

Лабораторно-практическое задание

№ 1

Сайфудинов Р.Р. 171-261

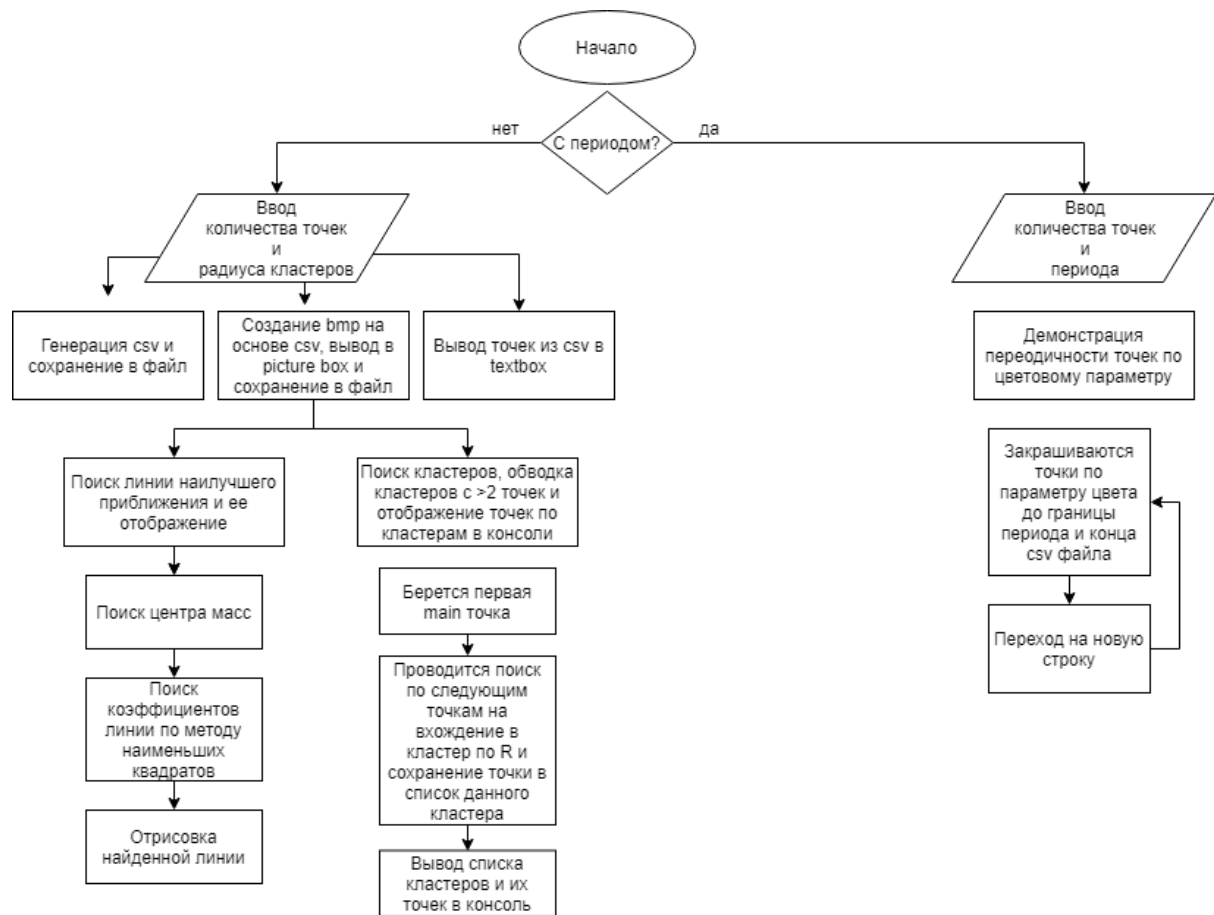
Тема: Освоение базовых инструментов обработки и хранения массивов данных на языке программирования.

Цель работы: Разработать базовые алгоритмы обмена данными с устройствами хранения, статистической обработки данных и визуализации результатов анализа.

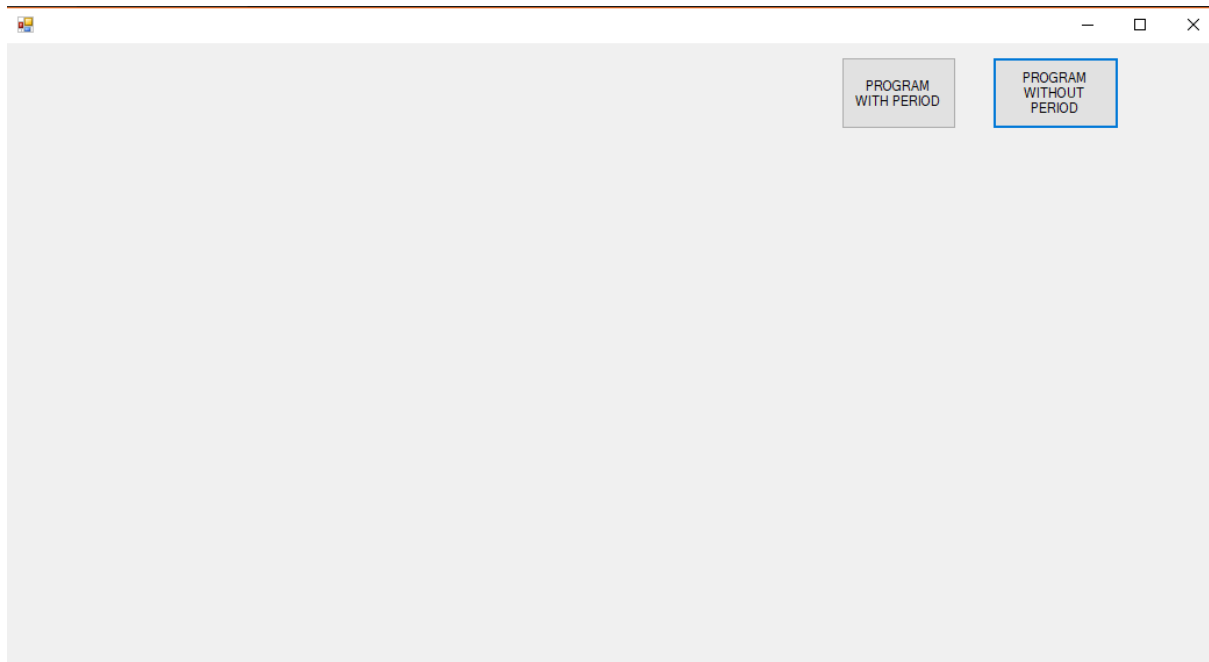
Задачи:

- Подготовить описание проектов задач для обработки данных;
- Подготовить модуль загрузки, сохранения и отображения массивов данных, хранимых в текстовом представлении (формат CSV);
- Подготовить модуль визуализации исходных (сырых) данных на двумерной плоскости в формате битовой карты (bitmap);
- Подготовить модуль визуализации исходных данных в виде линейного представления с заданным периодом;
- Рассчитать и отобразить на двумерной плоскости линию наилучшего приближения к данным, значение среднего и выполнить кластеризацию данных при фиксированном размере (радиусе) кластера.

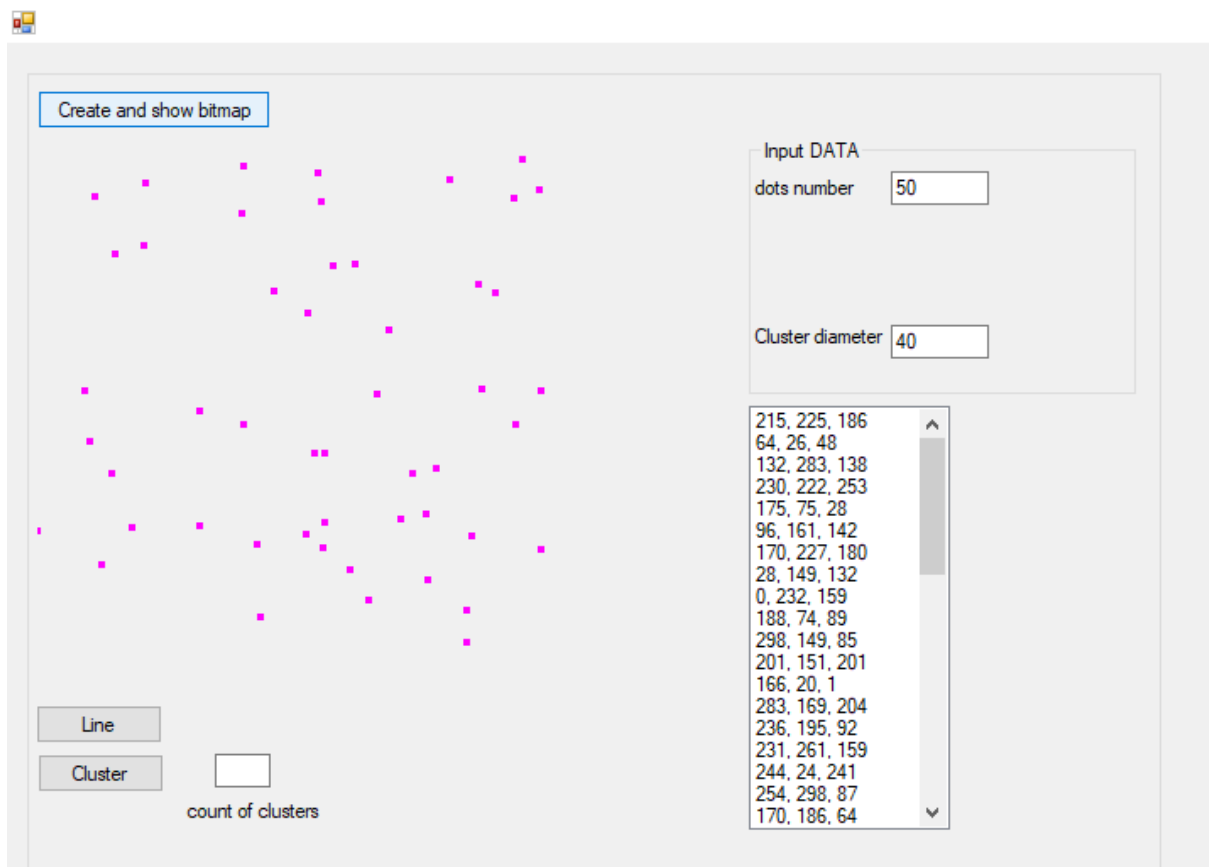
Блок схема



Интерфейс



Выбираем будем ли рисовать период или нет




Сгенерировали точки и отобразили в listbox



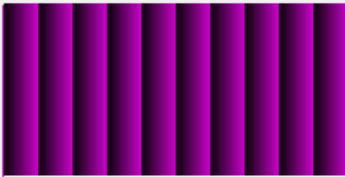
Ищем кластеры



Рисуем линию наилучшего приближения



Create and show bitmap



Period

Load to listbox

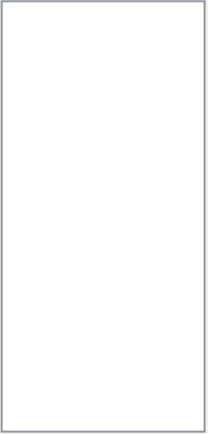
Input DATA

dots number

>5000

period

>100



Создаем период с рисунком

Код программы

```
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;

using System.IO;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;

namespace cv_2017_s1_l1
{
    public partial class Form1 : Form
    {
        int z = 0;
        public Form1() { InitializeComponent(); }
        private void Form1_Load(object sender, EventArgs e)
        {
            groupBox2.Visible = false;
        }
        //если хотим с кластерами
        private void button8_Click(object sender, EventArgs e)
        {
            groupBox2.Visible = true;
            button8.Visible = false;
            button7.Visible = false;
            textBox4.Visible = false;
            label3.Visible = false;
            button5.Visible = false;
            button6.Visible = false;
            textBox1.Visible = false;
            label4.Visible = false;

            z++;
        }
        //если хотим без кластеров
        private void button7_Click(object sender, EventArgs e)
        {
            groupBox2.Visible = true;
        }
    }
}
```

```

        button8.Visible = false;
        button7.Visible = false;
        button4.Visible = false;
        textBox2.Visible = false;
        label5.Visible = false;
        label6.Visible = false;
        label7.Visible = false;
    }

    /* СОЗДАЕМ CSV */
    private void button1_Click(object sender, EventArgs e)
    {
        Random random = new Random();

        // Generate new csv file and Write random numbers to csv
file
        using (StreamWriter sw = File.CreateText(@"..\data.csv")) {
            try
            {
                int color1 = 0;
                for (int i = 0; i < int.Parse(textBox3.Text); i++)
                {
                    int x = random.Next(0, 299);
                    int y = random.Next(0, 299);
                    if (z == 1)
                    {
                        color1 += 10;

                        sw.WriteLine(x + ", " + y + ", " + color1 +
" ");

                        if (color1 == 200) { color1 = 0; }
                    }
                    else if (z==0)
                    {
                        int color2 = random.Next(0, 255);
                        sw.WriteLine(x + ", " + y + ", " + color2 +
" ");

                        if (color2 == 200) { color2 = 0; }
                    }
                }
            }
            catch (Exception)
            {
                MessageBox.Show("YOU HAVE TO INPUT DOTS; RESTART THE
PROGRAM", "ERROR", MessageBoxButtons.OK,

```

```

        MessageBoxIcon.Information);
    }
}
button1.Visible = false;
}

/*ЗАГРУЖАЕМ CSV В LISTBOX */
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        // File path
        string path = @".\data.csv";
        // Upload file
        using (StreamReader sr = File.OpenText(path))
        {
            string s;
            while ((s = sr.ReadLine()) != null)
            {
                listBox1.Items.Add(s);
            }
        }
        button2.Visible = false;
    }
    catch (Exception)
    {
        MessageBox.Show("YOU HAVE TO CREATE CSV; RESTART THE
PROGRAM", "ERROR", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}

/* СОЗДАЕМ BITMAP */
private void button3_Click(object sender, EventArgs e)
{
    Random random = new Random();

    // Read the contents of CSV file
    string csv = File.ReadAllText(@".\data.csv");
    // Get the BMP file
    StreamWriter picture = File.CreateText(@".\data.bmp");
    picture.Close();
    Bitmap bmp = new Bitmap(300, 300);

```



```

int perX = 0, perY = 0;
foreach (string row in csv.Split('\n'))
{
    if (perX < 299) { perX++; }
    else if (perY < 299) { perY++; perX = 0; }
    if (!string.IsNullOrEmpty(row))
    {
        int x = 0, y = 0, r = 0;
        int i = 0;
        // take every number in row
        foreach (string cell in row.Split(','))
        {
            switch (i)
            {
                case 0:
                    x = int.Parse(cell);
                    i++;
                    break;
                case 1:
                    y = int.Parse(cell);
                    i++;
                    break;
                case 2:
                    r = int.Parse(cell);
                    i++;
                    break;
                default:
                    i++;
                    break;
            }
        }

        Color color = Color.FromArgb(r, 0, 0);
        var graphics = Graphics.FromImage bmp);
        graphics.FillRectangle(new
SolidBrush(Color.FromArgb(255, 0, 255)), x-2, y-2, 4, 4);
    }
}
pictureBox1.Image = bmp;
bmp.Save(@"..\data.bmp");
}

/* ПЕРИОД */
private void button4_Click(object sender, EventArgs e)

```

```

{
    try {
        int endX = Int32.Parse(textBox2.Text);
        //Random initialize
        Random random = new Random();

        // Read the contents of CSV file
        string csv = File.ReadAllText(@"..\data.csv");

        // Get the BMP file
        StreamWriter picture = File.CreateText(@"..\data.bmp");
        picture.Close();
        Bitmap bmp = new Bitmap(300, 300);
        int x = 0, y = 0;
        foreach (string row in csv.Split('\n'))
        {
            if (x < endX - 1) { x++; }
            else if (y < 299) { y++; x = 0; }
            if (!string.IsNullOrEmpty(row))
            {
                int r = 0;
                int b = 0;
                int i = 0;
                // take every number in row
                foreach (string cell in row.Split(','))
                {
                    switch (i)
                    {
                        case 0:
                            i++;
                            break;
                        case 1:
                            i++;
                            break;
                        case 2:
                            r = int.Parse(cell);
                            b = int.Parse(cell);
                            i++;
                            break;
                        default:
                            i++;
                            break;
                    }
                }
                // draw pixel in (x, y) with color (r, g, b)
                Color color = Color.FromArgb(r, 0, b);
            }
        }
    }
}

```

```

        bmp.SetPixel(x, y, color);
    }
}
pictureBox1.Image = bmp;
bmp.Save(@"data.bmp");
}
catch (Exception)
{
    MessageBox.Show("YOU HAVE TO INPUT CORRECT PERIOD;  
RESTART THE PROGRAM", "ERROR", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
}
}

/*ЛИНИЯ НАИЛУЧШЕГО ПРИБЛИЖЕНИЯ */
private void button5_Click(object sender, EventArgs e)
{
    // Read the contents of CSV file
    string csv = File.ReadAllText(@"data.csv");

    // Calculate sum for all X, Y, X^2, X*Y and count points
    int sumX = 0, sumXX = 0, sumY = 0, sumXY = 0, points = -1;
    int X = 0, Y = 0;
    foreach (string row in csv.Split('\n'))
    {
        int i = 0;
        points++;
        foreach (string cell in row.Split(','))
        {
            if (cell != "")
            {
                switch (i)
                {
                    case 0:
                        X = int.Parse(cell);
                        sumX += X;
                        sumXX += X * X;
                        i++;
                        break;
                    case 1:
                        Y = int.Parse(cell);
                        sumY += Y;
                        sumXY += X * Y;
                        i++;

```

```

        break;
    default:
        i++;
        break;
    }
}
}

// Calculate average X and Y and X^2

double avX = (double)sumX / (double)points;
double avY = (double)sumY / (double)points;
double avXX = (double)sumXX / (double)points;
double avXY = (double)sumXY / (double)points;

// Calculate a and b coefficients for the line (y = ax + b)
double a = (avXY - avY * avX) / (avXX - avX * avX); // a =
( (XY)̄ - Ȳ * X̄ ) / ( (X^2)̄ - (X̄)^2 )
double b = avY - a * avX;

try {
    // Draw the line
    Bitmap bmp = new Bitmap(pictureBox1.Image);
    Graphics g;
    g = Graphics.FromImage(bmp);
    Pen pen = new Pen(Color.Blue);
    pen.Width = 3;

    // Draw average X, Y lines
    g.DrawLine(Pens.Cyan, Convert.ToInt32(avX), 0,
Convert.ToInt32(avX), 300);
    g.DrawLine(Pens.Cyan, 0, Convert.ToInt32(avY), 300,
Convert.ToInt32(avY));

    // Prepare the line

    int X1 = -2000;
    int Y1 = Convert.ToInt32(a * X1 + b);

    int X2 = 2000;
    int Y2 = Convert.ToInt32(a * X2 + b);

    g.DrawLine(pen, X1, Y1, X2, Y2); // x1, y1, x2, y2
    pictureBox1.Image = bmp;
}

```



```

                s++;
                break;
            }
        }
    }
    k++;
}

try
{

    // Find and draw clusters
    Bitmap bmp_clusters = new Bitmap(pictureBox1.Image);

    Graphics g;
    g = Graphics.FromImage(bmp_clusters);

    int X = 0, Y = 0, C = 0;
    int S = Convert.ToInt32(textBox4.Text);
    int Size = S % 5 < 2.5 ? S -= S % 5 : S = S - S % 5 + 5;
    int i = 0;
    while (i <= points)
    {
        if (!pointUsed[i])
        {
            int main_x = pointList[i, 0];
            int main_y = pointList[i, 1];
            int main_c = pointList[i, 2];
            int z = 0;
            int j = i + 1;
            while (j <= points)
            {

                if (!pointUsed[j])
                {
                    X = pointList[j, 0];
                    Y = pointList[j, 1];
                    C = pointList[j, 2];
                    if ((Math.Pow(main_x - X, 2) +
Math.Pow(main_y - Y, 2)) <= Math.Pow(S / 2, 2))
                    {
                        z++;
                        if (z == 1)
                        {
                            g.FillRectangle(new
SolidBrush(Color.Green), X - 2, Y - 2, 4, 4);

```

```

        g.FillRectangle(new
SolidBrush(Color.Green), main_x - 2, main_y - 2, 4, 4);
        g.DrawEllipse(new
Pen(Color.Green, 2), main_x - S / 2, main_y - S / 2, S, S);
        pictureBox1.Image =
bmp_clusters;

        pointUsed[j] = true;
        pointUsed[i] = true;
        count++;
        g.TextRenderingHint =
System.Drawing.Text.TextRenderingHint.AntiAlias;
        g.DrawString(count.ToString(),
new Font("Cambria", Size / 5), Brushes.Black, new PointF(main_x - Size /
5 + 2, main_y - Size / 5 - Size / 10));
    }
    else
    {
        g.FillRectangle(new
SolidBrush(Color.Green), X - 2, Y - 2, 4, 4);
        g.FillRectangle(new
SolidBrush(Color.Green), main_x - 2, main_y - 2, 4, 4);
    }

}

}
j++;
}
}
i++;
}
g.Dispose();
textBox1.Text = count.ToString();
}
catch (Exception)
{
    MessageBox.Show("YOU HAVE TO CREATE BITMAP OR INPUT
RADIUS; RESTART THE PROGRAM", "ERROR", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

}

}}

```

