**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет прикладной математики и физики

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 1**

Тема: Построение изображений 2D- кривых..

Студент: Шевчук П.В.

Группа: 80-304

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2018

1. **Постановка задачи**

Задание: Написать и отладить программу, строящую изображение заданной замечательной кривой.

*Вариант**15*:

,

- полярные координаты; a, A и B - константы, значения которых выбираются пользователем (вводится в окне программы), a > 0.

Обеспечить автоматическое масштабирование и центрирование кривой при изменении размеров окна .

1. **Решения задачи**

Язык программирования:С#, класс Graphics.

Выбранный язык удобен для решения задачи, т.к. имеется мощный элемент управления DrawBox для отображения на экране графической информации. Построение графика (диаграммы) производится после вычисления таблицы значений функции y=f(x) с заданным шагом. Полученная таблица передается в специальный массив Points объекта Series элемента управления DrawBox с помощью метода DataBindXY. Элемент управления DrawBox осуществляет всю работу по отображению графиков: строит и размечает оси, рисует координатную сетку, подписывает название осей и самого графика, отображает переданную таблицу в виде всевозможных графиков или диаграмм. В элементе управления DrawBox можно настроить толщину, стиль и цвет линий, параметры шрифта подписей, шаги разметки координатной сетки и многое другое.

1. **Руководство по использованию программы**

Пользователь в окне вводит параметр *а*, который влияет на вытянутость кривой. Задаются значения границ угла А и В. Для вывода графика на экран необходимо нажать кнопку *draw.* Для увеличения масштаба нужно воспользоваться колесиком мышки. Если пользователь вводит некорректные данные, то будет появится окно с ошибкой.

Вводится параметр а, границы А и В угла. На рисунке 1 параметр а = 50 и границы A = -3 и B = 3. На рисунке 2 параметр а = 30 и границы A = 0 и B = 2.

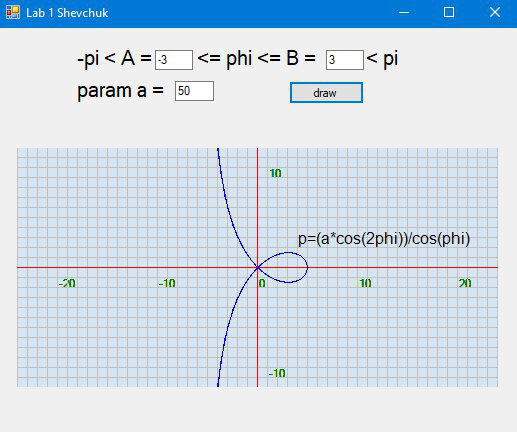


Рис. 1

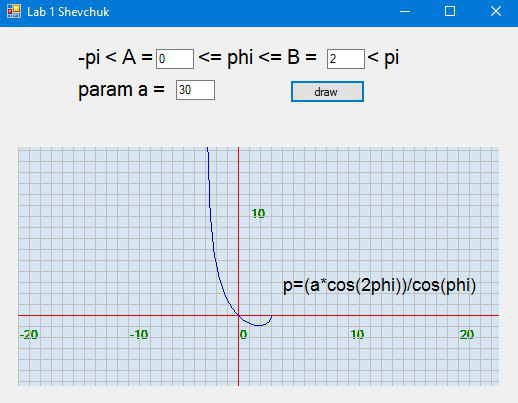


Рис. 2

**4. Листинг программы**

Form1.cs

// Шевчук П.В. М80-304Б-16

// Вариант № 15

// Построение кривой p=(a\*cos(2 phi))/cos(phi),

// С условием -pi < A <= x <= B < pi

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace lab1

{

public partial class Form1 : Form

{

// Глобальные переменные

float scale = 1; // масштаб

Point zero; // центр

bool drawcurve = false;

Graphics g;

bool mousePress = false;

Point mousePressPoint; // начальное положение

// Центр

private Point CalcCentr()

{

return new Point(drawBox.Width / 2, drawBox.Height / 2);

}

// Конструктор

public Form1()

{

InitializeComponent();

zero = CalcCentr();

drawBox.MouseWheel += DrawBox\_MouseWheel;

}

// Моя функция

private double myFunc(double a, double t)

{

return a \* Math.Cos(2 \* t) / Math.Cos(t);

}

// Построить оконо с графиком по функциям

private void drawBox\_Paint(object sender, PaintEventArgs e)

{

g = e.Graphics;

g.ScaleTransform(scale, scale);

g.TranslateTransform(zero.X + offset.X, zero.Y + offset.Y);

DrawCoordinates(g, Pens.Red);

if (drawcurve) DrawCurve(g);

}

int MAX = 1000000;

// Посторить кривую

private void DrawCurve(Graphics g)

{

Pen linePen = new Pen(Color.Blue, 1);

double aBorder;

double bBorder;

double paramA;

double step;

try

{

aBorder = Convert.ToDouble(textBox1.Text); // A

bBorder = Convert.ToDouble(textBox2.Text); // B

paramA = Convert.ToDouble(textBox3.Text); // a

step = 100; // approax

}

catch (Exception)

{

MessageBox.Show("Неверный формат ввода параметров");

return;

}

step /= 1000;

var points = new List<PointF>();

for (double i = aBorder; i < bBorder; i += step)

{

var r = myFunc(paramA, i); // Моя функция

PointF begin = ToDecart(r, i);

points.Add(begin);

}

if (points.Count < 2)

{

MessageBox.Show("Задан слишком большой шаг");

return;

}

var ARR = points.ToArray();

var last = ARR[0];

for (int i = 1; i < ARR.Length; i++)

{

if (ARR[i].Y - last.Y > 10000 \* step)

{

last = ARR[i];

continue;

}

g.DrawLine(linePen, last, ARR[i]);

last = ARR[i];

}

}

Point offset;

Point oldoffset;

// В декартовых координатах

private PointF ToDecart(double r, double d)

{

float x = (float)(r \* Math.Cos(d));

if (Math.Abs(x) < 0.001) x = 0;

if (Math.Abs(x) > 100000) x = 100000;

float y = (float)(r \* Math.Sin(d));

if (Math.Abs(y) < 0.001) y = 0;

if (Math.Abs(y) > 1000)

{

if (y >= 0)

y = 1000;

else

y = -1000;

}

return new PointF(x, y);

}

// Построение осей

private void DrawCoordinates(Graphics g, Pen p)

{

Pen grid\_pen = new Pen(Color.Silver, 0.0001f);

for (int i = -1000; i < 1000; i += 10)

{

for (int j = -1000; j < 1000; j += 10)

{

g.DrawLine(grid\_pen, new Point(i, -j), new Point(i, j));

g.DrawLine(grid\_pen, new Point(-i, j), new Point(i, j));

if (i % 100 == 0)

{

g.DrawString((i / 10).ToString(), DefaultFont, Brushes.Green, new Point(i, 10));

if (i != 0)

{

g.DrawString((i / 10).ToString(), DefaultFont, Brushes.Green, new Point(10, -i));

}

}

g.DrawLine(p, new Point(-i, i), new Point(i, i));

}

}

g.DrawLine(p, new Point(-MAX, 0), new Point(MAX, 0));

g.DrawLine(p, new Point(0, -MAX), new Point(0, MAX));

}

void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

ControlPaint.DrawGrid(e.Graphics, e.ClipRectangle, new Size(10, 10), Color.White);

}

// Масштабирование

private void DrawBox\_MouseWheel(object sender, MouseEventArgs e)

{

if (e.Delta > 0)

scale \*= 1.1f;

else

{

scale \*= 0.9f;

}

drawBox.Invalidate();

}

// Прокрутка вверх

private void drawBox\_MouseDown(object sender, MouseEventArgs e)

{

mousePress = true;

mousePressPoint = e.Location;

oldoffset = new Point(offset.X, offset.Y);

}

// Прокрутка вниз

private void drawBox\_MouseUp(object sender, MouseEventArgs e)

{

mousePress = false;

//mousePressPoint = new Point();

}

// Просмотр графика

private void drawBox\_MouseMove(object sender, MouseEventArgs e)

{

if (mousePress)

{

int dx = e.X - mousePressPoint.X;

int dy = e.Y - mousePressPoint.Y;

offset = new Point(oldoffset.X + (int)(dx / scale), oldoffset.Y + (int)(dy / scale));

drawBox.Invalidate();

}

}

private void drawBox\_SizeChanged(object sender, EventArgs e)

{

zero = CalcCentr();

}

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

{

scale = 1;

drawcurve = !drawcurve;

offset = new Point();

drawBox.Invalidate();

}

**5. Выводы**

В ходе данной лабораторной работы №1 получены основные навыки работы с базовой графикой c#, изучена специфика работы с DrawBox. Этот ценный навык поможет выполнять следующие работы на 20 баллов.

Литература

1.Введение в Windows Forms [Электронный ресурс].

URL: <https://metanit.com/sharp/windowsforms/1.1.php>

2. Построение графиков с помощью DrawBox[Электронный ресурс].

URL: <https://studfiles.net/preview/5428660/page:16/>