Лабораторная работа №12-13.

Выполнил: Новиков Даниил

Группа: ФИб-2301

1 задание.

Реализация алгоритма Грэхема.

Код:

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 Task1

{

public partial class Form1 : Form

{

Random rand;

Graphics graph;

Bitmap bmp;

List<PointF> points;

public static PointF startPoint;

public Form1()

{

InitializeComponent();

bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

graph = Graphics.FromImage(bmp);

points = new List<PointF>();

rand = new Random();

}

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

{

if (points.Count() > 2)

{

DrawShell(points.Count());

}

else MessageBox.Show("Добавьте точки");

}

private void DrawShell(int n)

{

var sortPoints = points.OrderBy(x => x.Y).ThenBy(x => x.X).ToList();

startPoint = sortPoints[0];//точка с минимальным Y и минимальным X

sortPoints.Sort(1, n - 1, new CompareByPointF());

int m = 1;

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

{

while (i < n - 1 && ClockWise(startPoint, sortPoints[i], sortPoints[i + 1]) == 0)//если име.т одинаковый угол, то не берем их в расчет

i++;

sortPoints[m++] = sortPoints[i];

}

if (m < 3) return;//оболочку невозможно построить

Stack<PointF> fs = new Stack<PointF>();//с помощью него будем искать точки оболочки

fs.Push(startPoint);

fs.Push(sortPoints[1]);

fs.Push(sortPoints[2]);

for (int i = 3; i < m; i++)

{

while (ClockWise(NextTop(ref fs), fs.Peek(), sortPoints[i]) != -1)

fs.Pop();

fs.Push(sortPoints[i]);

}

var polygon = fs.ToArray();

graph.DrawPolygon(new Pen(Color.Black, 1), polygon);

pictureBox1.Image = bmp;

}

private PointF NextTop(ref Stack<PointF> fs)

{

PointF p = fs.Peek();

fs.Pop();

PointF res = fs.Peek();

fs.Push(p);

return res;

}

private void pictureBox1\_MouseClick(object sender, MouseEventArgs e)

{

points.Add(new PointF(e.X, e.Y));

graph.FillEllipse(Brushes.Blue, e.X, e.Y, 3, 3);

pictureBox1.Image = bmp;

}

private int Length(PointF a, PointF b) => (int)(Math.Pow(b.X - a.X, 2) + Math.Pow(b.Y - a.Y, 2));

private int ClockWise(PointF a, PointF b, PointF c)

{

int val = (int)((b.Y - a.Y) \* (c.X - b.X) - (b.X - a.X) \* (c.Y - b.Y));

if (val == 0) return 0;

return (val > 0) ? 1 : -1; // по часовой или против часовой стрелки

}

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

{

points.Clear();

graph.Clear(pictureBox1.BackColor);

pictureBox1.Image = bmp;

}

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

{

points = Enumerable.Range(0, rand.Next(40) + 15).Select(x => new PointF(rand.Next(bmp.Width), rand.Next(bmp.Height))).ToList();

points.ForEach(x => graph.FillEllipse(Brushes.Blue, x.X, x.Y, 3, 3));

pictureBox1.Image = bmp;

}

}

class CompareByPointF : IComparer<PointF>

{

public int Length(PointF a, PointF b) => (int)(Math.Pow(b.X - a.X, 2) + Math.Pow(b.Y - a.Y, 2));

public int ClockWise(PointF a, PointF b, PointF c)

{

int val = (int)((b.Y - a.Y) \* (c.X - b.X) - (b.X - a.X) \* (c.Y - b.Y));

if (val == 0) return 0;

return (val > 0) ? 1 : -1; // по часовой или против часовой стрелки

}

#region IComparer<PointF> Points

public int Compare(PointF x, PointF y)

{

int orient = ClockWise(Form1.startPoint, x, y);

if (orient == 0)

return (Length(Form1.startPoint, y) >= Length(Form1.startPoint, x)) ? -1 : 1;

return orient;

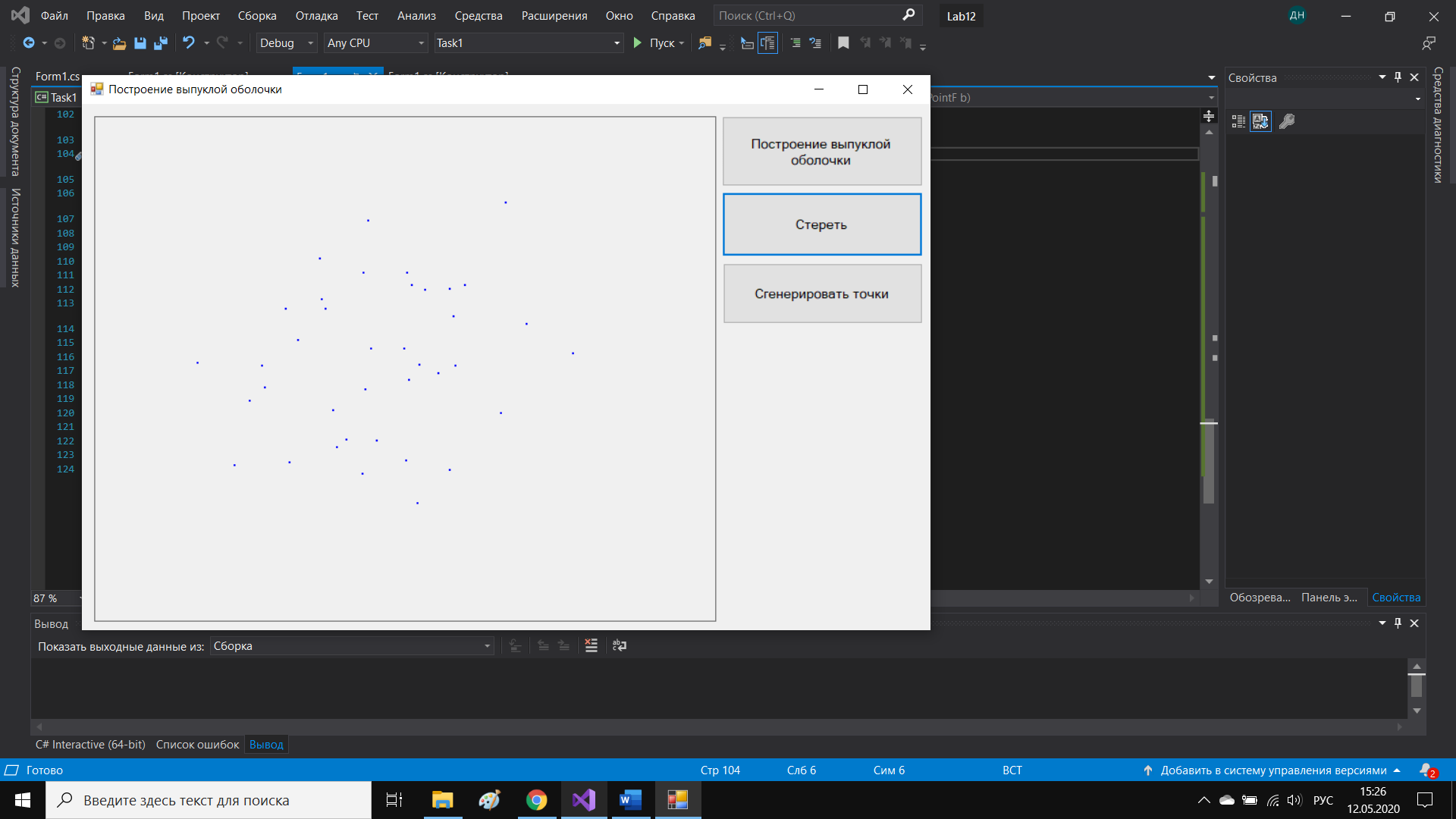
}

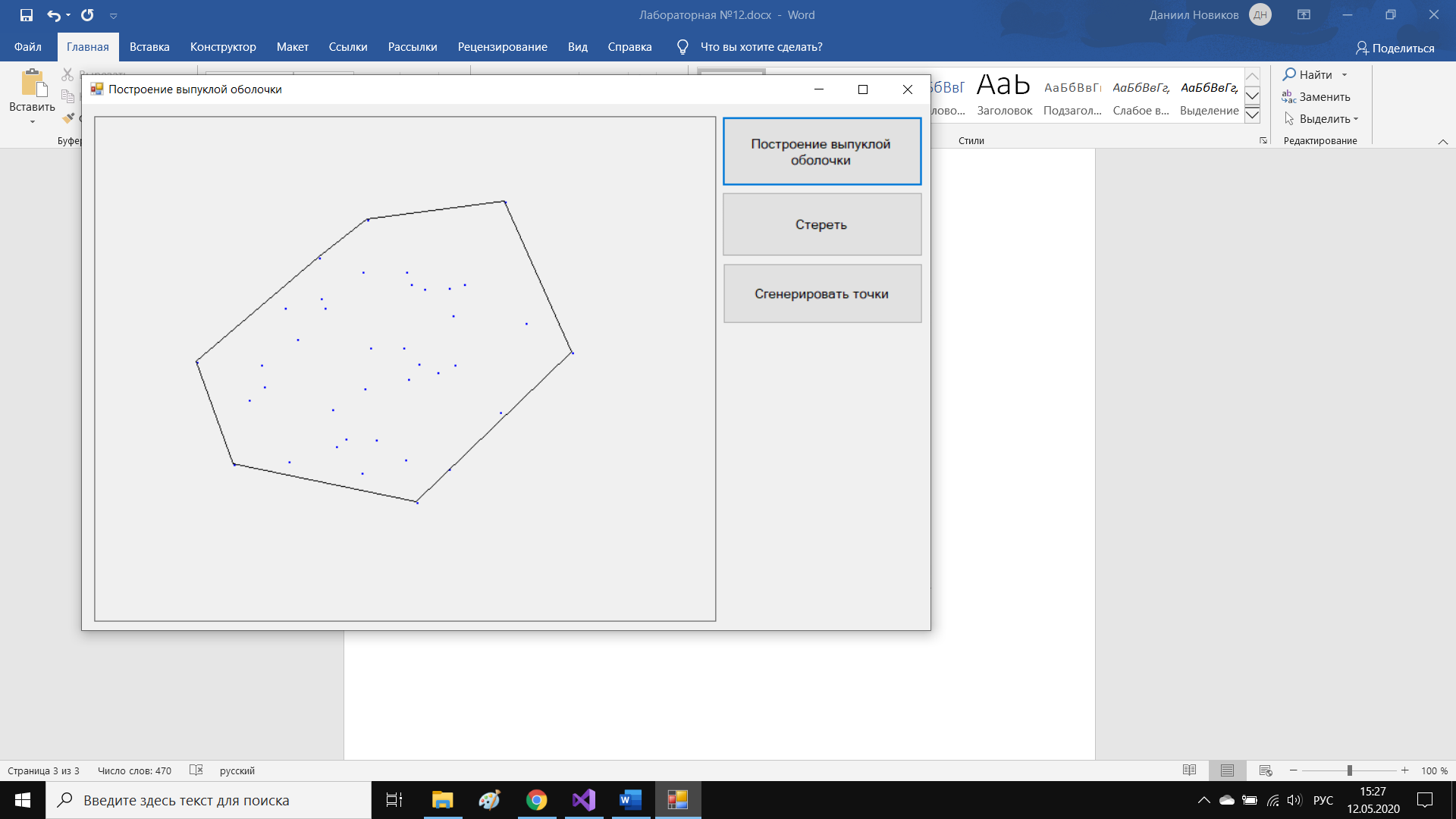
#endregion

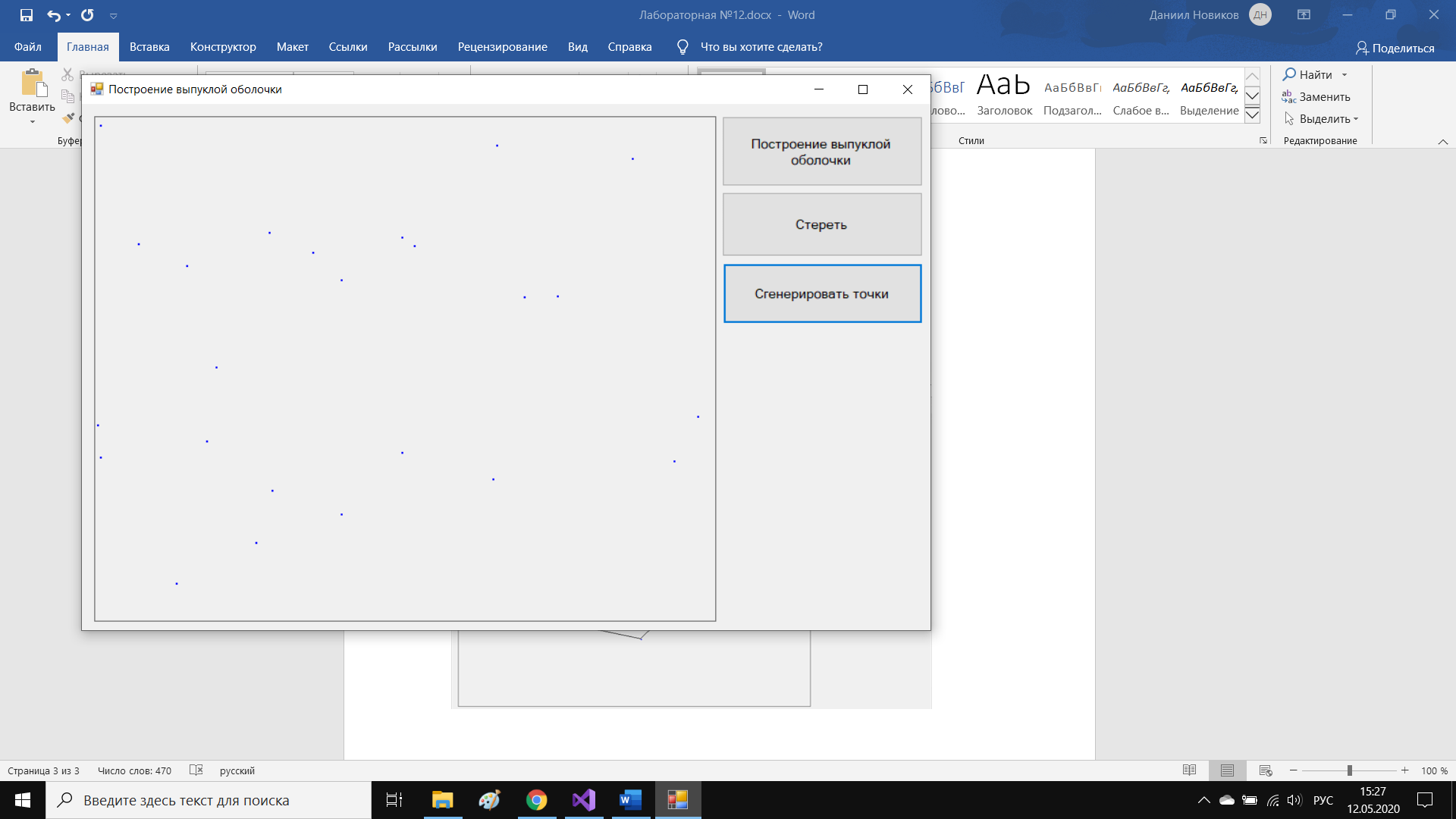
}

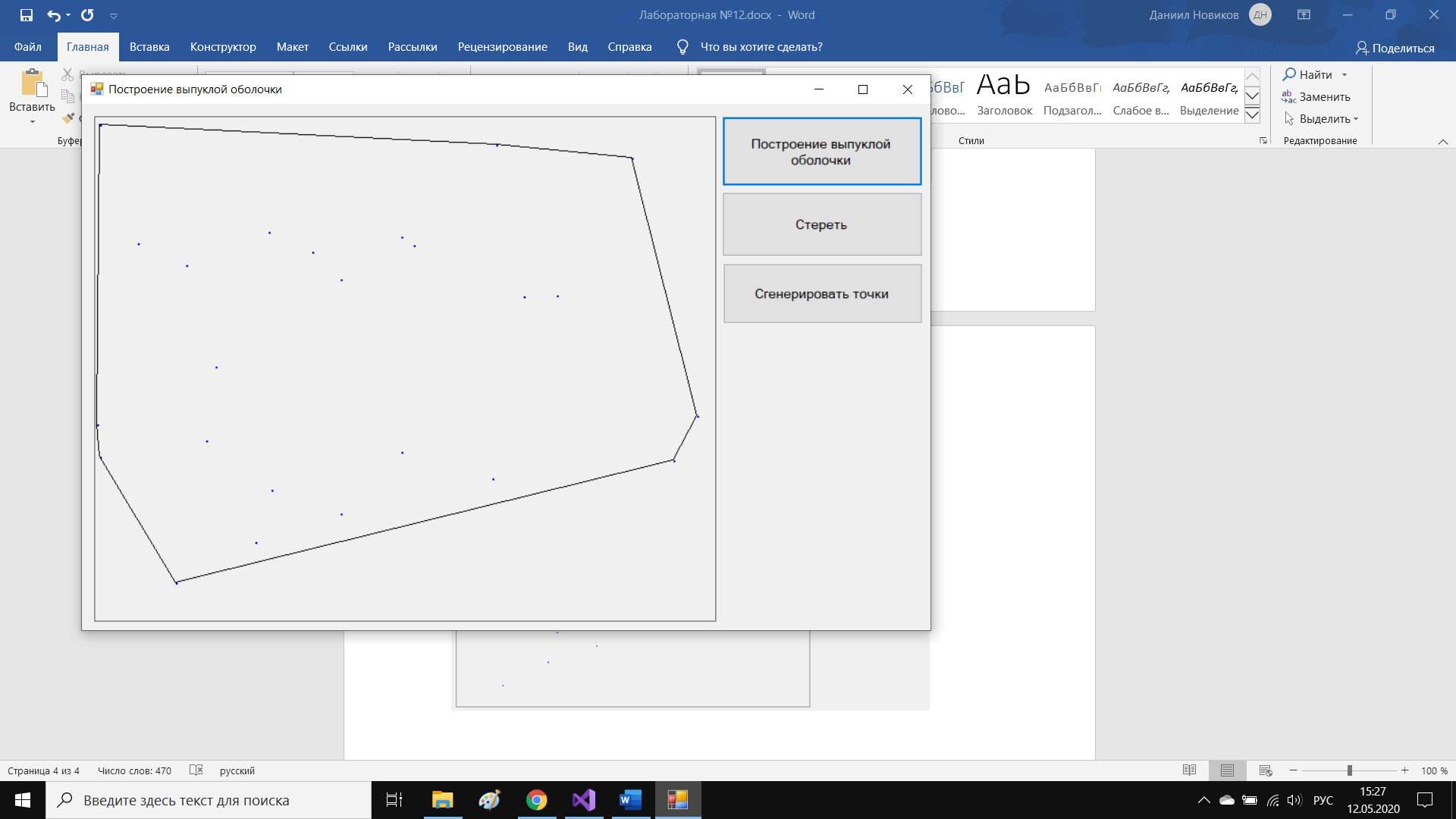
}

Результаты работы программы:









2 задание.

Реализация алгоритма Джарвиса.

Код:

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 Task2

{

public partial class Form1 : Form

{

Random rand;

Graphics graph;

Bitmap bmp;

List<PointF> points;

public static PointF startPoint;

public Form1()

{

InitializeComponent();

bmp = new Bitmap(pictureBox1.Width, pictureBox1.Height);

graph = Graphics.FromImage(bmp);

points = new List<PointF>();

rand = new Random();

}

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

{

if (points.Count() > 2)

{

DrawShell(points.Count());

}

else MessageBox.Show("Добавьте точки");

}

private void DrawShell(int n)

{

int minIndex = 0;

for (int i = 1; i < n; i++)//находим точку с минимальной координатой по X

if (points[i].X < points[minIndex].X)

minIndex = i;

int first = minIndex, second;

var polygon = new List<PointF>();

do

{

polygon.Add(points[first]);

second = (first + 1) % n;//чтобы не уйти за границы, применяем операцию остатка от деления

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

if (ClockWise(points[first], points[i], points[second]) == -1)//если против часовой стрелки

second = i;

first = second;

} while (first != minIndex);

graph.DrawPolygon(new Pen(Color.Black, 1), polygon.ToArray());

pictureBox1.Image = bmp;

}

private int ClockWise(PointF a, PointF b, PointF c)

{

int val = (int)((b.Y - a.Y) \* (c.X - b.X) - (b.X - a.X) \* (c.Y - b.Y));

if (val == 0) return 0;

return (val > 0) ? 1 : -1; // по часовой или против часовой стрелки

}

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

{

points.Clear();

graph.Clear(pictureBox1.BackColor);

pictureBox1.Image = bmp;

}

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

{

points = Enumerable.Range(0, rand.Next(40) + 15).Select(x => new PointF(rand.Next(bmp.Width), rand.Next(bmp.Height))).ToList();

points.ForEach(x => graph.FillEllipse(Brushes.Blue, x.X, x.Y, 3, 3));

pictureBox1.Image = bmp;

}

private void pictureBox1\_MouseClick(object sender, MouseEventArgs e)

{

points.Add(new PointF(e.X, e.Y));

graph.FillEllipse(Brushes.Blue, e.X, e.Y, 3, 3);

pictureBox1.Image = bmp;

}

}

}

Результаты работы программы:

