурсивными и похожи друг на друга. Функции используют значения элементов формы, отвечающих за параметры. Внутри функции создается объект класса «Pen», которым выполняется рисование линий. Всего добавлено 3 типа фракталов: фрактальное дерево, папоротник Барнсли и фрактал Квазиклевера. Добавление новых типов в дальнейшем не будет вызывать сложности у разработчика.

После генерации фрактала пользователь может сгенерировать новый или сохранить получившийся. Сохранение происходит после нажатия на кнопку «Save». Кнопка в свою очередь вызывает функцию «SaveButtonClick», которая открывает пользователю окно сохранения, с возможностью выбрать местоположение файла, его тип и название (рисунок 9).

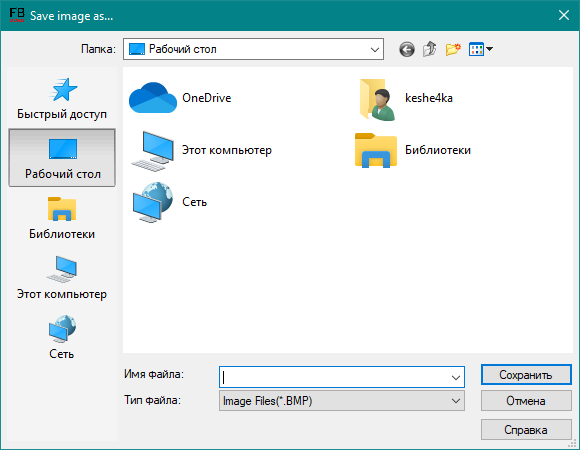


Рисунок 9 – Окно сохранения фрактала

# 4 ТЕСТИРОВАНИЕ И ОТЛАДКА

Целью тестирования является выявление, локализация и исправление ошибок. Во время написания программы, после написания каждой новой функции, улучшения или исправления, проводилось тестирование «белого ящика».

При системном тестировании ошибок выявлено не было.

После завершения написания всех частей программы было проведено нагрузочное тестирование, которое определило границы параметров, которые мог задать пользователь. Например, ползунок «Initial length» у фрактала «Fractal tree» задает начальную длину первого ствола. При слишком больших значениях всей картины не было видно, а время выполнения задачи могло составлять минуту. Тестирование позволило подобрать оптимальные границы параметров.

Было проведено тестирование производительности. При стрессовом тестировании было выявлено, что при многочисленных быстрых нажатиях на кнопку «Draw», программа создавала после каждого нажатия новый фрактал, из-за чего время ответа программы было большим. Для устранения была исправлена возможность так часто нажимать кнопку «Draw», повторное нажатие регистрировалось только через секунду (рисунок 10).

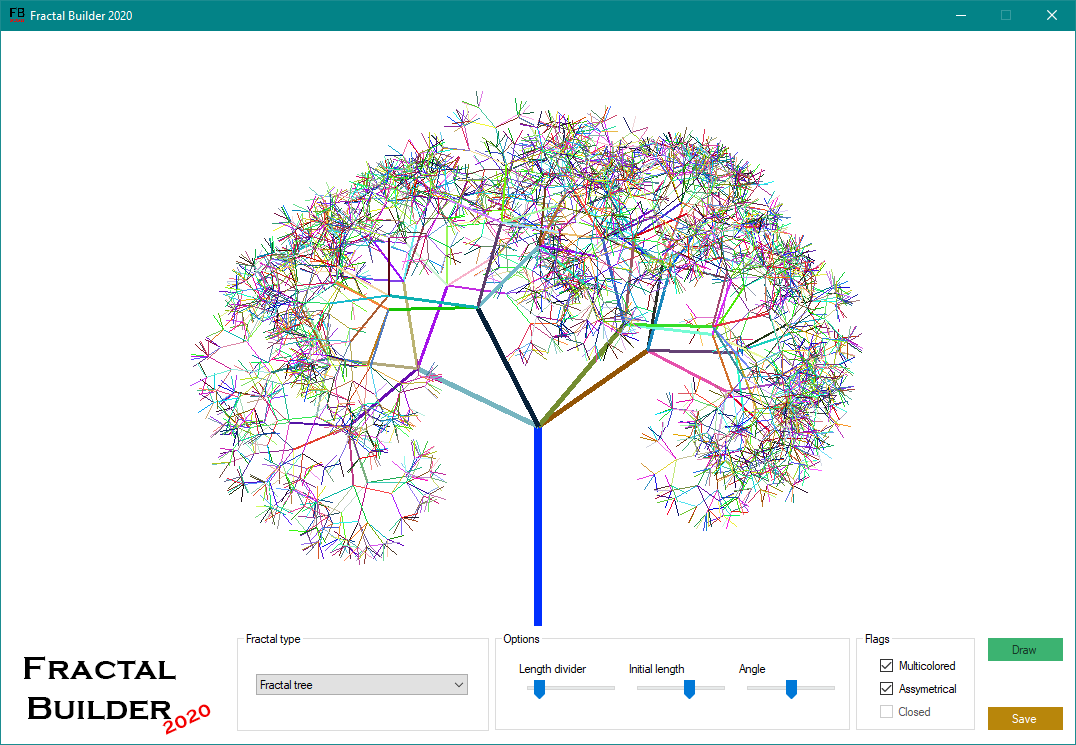


Рисунок 10 – Неактивная кнопка «Draw»

При поиске «User errors» была найдена возможность сохранить пустое изображение. Она возникала тогда, когда пользователь нажимал кнопку «Draw», не выбирая перед этим тип фрактала в развертывающемся списке. Ошибка была связана с заполнением «pictureBox.Image» белым фоном, из-за чего при проверке «pictureBox» на «null», после которого вызывалось сохранение при «false», система давала сохранять пустое изображение. Решением было убрать заполнение «pictureBox.Image» белым фоном в условие, при котором выполнялась проверка на наличие выбора в разворачивающемся списке типа фракталов.

Во время написания программы также использовалась пошаговая отладка (рисунок 11), которая была нужна для проверки правильности работы функции создания фрактала.

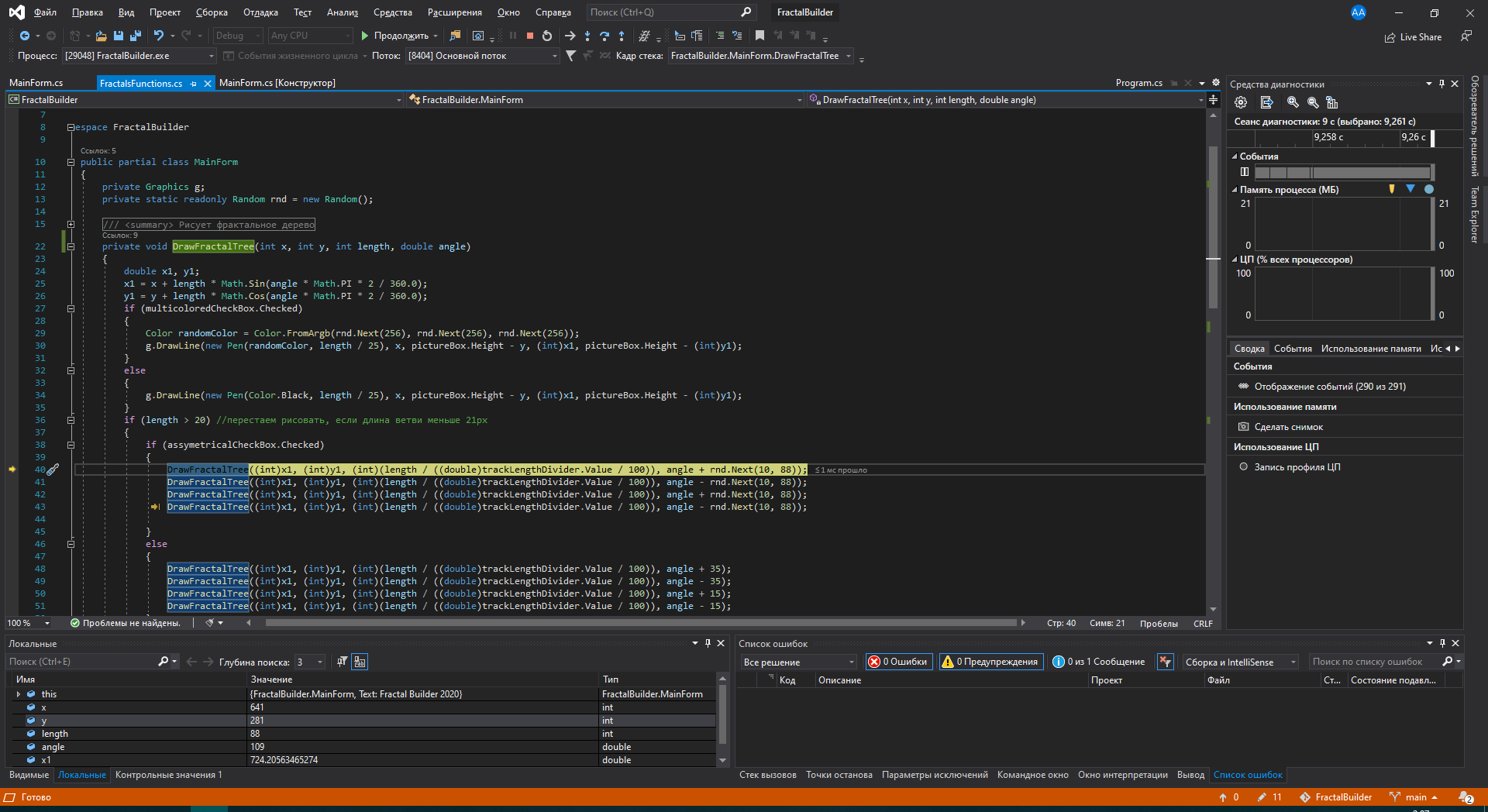


Рисунок 11 – Выполнение пошаговой отладки при проверке функции «DrawFractalTree»

# 5 ДОКУМЕНТИРОВАНИЕ

5.1 Руководство пользователя.

После запуска программы выберите тип создаваемого фрактала в области «Fractal type», для этого нажмите левой кнопкой мыши на разворачивающийся список, а затем левой кнопкой мыши нажмите по нужной строчке (рисунок 12).

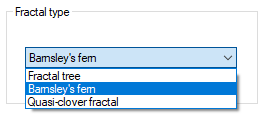


Рисунок 12 – Выбор типа фрактала

После выбор типа фрактала, вы можете задать для него параметры. Для ползунков в области «Options», путем наведения на них курсора, нажатием левой кнопки мыши и перетаскиванием в нужную сторону, вы можете регулировать различные параметры. В области «Flags» вам представлены флаговые кнопки с некоторыми доступными параметрами, которые вы можете включить путем нажатия левой кнопки мыши по квадратику флаговой кнопки.

Список параметров для «Fractal tree» представлен на рисунке 13. В области «Options» находятся ползунки, где «Length divider» - делитель длины ствола каждой новой ветви, «Initial length» - начальная длина ствола, «Angle» - угол наклона фрактала. В области «Flags» вы можете сделать фрактал разноцветным (флаговая кнопка «Multicolored») или ассиметричным (флаговая кнопка «Assymetrical»).

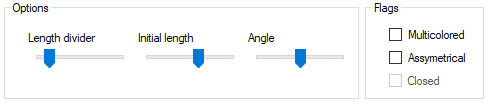


Рисунок 13 – Параметры для фрактала «Fractal tree»

Список параметров для «Barnsley’s fern» представлен на рисунке 14. В области «Options» находятся ползунки, где «Length divider» - делитель длины ствола каждой новой ветви, «Initial length» - начальная длина ствола, «Angle» - угол наклона ветвей папоротника к стволу. В области «Flags» вы можете сделать фрактал разноцветным (флаговая кнопка «Multicolored»).

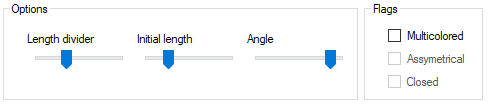


Рисунок 14 – Параметры для фрактала «Barnsley’s fern»

Список параметров для «Quasi-clover fractal» представлен на рисунке 15. В области «Options» находятся ползунки, где «Depth» - глубина фрактала, «Radius» - начальный радиус круга. В области «Flags» вы можете сделать фрактал разноцветным (флаговая кнопка «Multicolored») или «закрытым» (флаговая кнопка «Closed»).

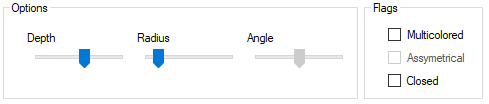


Рисунок 15 – Параметры для фрактала «Quasi-clover fractal»

После указания параметров вы можете нарисовать фрактал путем клика по кнопке «Draw» левой кнопкой мыши. Вы увидите созданный фрактал (рисунок 16).

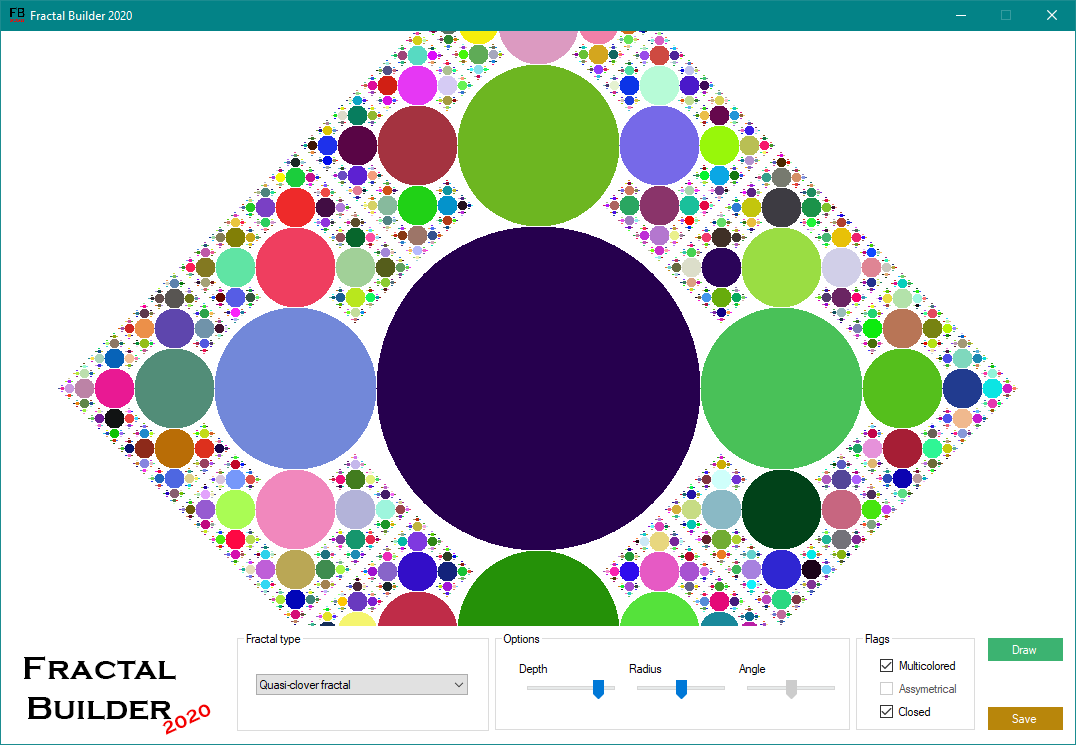


Рисунок 16 – Созданный фрактал

Для сохранения созданного фрактала нажмите на кнопку «Save». Вы увидите окно сохранения (рисунок 17). Для сохранения изображения, выберите место расположения сохраняемого изображения, его тип (по умолчанию .bmp) и укажите название файла.

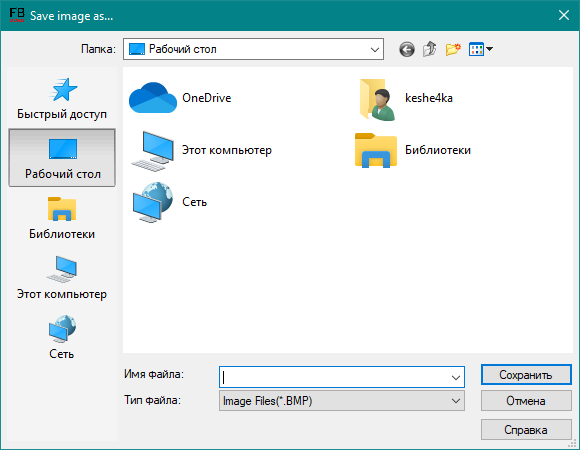


Рисунок 17 – Окно сохранения фрактала

Для выхода из программы нажмите на кнопку крестика в правом верхнем углу программы.

5.1 Документирование кода

Код программы является само документируемым. Для функций создания фракталов написана XML документация, для некоторых функций других функций или частей программы написаны комментарии, например для функции таймера (рисунок 18).

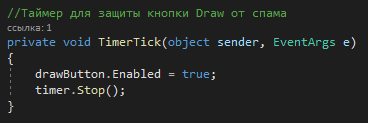


Рисунок 18 – Пример комментария в коде программы

## ЗАКЛЮЧЕНИЕ

В конце работы была получена готовая версия программы, пригодная к эксплуатации пользователем. Программа позволяет строить фракталы с задаваемыми параметрами, позволяющими получать различные друг от друга изображения. В то же время интерфейс программы получился интуитивно понятным, не загруженным, но в то же время функциональным. ПО получилось сопровождаемым, код программы приведен к «code style», что облегчит его понимание другими разработчиками.

## ПРИЛОЖЕНИЕ А

(обязательное)

Файл «MainForm.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;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FractalBuilder

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

/// <summary>

/// Заполняет pictureBox белым

/// </summary>

private void FillPictureBox()

{

var bitmap = new Bitmap(pictureBox.Width, pictureBox.Height);

pictureBox.Image = bitmap;

g = Graphics.FromImage(bitmap);

g.FillRectangle(Brushes.White, 0, 0, pictureBox.Width, pictureBox.Height);

}

private void DrawButtonClick(object sender, EventArgs e)

{

drawButton.Enabled = false;

timer.Start();

if (comboBox.SelectedItem == comboBox.Items[0])

{

FillPictureBox();

DrawFractalTree(pictureBox.Width / 2, 0, trackLength.Value, trackAngle.Value);

}

else if (comboBox.SelectedItem == comboBox.Items[1])

{

FillPictureBox();

DrawBarnsleysFern(pictureBox.Width / 2, pictureBox.Height, trackLength.Value, Math.PI / 2);

}

else if (comboBox.SelectedItem == comboBox.Items[2])

{

FillPictureBox();

int dir = 1;

if (closedCheckBox.Checked) { dir = 6; } //при dir == 6, фрактал получается "закрытого" типа

DrawQuasiCloverFractal(pictureBox.Width / 2, 3 \* pictureBox.Height / 5, trackLength.Value, dir, trackLengthDivider.Value);

}

}

private void ComboBoxSelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox.SelectedItem == comboBox.Items[0])

{

trackLength.Minimum = 40;

trackLength.Maximum = 300;

trackLength.Value = 200;

labelInitialLength.Text = "Initial length";

trackLengthDivider.Minimum = 140;

trackLengthDivider.Maximum = 240;

trackLengthDivider.Value = 150;

labelLengthDivider.Text = "Length divider";

trackAngle.Minimum = -75;

trackAngle.Maximum = 75;

trackAngle.Value = 0;

closedCheckBox.Enabled = false;

assymetricalCheckBox.Enabled = true;

trackAngle.Enabled = true;

}

else if (comboBox.SelectedItem == comboBox.Items[1])

{

trackLength.Minimum = 40;

trackLength.Maximum = 300;

trackLength.Value = 100;

labelInitialLength.Text = "Initial length";

trackLengthDivider.Minimum = 50;

trackLengthDivider.Maximum = 200;

trackLengthDivider.Value = 100;

labelLengthDivider.Text = "Length divider";

trackAngle.Minimum = 4;

trackAngle.Maximum = 15;

trackAngle.Value = 14;

assymetricalCheckBox.Enabled = false;

closedCheckBox.Enabled = false;

trackAngle.Enabled = true;

}

else if (comboBox.SelectedItem == comboBox.Items[2])

{

trackLength.Minimum = 20;

trackLength.Maximum = 300;

trackLength.Value = 50;

labelInitialLength.Text = "Radius";

trackLengthDivider.Minimum = 1;

trackLengthDivider.Maximum = 8;

trackLengthDivider.Value = 5;

labelLengthDivider.Text = "Depth";

assymetricalCheckBox.Enabled = false;

closedCheckBox.Enabled = true;

trackAngle.Enabled = false;

}

}

private void SaveButtonClick(object sender, EventArgs e)

{

if (pictureBox.Image != null)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Title = "Save image as...";

sfd.OverwritePrompt = true;

sfd.CheckPathExists = true;

sfd.Filter = "Image Files(\*.BMP)|\*.BMP|Image Files(\*.JPG)|\*.JPG|Image Files(\*.GIF)|\*.GIF|Image Files(\*.PNG)|\*.PNG|All files (\*.\*)|\*.\*";

sfd.ShowHelp = true;

if (sfd.ShowDialog() == DialogResult.OK)

{

try

{

pictureBox.Image.Save(sfd.FileName, System.Drawing.Imaging.ImageFormat.Jpeg);

}

catch

{

MessageBox.Show("Unable to save image", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

//Таймер для защиты кнопки Draw от спама

private void TimerTick(object sender, EventArgs e)

{

drawButton.Enabled = true;

timer.Stop();

}

}

}

## ПРИЛОЖЕНИЕ Б

(обязательное)

Файл «FractalsFunctions.cs»

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FractalBuilder

{

public partial class MainForm

{

private Graphics g;

private static readonly Random rnd = new Random();

/// <summary>

/// Рисует фрактальное дерево

/// </summary>

/// <param name="x">координата x</param>

/// <param name="y">координата y</param>

/// <param name="length">длина ветви</param>

/// <param name="angle">угол поворота дерева</param>

private void DrawFractalTree(int x, int y, int length, double angle)

{

double x1, y1;

x1 = x + length \* Math.Sin(angle \* Math.PI \* 2 / 360.0);

y1 = y + length \* Math.Cos(angle \* Math.PI \* 2 / 360.0);

if (multicoloredCheckBox.Checked)

{

Color randomColor = Color.FromArgb(rnd.Next(256), rnd.Next(256), rnd.Next(256));

g.DrawLine(new Pen(randomColor, length / 25), x, pictureBox.Height - y, (int)x1, pictureBox.Height - (int)y1);

}

else

{

g.DrawLine(new Pen(Color.Black, length / 25), x, pictureBox.Height - y, (int)x1, pictureBox.Height - (int)y1);

}

if (length > 20) //перестаем рисовать, если длина ветви меньше 21px

{

if (assymetricalCheckBox.Checked)

{

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle + rnd.Next(10, 88));

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle - rnd.Next(10, 88));

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle + rnd.Next(10, 88));

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle - rnd.Next(10, 88));

}

else

{

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle + 35);

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle - 35);

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle + 15);

DrawFractalTree((int)x1, (int)y1, (int)(length / ((double)trackLengthDivider.Value / 100)), angle - 15);

}

}

}

/// <summary>

/// Рисует папоротник Барнсли

/// </summary>

/// <param name="x">координата x</param>

/// <param name="y">координата у</param>

/// <param name="length">длина ветви</param>

/// <param name="angle">угол наклона</param>

private void DrawBarnsleysFern(int x, int y, double length, double angle)

{

if (multicoloredCheckBox.Checked)

{

Color randomColor = Color.FromArgb(rnd.Next(256), rnd.Next(256), rnd.Next(256));

g.DrawLine(new Pen(randomColor, (float)length / 25), x, y, (int)Math.Round(x + length \* Math.Cos(angle)), (int)Math.Round(y - length \* Math.Sin(angle)));

}

else

{

g.DrawLine(new Pen(Color.Black, (float)length / 25), x, y, (int)Math.Round(x + length \* Math.Cos(angle)), (int)Math.Round(y - length \* Math.Sin(angle)));

}

if (length > (double)trackLengthDivider.Value / 100)

{

x = (int)Math.Round(x + length \* Math.Cos(angle));

y = (int)Math.Round(y - length \* Math.Sin(angle));

DrawBarnsleysFern(x, y, length \* 0.4, angle - trackAngle.Value \* Math.PI / 30);

DrawBarnsleysFern(x, y, length \* 0.4, angle + trackAngle.Value \* Math.PI / 30);

DrawBarnsleysFern(x, y, length \* 0.7, angle + Math.PI / 30);

}

}

/// <summary>

/// Рисует фрактал Квазиклевера

/// </summary>

/// <param name="x0">x координата центра круга</param>

/// <param name="y0">y координата центра круга</param>

/// <param name="r">радиус круга</param>

/// <param name="dir">индекс, определяющий положение родителя</param>

/// <param name="iteration">количество итераций</param>

private void DrawQuasiCloverFractal(int x0, int y0, int r, int dir, int iteration)

{

if (multicoloredCheckBox.Checked)

{

Color randomColor = Color.FromArgb(rnd.Next(256), rnd.Next(256), rnd.Next(256));

g.FillEllipse(new SolidBrush(randomColor), x0 - r, y0 - r, 2 \* r, 2 \* r);

}

else

{

g.FillEllipse(new SolidBrush(Color.Black), x0 - r, y0 - r, 2 \* r, 2 \* r);

}

if (iteration > 0)

{

int[] x = new int[4];

int[] y = new int[4];

int d = 3 \* r / 2;

x[0] = x0 - d;

y[0] = y0;

x[1] = x0;

y[1] = y0 - d;

x[2] = x0 + d;

y[2] = y0;

x[3] = x0;

y[3] = y0 + d;

for (int i = 0; i < 4; i++)

{

if (i - dir == 2 || i - dir == -2)

continue;

DrawQuasiCloverFractal(x[i], y[i], r / 2, i, iteration - 1);

}

}

}

}

}