





//Задание:

//Для вариантов 1–6:

//Получить кривую дракона N-го порядка. Каждой кривой ставится в

//соответствие последовательность, состоящая из нулей и единиц, где единица

//соответствует повороту кривой налево, а ноль – повороту направо. Кривая

//дракона первого порядка имеет двоичную

//формулу 1. Для того, чтобы получить двоичную формулу кривой дракона

//каждого следующего порядка, следует приписать справа к формуле кривой

//предыдущего порядка единицу. Полученная последовательность даёт половину

//искомой формулы. Затем в последовательности цифр, предшествующих

//приписанной единице, следует заменить на ноль единицу, стоящую в её

//середине, после чего приписать полученную последовательность справа от уже

//построенной части формулы.

//Для кривой дракона 2-го порядка это выглядит следующим образом:

//а) 1; б) 11; в) 110.

//Для кривой дракона 3-го порядка это выглядит следующим образом:

//а) 110; б) 1101; в) 1101100.

//Для кривой дракона 4-го порядка это выглядит следующим образом:

//а) 1101100; б) 11011001; в) 110110011100100.

//Жирным шрифтом выделена последовательность цифр, предшествующая

//приписанной единице, в которой затем средняя единица меняется на ноль, и эта

//последовательность приписывается справа.

//Кривые строятся от хвоста к голове дракона и повёрнуты так, чтобы

//драконы «плыли» направо, а пасть и кончик хвоста находились на одном

//уровне.

//Вариант 4. Получить кривую дракона 6-го порядка. Кривая изображается

//комбинированной линией. Голова дракона рисуется точечной линией.

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;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private string CreateDragonLine()

{

// задаем порядок кривой дракона 1101100111001001110110001100100111001100111001000110110001100100

int orderCurve = 8;

// задаем кривую дракона 5 порядка

string str = "1101100111001001110110001100100";

for (int i = 1; i < orderCurve; i++)

{ // создаем на базе строки предыдущего порядка изменяемую

//строку символов

StringBuilder sb = new StringBuilder(str);

// находим индекс центральной цифры

int seredina = (int)Math.Floor((double)sb.Length / 2);

// меняем символ с этим индексом на 0

sb[seredina] = '0';

// формируем строку нового порядка кривой дракона

str = str + "1" + sb;

}

return str;

}

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

{

} // конец метода обработки события клика на кнопку

private void Form1\_Load(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

Graphics g = pictureBox1.CreateGraphics();

// g.PageUnit = GraphicsUnit.Point;

Pen p = new Pen(Color.Red);

p.DashStyle = System.Drawing.Drawing2D.DashStyle.Dash;

int dx = 6;

// Формируем строку кривой дракона

string str = CreateDragonLine();

// начальная точка первой линии

int x1 = pictureBox1.Size.Width / 2;

int y1 = pictureBox1.Size.Height / 2;

// конечная точка первой линии

int x2 = pictureBox1.Size.Width / 2;

int y2 = pictureBox1.Size.Height / 2 - dx;

// сохраняем координаты конечной точки

int x3 = x2; int y3 = y2;

// рисуем линию из начальной в конечную точку

g.DrawLine(p, x1, y1, x3, y3);

// Цикл по всем цифрам кривой дракона

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

{

if (y2 - y1 < 0)

{ // рисовали вверх на предыдущем шаге

if (str[i] == '1') x3 = x2 - dx; // поворот налево

else x3 = x2 + dx; // поворот направо

y3 = y2;

}

if (x2 - x1 < 0)

{ // рисовали влево на предыдущем шаге

if (str[i] == '1') y3 = y2 - dx; // поворот налево

else y3 = y2 + dx; // поворот направо

x3 = x2;

}

if (x2 - x1 > 0)

{ // рисовали вправо на предыдущем шаге

if (str[i] == '1') y3 = y2 + dx; // поворот налево

else y3 = y2 - dx; // поворот направо

x3 = x2;

}

if (y2 - y1 > 0)

{ // рисовали вниз на предыдущем шаге

if (str[i] == '1') x3 = x2 + dx; // поворот налево

else x3 = x2 - dx; // поворот направо

y3 = y2;

}

if (i == str.Length - 1)

{// цвет и стиль последней линии

g.DrawLine(p, x2, y2, x3, y3);

}

else

{ // цвет и стиль остальных линий

float[] dashValues = { 5, 2, 15, 4 };

Pen blackPen = new Pen(Color.Black, 5);

blackPen.DashPattern = dashValues;

g.DrawLine(blackPen, x2, y2, x3, y3);

}

// переприсваивание для следующего шага

x1 = x2; y1 = y2;

x2 = x3; y2 = y3;

} // конец цикла по цифрам кривой дракона

}

}

}

