МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ЧЕЛЯБИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт Информационных Технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

**№**

по курсу

Алгоритмы и анализ сложности

Группа: **ПрИ-202**

ФИО студента:

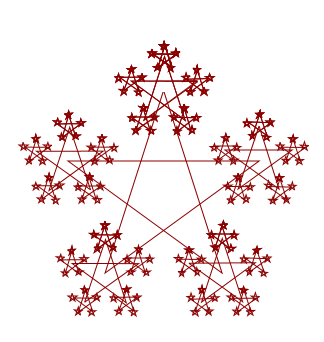
Соколов Данил Олегович

Отметки о ходе выполнения работы (дата, комментарии, балл):

Задание 1. Отрисовка фракталов

Древо Пифагора || Звезда

Звезда



Звезда – обычная пяти угольная звезда где на каждом конце рисуется ее более миниатюрная версия

const double scale = 0.35;

const int depth = 4;

double rotate = 0;

private void DrawStar(int level, double x, double y, double r)

{

const double offset = -Math.PI / 2;

const double angle = 4 \* Math.PI / 5;

Polyline star = new Polyline();

star.Stroke = Brushes.DarkRed;

image.Children.Add(star);

for (int i = 0; i <= 5; i++)

{

star.Points.Add(new Point((int)(x + r \* Math.Cos(offset + i \* angle)),

(int)(y + r \* Math.Sin(offset + i \* angle))));

star.RenderTransform = new RotateTransform(rotate, x, y);

if (level < depth)

{

rotate += 72;

DrawStar(level + 1, x + r \* Math.Cos(offset + i \* angle),

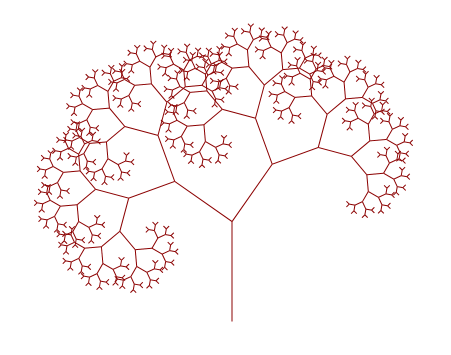
y + r \* Math.Sin(offset + i \* angle), r \* scale);

}

}

}

Древо Пифагора



Древо Пифагора – суть такова что каждая линия делиться на 2 других, причем каждая линия идет под определенным углом, в итоге получаем что то схожее на дерево

rivate static double \_angle1 = 55 \* (Math.PI / 180);

private static double Angle1

{

set => \_angle1 = value \* (Math.PI / 180);

}

private static double \_angle2 = 35 \* (Math.PI / 180);

private static double Angle2

{

set => \_angle2 = value \* (Math.PI / 180);

}

private static double \_ratio = 0.7;

private static int LengthRatio

{

set => \_ratio = value / 100d;

}

private double FirstIterationLineLength = 100;

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

{

image.Children.Clear();

double xmid = image.Width;

double ymid = image.Height;

Draw(xmid / 2, ymid / 2, 1);

}

public void Draw(double x, double y, uint level)

{

Line line = new Line

{

X1 = x,

X2 = x,

Y1 = y,

Y2 = y - FirstIterationLineLength,

Stroke = Brushes.DarkRed

};

image.Children.Add(line);

if (level < 12)

{

Draw(line, level + 1, Math.PI / 2);

}

}

private void Draw(Line previousLine, uint level, double prevAngle)

{

double leftAngle = prevAngle + \_angle1;

double rightAngle = prevAngle - \_angle2;

Line line1 = new Line

{

X1 = previousLine.X2,

X2 = previousLine.X2 + Math.Cos(leftAngle) \* FirstIterationLineLength \* Math.Pow(\_ratio, level - 1),

Y1 = previousLine.Y2,

Y2 = previousLine.Y2 - Math.Sin(leftAngle) \* FirstIterationLineLength \* Math.Pow(\_ratio, level - 1),

Stroke = Brushes.DarkRed

};

Line line2 = new Line

{

X1 = previousLine.X2,

X2 = previousLine.X2 + Math.Cos(rightAngle) \* FirstIterationLineLength \* Math.Pow(\_ratio, level - 1),

Y1 = previousLine.Y2,

Y2 = previousLine.Y2 - Math.Sin(rightAngle) \* FirstIterationLineLength \* Math.Pow(\_ratio, level - 1),

Stroke = Brushes.DarkRed

};

image.Children.Add(line1);

image.Children.Add(line2);

if (level >= 10)

return;

Draw(line1, level + 1, leftAngle);

Draw(line2, level + 1, rightAngle);

}

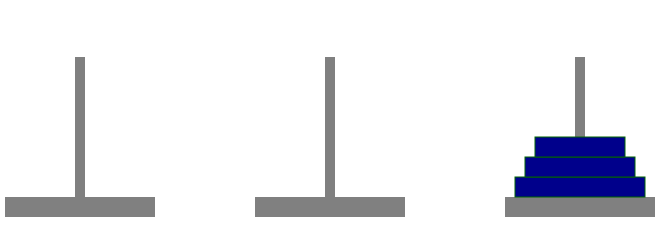
Задание 2. Реализовать алгоритм "Ханойские башни"

Ханойские башни , решенные с помощью рекурсии где внутри метода будет 3 шага:

1 шаг) Вызов рокировки башен для того чтоб понять с какой на какую башню стоит начать

2 шаг) Переброска с 1 башни к другой

3 шаг) Метод где меняем стартовую и вспомогательную или вспомогательную и конечную для переброски более крупного круга в конец конечной башни



Алгоритм

private double x = 0;

private double y = 0;

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

{

Town[] towns = new Town[3];

towns[0] = new Town() { x = 50 , num = 1};

towns[1] = new Town() { x = 300, num = 2 };

towns[2] = new Town() { x = 550, num = 3 };

for (int i = 3; i >= 1; i--)

{

towns[0].sizeDisks.Add(i);

}

image.Children.Clear();

x = image.Width;

y = image.Height;

for (int i = 0; i < towns.Length; i++)

{

DrawTown(towns[i]);

}

for (int i = 0; i < towns.Length; i++)

{

DrawDisks(towns[i]);

}

MoveDisks(towns, towns[0], towns[1], towns[2], 3);

DrawFinal(towns);

}

void MoveDisks(Town[] towns, Town start, Town temp, Town end, int disks)

{

if (disks > 1)

{

MoveDisks(towns,start, end, temp, disks - 1); // 1 шаг

}

end.sizeDisks.Add(start.sizeDisks[start.sizeDisks.Count - 1]); // 2 шаг

start.sizeDisks.RemoveAt(start.sizeDisks.Count - 1);

if (disks > 1) // 3 шаг

{

MoveDisks(towns, temp, start, end, disks - 1);

}

}

Вспомогательный класс Башни

using System;

using System.Collections.Generic;

using System.Text;

namespace HomeWork\_2

{

class Town

{

public List<int> sizeDisks = new List<int>();

public int x { get; set; }

public int lengthTown { get; } = 150;

public int CountDisk { get; set; } = 0;

public int num { get; set; }

}

}

Отрисовка

private void DrawFinal(Town[] towns)

{

List<Polygon> deleteDiskLisk = new List<Polygon>();

foreach (Polygon child in image.Children)

{

if (child.Name == "disk")

{

deleteDiskLisk.Add(child);

}

}

for (int i = 0; i < deleteDiskLisk.Count; i++)

{

image.Children.Remove(deleteDiskLisk[i]);

}

for (int i = 0; i < towns.Length; i++)

{

DrawDisks(towns[i]);

}

}

private void DrawTown(Town town)

{

Polygon basis = new Polygon();

basis.Points.Add(new Point(town.x, y - 25));

basis.Points.Add(new Point(town.x + town.lengthTown, y - 25));

basis.Points.Add(new Point(town.x + town.lengthTown, y - 45));

basis.Points.Add(new Point(town.x, y - 45));

basis.Visibility = Visibility.Visible;

basis.Fill = Brushes.Gray;

image.Children.Add(basis);

Polygon height = new Polygon();

height.Points.Add(new Point(town.x + 80, y - 45));

height.Points.Add(new Point(town.x + town.lengthTown -80, y - 45));

height.Points.Add(new Point(town.x + town.lengthTown -80, y - 185));

height.Points.Add(new Point(town.x + 80, y - 185));

height.Visibility = Visibility.Visible;

height.Fill = Brushes.Gray;

image.Children.Add(height);

}

private void DrawDisks(Town town)

{

for (int i = 1; i <= town.sizeDisks.Count; i++)

{

town.CountDisk++;

Polygon disks = new Polygon();

disks.Points.Add(new Point(town.x + 10 \* i, y - 25 - 20 \* town.CountDisk));

disks.Points.Add(new Point(town.x + town.lengthTown - 10 \* i, y - 25 - 20 \* town.CountDisk));

disks.Points.Add(new Point(town.x + town.lengthTown - 10 \* i, y - 45 - 20 \* town.CountDisk));

disks.Points.Add(new Point(town.x + 10 \* i, y - 45 - 20 \* town.CountDisk));

disks.Visibility = Visibility.Visible;

disks.Fill = Brushes.DarkBlue;

disks.Stroke = Brushes.DarkGreen;

disks.Name = "disk";

image.Children.Add(disks);

}

}