МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

Институт информационных технологий

Кафедра информационных технологий и экономической информатики

ОТЧЕТ

по лабораторной работе №2

Авторы отчета С.М. Панов ПрИ-201

подпись инициалы, фамилия группа

А.Д. Казбеков ПрИ-201

подпись инициалы, фамилия группа

В.И. Кочетков ПрИ-201

подпись инициалы, фамилия группа

Отчет защищен \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

дата оценка

Челябинск 2024 г.

**Цель работы:** понять и реализовать рекурсивные алгоритмы.

**Задание.**

I. **Отрисовка фрактала.** Реализовать программу с графическим интерфейсом отрисовки фракталов. Для отрисовки фракталов использовать рекурсивный алгоритм.

II. **Реализовать алгоритм "Ханойские башни".** Реализовать программу с графическим интерфейсом решения головоломки «Ханойская башня».

Лабораторная работа была выполнена с использованием 6-ти классов: Абстрактный класс “Algorithms” с виртуальным методом “Draw”, который будет переопределяться в каждом из классов для отрисовки фракталов и ханойских башен. Класс “Julia” – это фрактал Жюлиа, в котором переопределен метод “Draw” под нужды отрисовки этого фрактала, класс “PythagorasTree” – это фрактал “пифагорово дерево”, класс “SnowflakeCurve” – это фрактал “снежинка Коха”. Класс “FractalManager” используется для реализации логики и удобства отрисовки каждого фрактала в xaml. Заключительный класс “Tower” описывает логику для реализации решения ханойских башен.

Стек технологий:

1. C#
2. Wpf (Windows Presentation Foundation)

**Задание 1**

Была реализована логика отрисовки фракталов таких как: “Julia”, “PythagorasTree” и “SnowflakeCurve”.

1. Julia’s Fractal (фрактал Жюлиа). Множеством Жюлиа полинома f(z) = z2 + c, соответственно называется такое подмножество множества комплексных чисел, для каждой точки которого, поведение функции под действием итераций является хаотичным, т.е. небольшие изменения в начальных условиях в некоторой небольшой окрестности начальной точки, значительно влияют на траекторию. В нашем случае будет 57 итераций, получится, что чем дальше точки из множества комплексных чисел, тем темнее они будут отрисованы. (см. рис. 1.1)

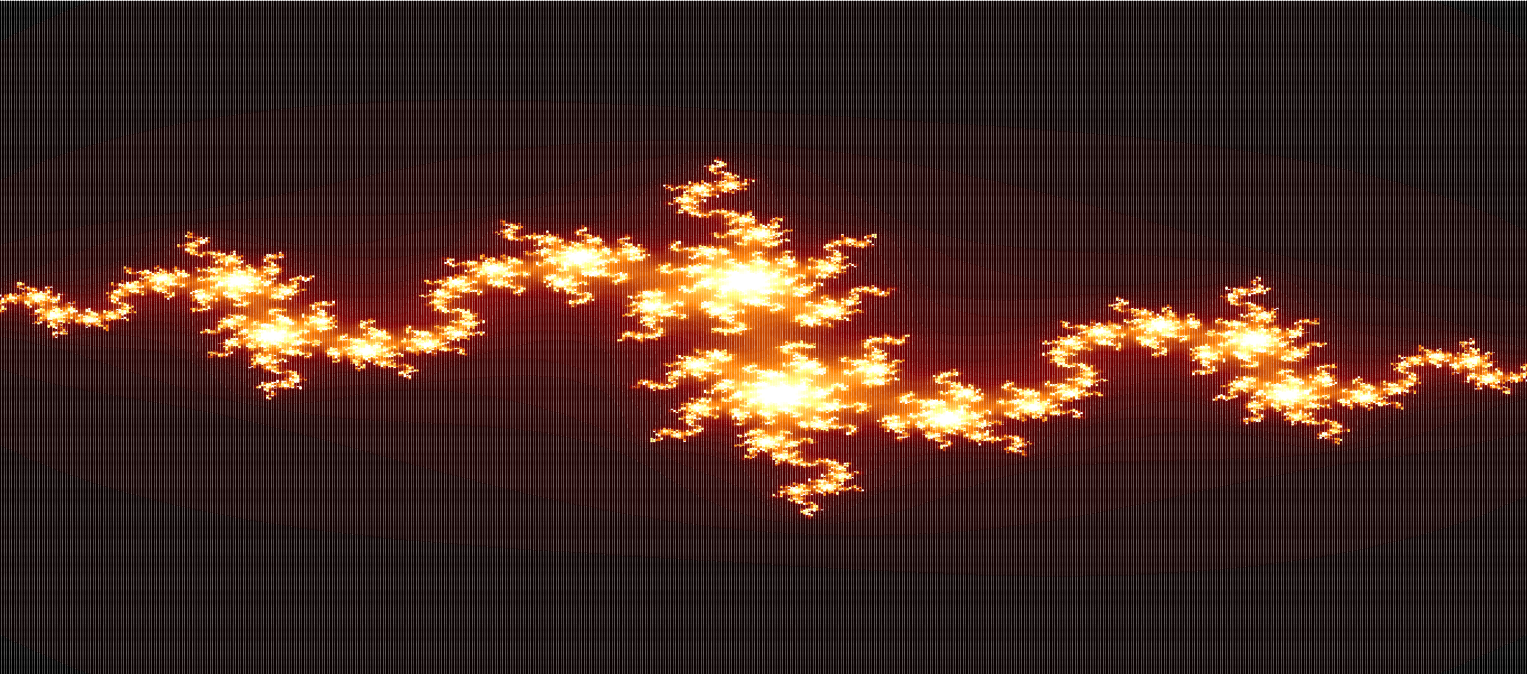


Рис. 1.1 Фрактал Жюлиа при 57 итерациях

Далее рассмотрим фрактал уже при 120 итерациях. (см. рис. 1.2)

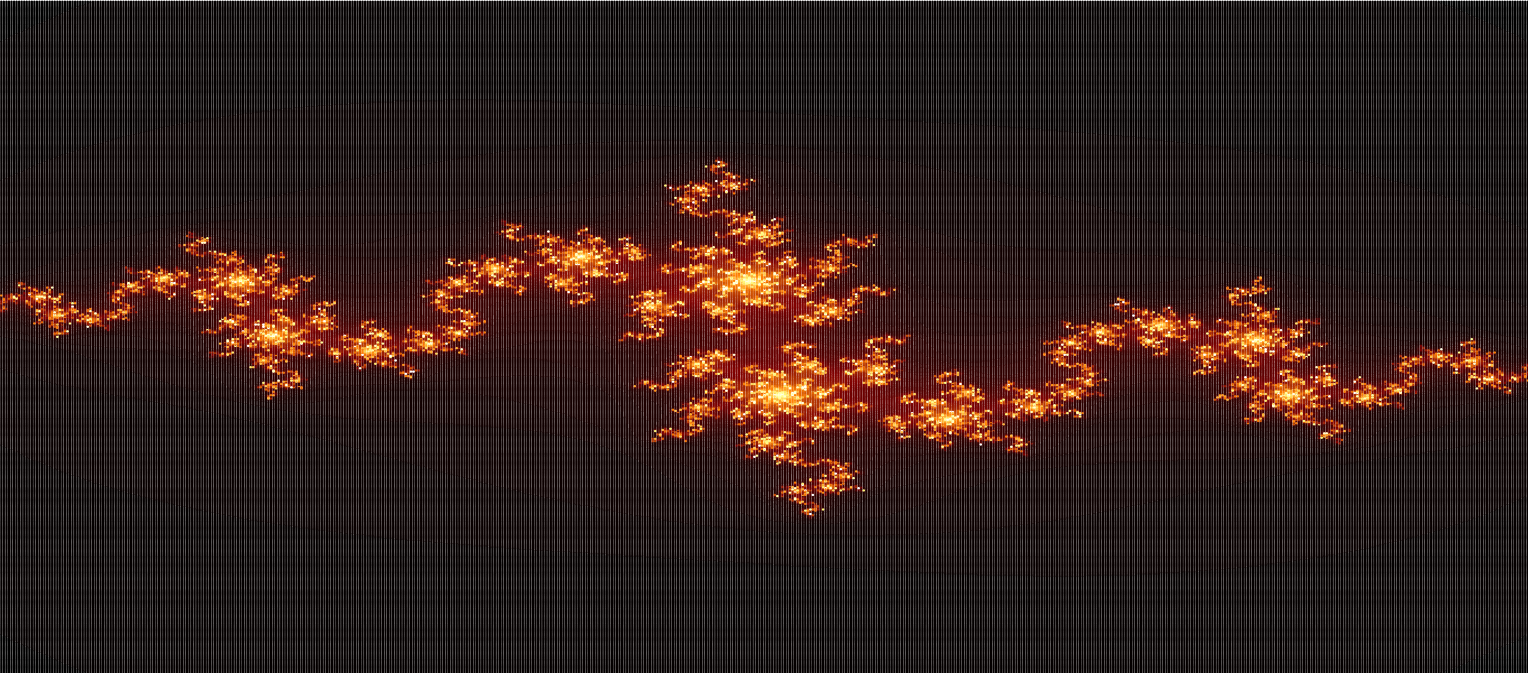


Рис. 1.2 Фрактал Жюлиа при 120 итерациях

Следующий пример, когда итераций уже 321. Видно, что точки все дальше выходят из заданного комплексного множества от чего становятся темнее. (см. рис. 1.3)

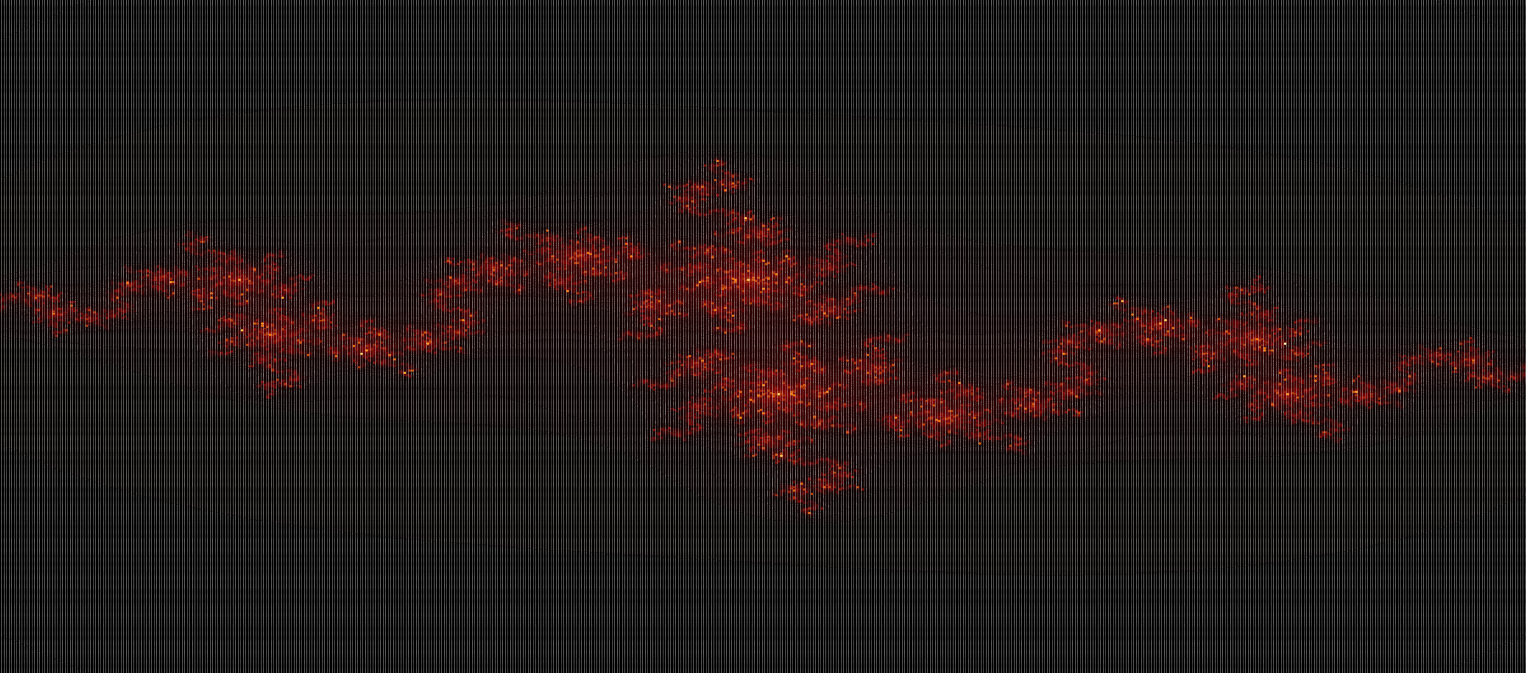


Рис. 1.3 Фрактал Жюлиа при 321 итерации

С помощью метода, представленного на картинке ниже (см. рис. 1.1.1), мы задаем комплексное множество.

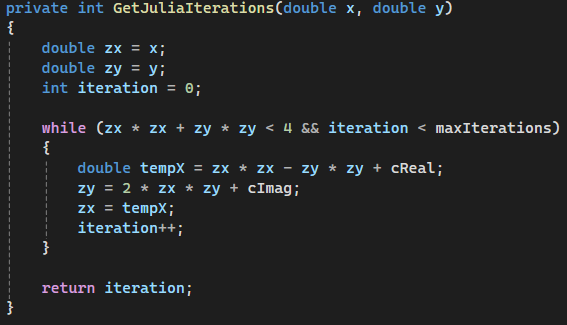


Рис 1.1.1 Метод для задания комплексного множества

Метод для отрисовки фрактала Жюлиа (см. рис. 1.1.2) на основе кол-ва итераций, полученных из метода выше (см. рис. 1.1.1).

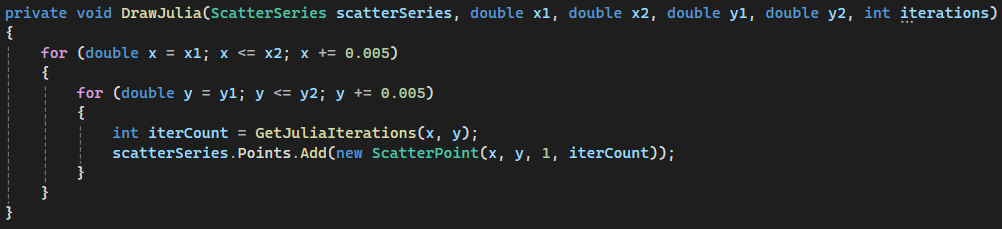


Рис. 1.1.2 Метод для рисования фрактала

1. PythagorasTree (дерево Пифагора). Называется так потому, что каждая тройка попарно соприкасающихся квадратов ограничивает прямоугольный треугольник и получается картинка, которой часто иллюстрируют теорему Пифагора, «пифагоровы штаны во все стороны равны». В случае с нашей реализацией получится дерево - которое будет “обдуваемое ветром”, т.е. будут прямые линии, а не квадраты. На рисунке представлен фрактал с 5 итерациями. (см. рис. 2.1)

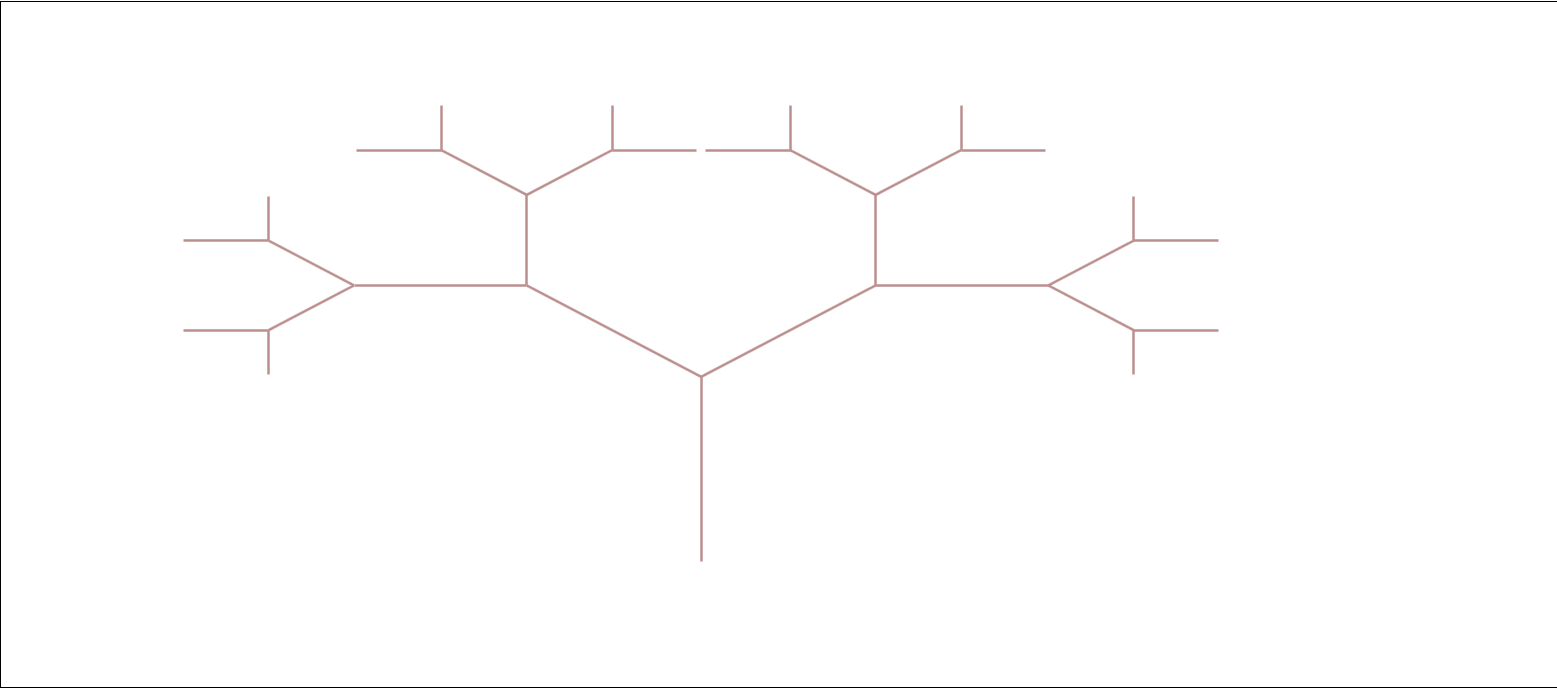


Рис. 2.1 Дерево Пифагора с 5 итерациями

Далее показан тот же фрактал, но уже с 13 итерациями. (см. рис. 2.2)

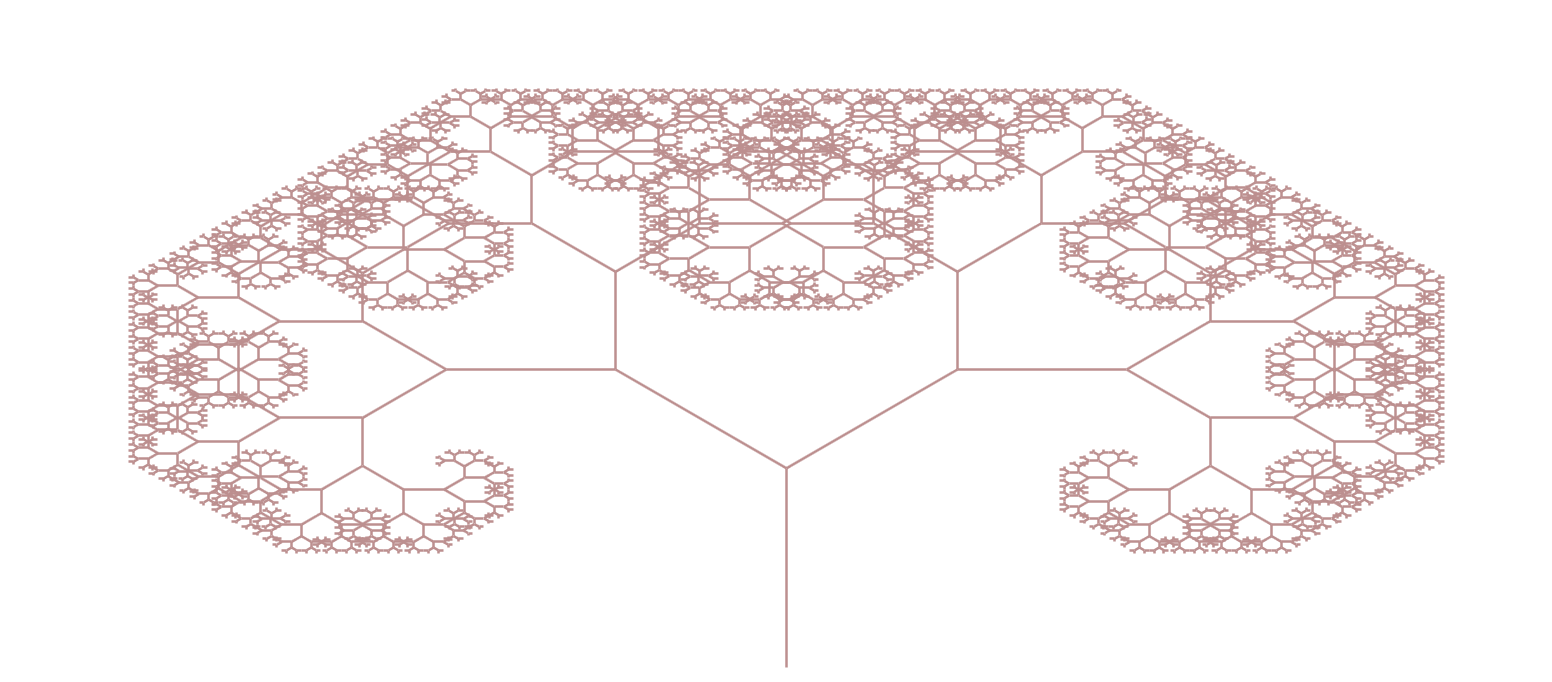


Рис. 2.2 Дерево Пифагора с 13 итерациями.

Ниже представлен код для построения дерева Пифагора (см. рис. 2.1.1)

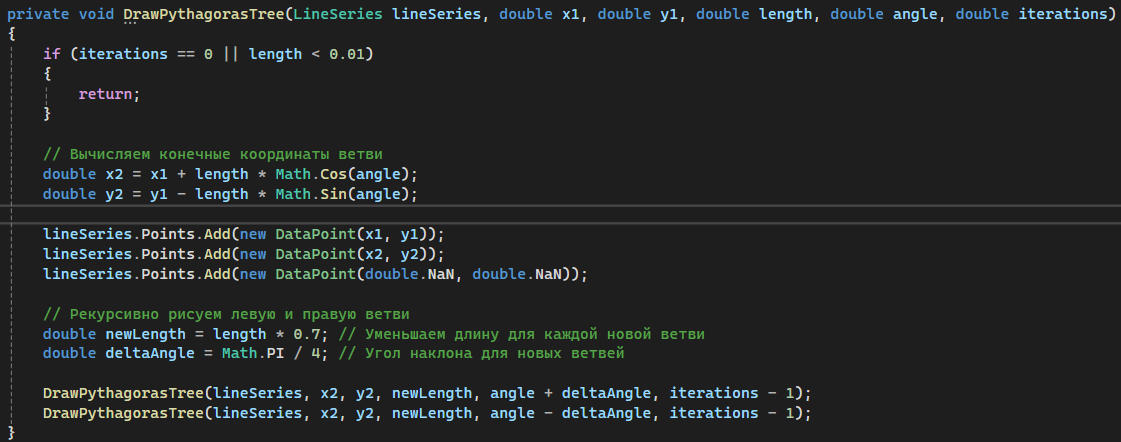


Рис. 2.1.1 Код для отрисовки дерева Пифагора

1. SnowflakeCurve (Снежинка Коха).  Три копии кривой Коха, построенные на сторонах правильного треугольника, образуют замкнутую кривую бесконечной длины, называемую снежинкой Коха. Кривая Коха, это само подобный отрезок. На рисунке представлена Снежинка Коха для 7 итераций. (см. рис. 3.1)

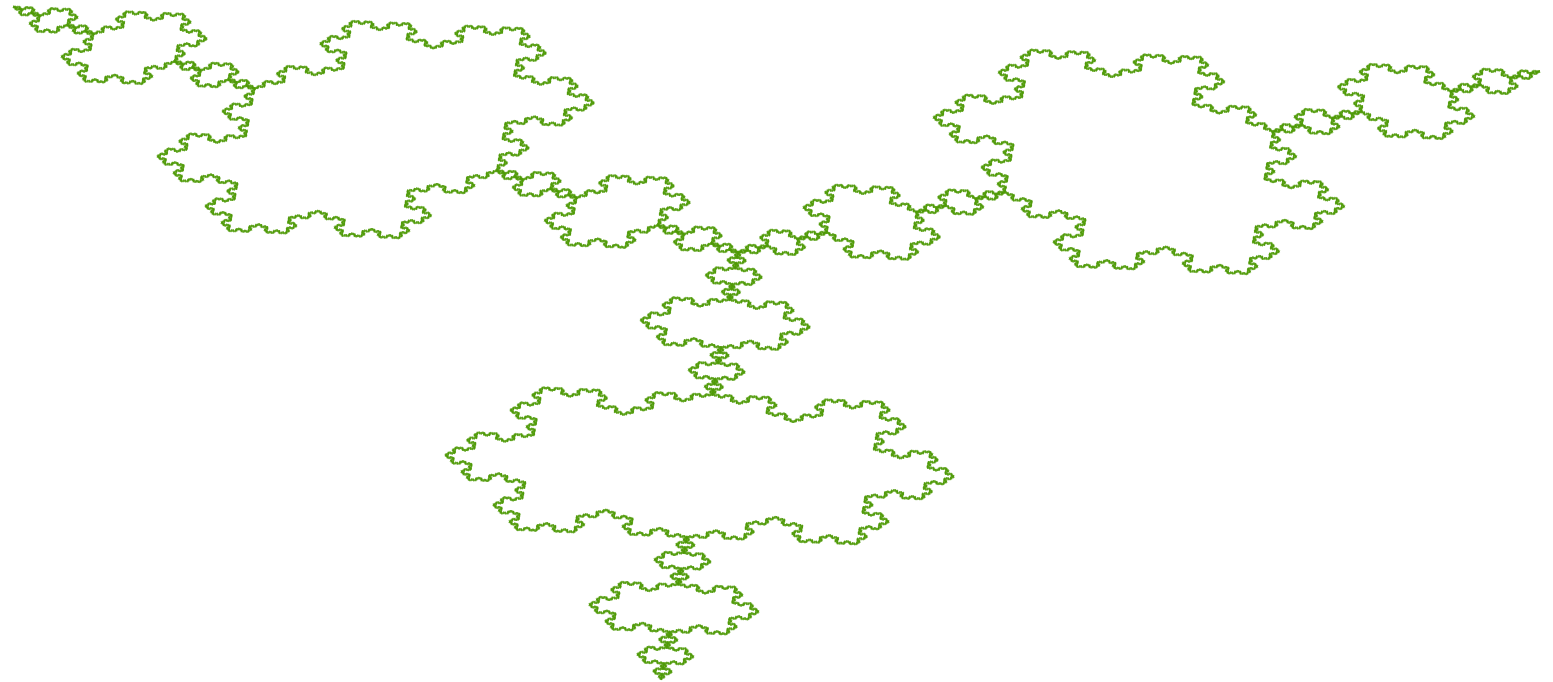


Рис. 3.1 Снежинка Коха для 7 итераций

Ниже представлен код для отрисовки снежинки Коха. (см. рис. 3.1.1)

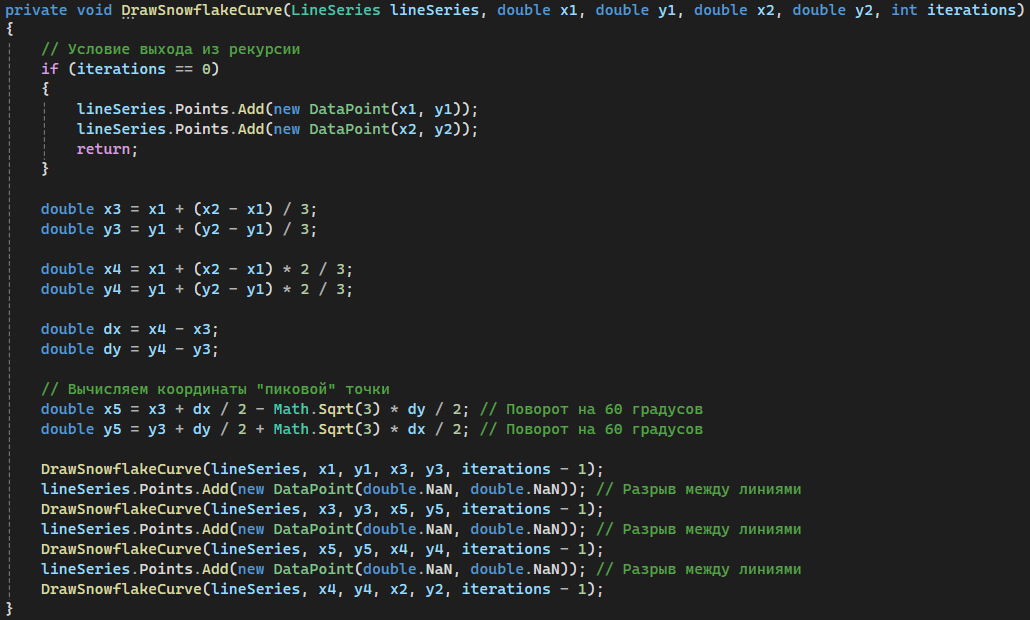


Рис. 3.1.1 Код для отрисовки снежинки Коха

**Задание 2**

Даны три стержня, на один из которых нанизаны n колец, причём кольца отличаются размером и лежат меньшее на большем. Задача состоит в том, чтобы перенести пирамиду из восьми колец за наименьшее число ходов на другой стержень. За один раз разрешается переносить только одно кольцо, причём нельзя класть большее кольцо на меньшее. (см. рис. 4.1)

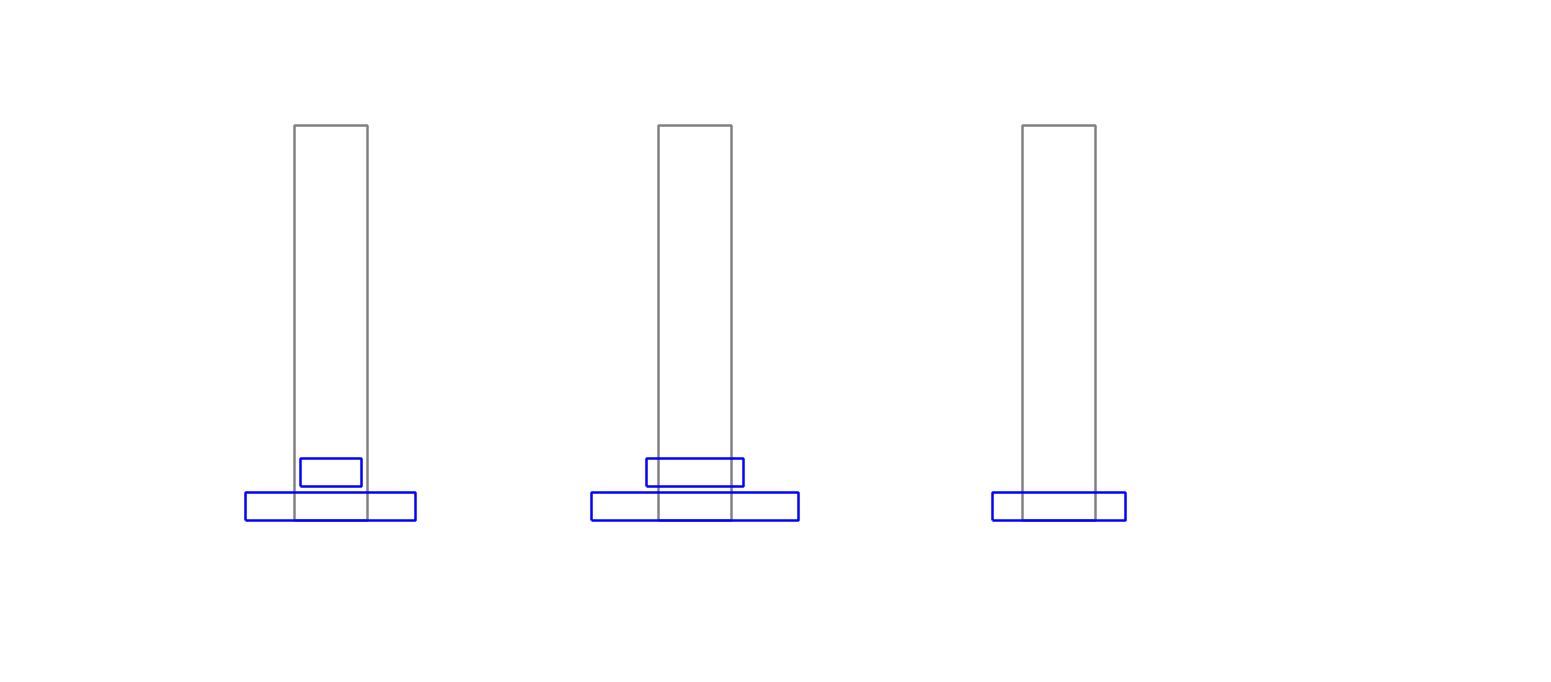


Рис. 4.1 Ханойские башни для 5-ти колец в процессе решения.

После выбора кол-ва колец, появляется уведомление (см. рис. 4.2), которое покажет сколько будет шагов для решения, которые высчитываются по формуле steps = 2n-1.

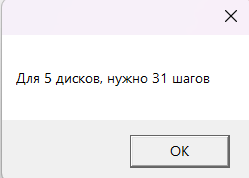


Рис. 4.2 Кол-во шагов для решения при n дисков

Реализация решения ханойских башен. Ниже представлен код метода рисования башен и дисков. (см. рис. 4.1.1).

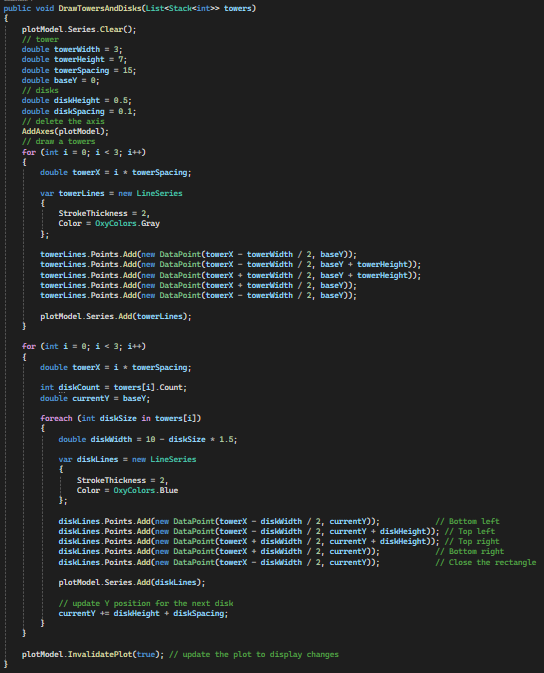


Рис. 4.1.1 Метод для рисования башен и дисков

Следующий метод позволяет перемещать диски между башнями. (см. рис. 4.1.2).

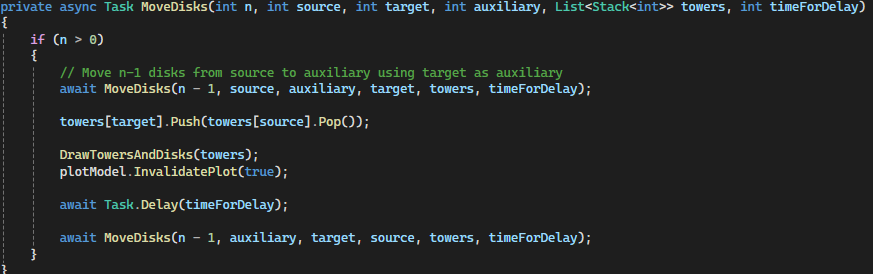


Рис. 4.1.2 Реализация метода перемещения дисков

Следующий метод, есть решение ханойских башен с использованием рекурсивного стека. (см. рис. 4.1.3).

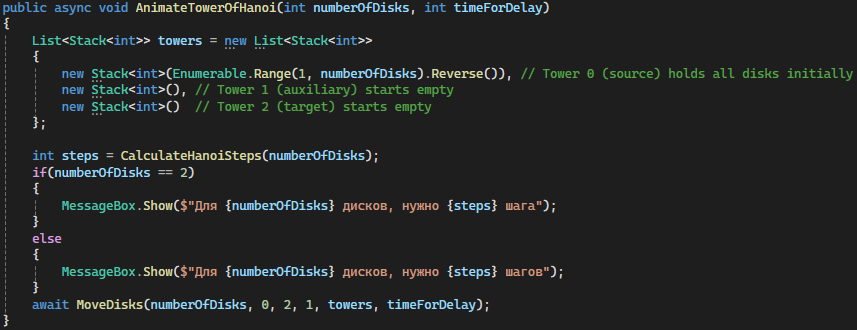


Рис. 4.1.3 Реализация решения ханойских башен через рекурсивный стек