Лабораторная работа №6

Выполнил: Новиков Даниил

Группа: ФИб-2301

Задание 3.1.1

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;

namespace \_3.\_1.\_1

{

public partial class Form1 : Form

{

Graphics graph;

Brush brush;

Random rand;

Pen pen;

int x, y;

int xMax, yMax;

int var;

Point[] rectangle;//прямоугольник для первого варианта

struct Ellipse//круг для второго варианта

{

public Point center;

public int R;

}

Ellipse ellipse;

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

rand = new Random();

pen = new Pen(Color.Black, 2);

rectangle = new Point[4] { new Point(80, 80), new Point(ClientSize.Width - 80, 80), new Point(ClientSize.Width - 80, ClientSize.Height - 80), new Point(80, ClientSize.Height - 80) };

ellipse = new Ellipse();

ellipse.R = 100;

ellipse.center = new Point(ClientSize.Width / 2, ClientSize.Height / 2);

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Escape)

Close();

}

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

{

graph.Clear(BackColor);

var = int.Parse(VarTextBox.Text);

if (var == 1)

{

xMax = rectangle[1].X - rectangle[0].X;

yMax = rectangle[3].Y - rectangle[0].Y;

}

else if (var == 2)

{

xMax = 2 \* ellipse.R;

yMax = 2 \* ellipse.R;

}

else if(var==3)

{

xMax = ClientSize.Width;

yMax = ClientSize.Height;

}

if(timer1.Enabled)timer1.Stop();

timer1.Start();

}

private void FirstVariant()//рисуем звезды в прямоугольнике

{

x = rand.Next(xMax); y = rand.Next(yMax);

graph.FillRectangle(brush, x + rectangle[0].X, y + rectangle[0].Y, 2, 2);

for (int j = 0; j < rand.Next(900); j++)

graph.FillRectangle(Brushes.White, x + rectangle[0].X - 15 + rand.Next(31), y + rectangle[0].Y - 15 + rand.Next(31), 2, 2);

}

private void SecondVariant()

{

do

{

x = rand.Next(xMax); y = rand.Next(yMax);

}

while (x \* x + y \* y >= ellipse.R \* ellipse.R);

graph.FillRectangle(brush, x + ellipse.center.X, y + ellipse.center.Y, 2, 2);

graph.FillRectangle(brush, -x + ellipse.center.X, y + ellipse.center.Y, 2, 2);

graph.FillRectangle(brush, x + ellipse.center.X, -y + ellipse.center.Y, 2, 2);

graph.FillRectangle(brush, -x + ellipse.center.X, -y + ellipse.center.Y, 2, 2);

for (int j = 0; j < rand.Next(900); j++)

{

graph.FillRectangle(Brushes.White, x + ellipse.center.X + rand.Next(31), y + ellipse.center.Y + rand.Next(31), 2, 2);

graph.FillRectangle(Brushes.White, -x + ellipse.center.X + rand.Next(31), y + ellipse.center.Y + rand.Next(31), 2, 2);

graph.FillRectangle(Brushes.White, x + ellipse.center.X + rand.Next(31), -y + ellipse.center.Y + rand.Next(31), 2, 2);

graph.FillRectangle(Brushes.White, -x + ellipse.center.X + rand.Next(31), -y + ellipse.center.Y + rand.Next(31), 2, 2);

}

}

private void ThirdVariant()//рисуем в нижнем треуголнике

{

do

{

x = rand.Next(xMax); y = rand.Next(yMax);

}

while (y <=ClientSize.Height-(double)ClientSize.Height/(double)ClientSize.Width\*x);

graph.FillRectangle(brush, x, y, 2, 2);

for (int j = 0; j < rand.Next(900); j++)

graph.FillRectangle(Brushes.White, x + rectangle[0].X - 15 + rand.Next(31), y + rectangle[0].Y - 15 + rand.Next(31), 2, 2);

}

private void DrawSkyInDiamonds()

{

int i = rand.Next(7) + 9;

switch (i)

{

case 9:

brush = Brushes.LightBlue;

break;

case 10:

brush = Brushes.LightGreen;

break;

case 11:

brush = Brushes.LightCyan;

break;

case 12:

brush = Brushes.Red;

break;

case 13:

brush = Brushes.Magenta;

break;

case 14:

brush = Brushes.Yellow;

break;

case 15:

brush = Brushes.White;

break;

}

switch (var)

{

case 1:

FirstVariant();

break;

case 2:

SecondVariant();

break;

case 3:

ThirdVariant();

break;

default:

break;

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (var == 1) graph.DrawPolygon(pen, rectangle);

else if (var == 2) graph.DrawEllipse(pen, ellipse.center.X - ellipse.R, ellipse.center.Y - ellipse.R, 2 \* ellipse.R, 2 \* ellipse.R);

else if (var == 3) graph.DrawLine(pen, ClientSize.Width, 0, 0, ClientSize.Height);

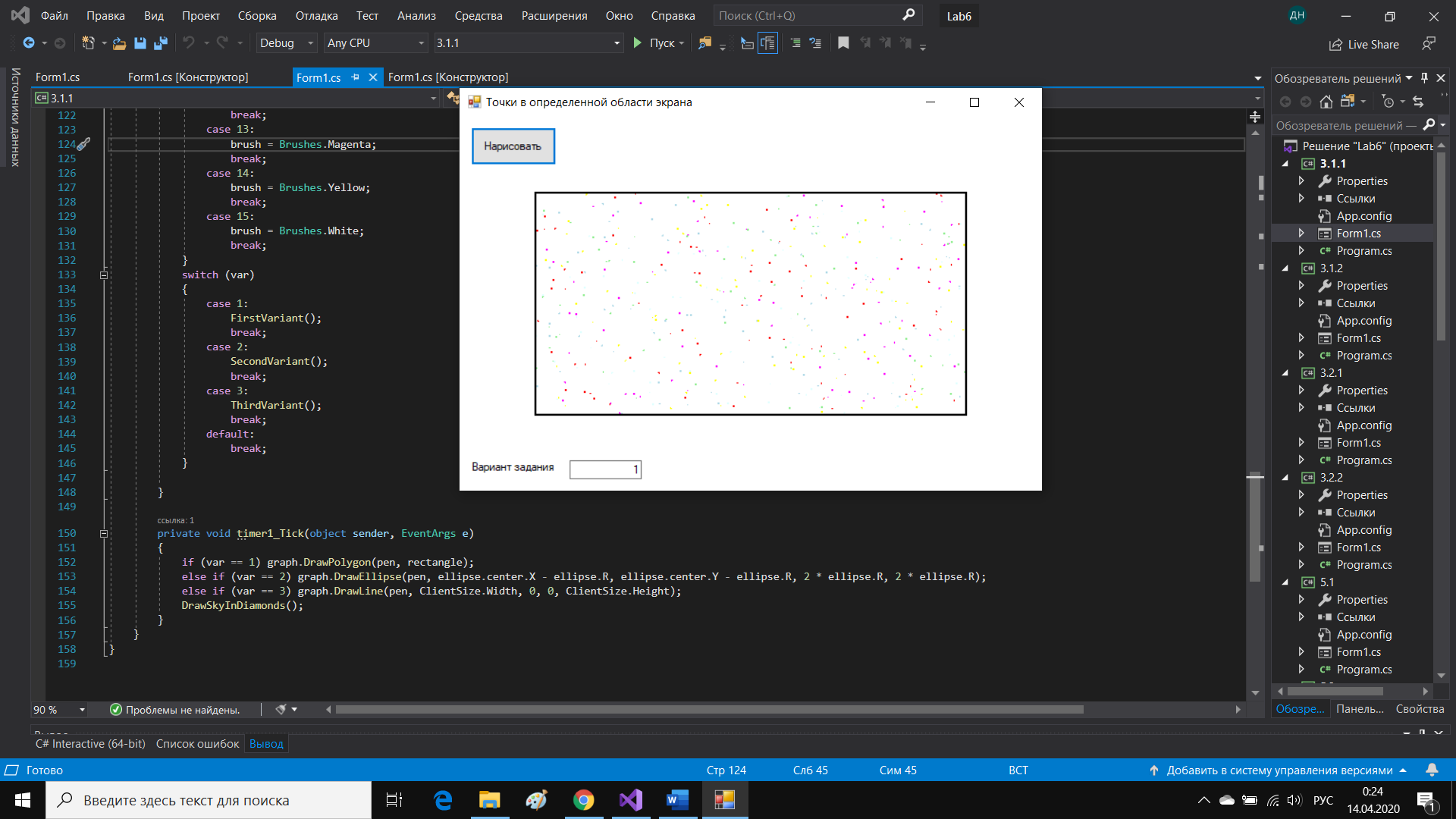
DrawSkyInDiamonds();

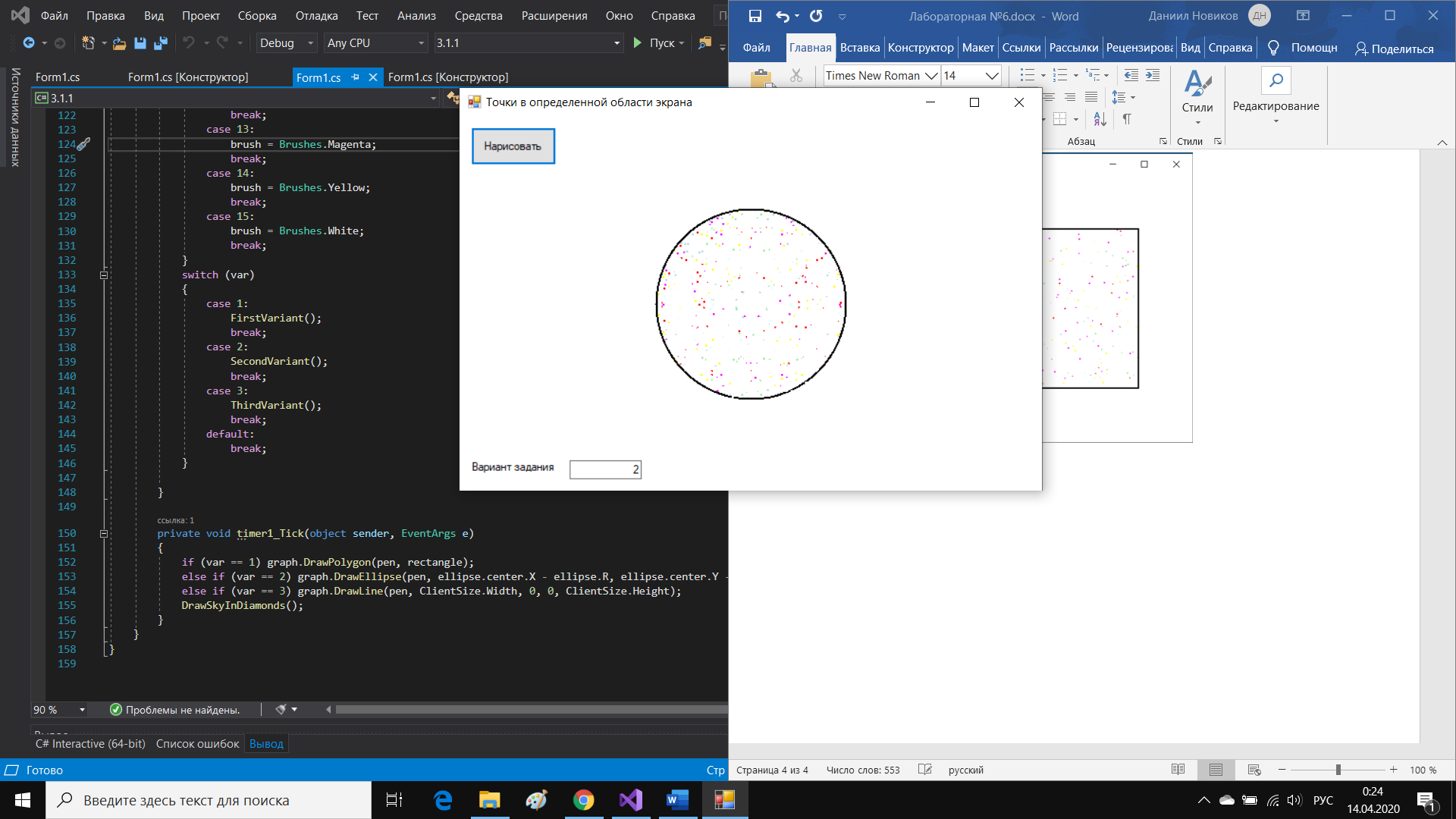
}

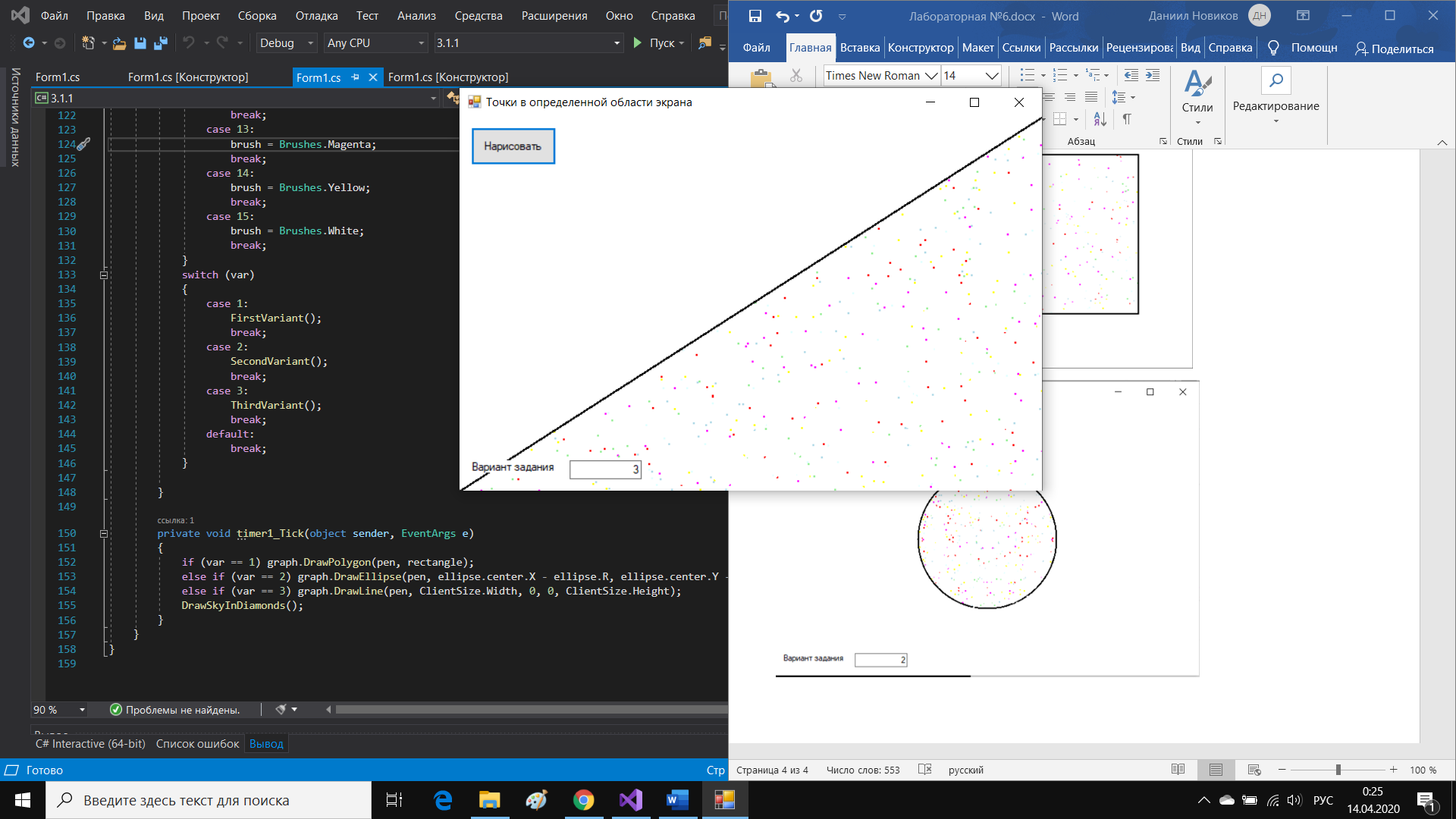
}

}

Результат работы программы:







Задание 3.1.2

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;

namespace \_3.\_1.\_2

{

public partial class Form1 : Form

{

Graphics graph;

Brush brush;

Random rand;

Pen pen;

int x, y;

int xMax, yMax;

int var;

public Form1()

{

InitializeComponent();

xMax = ClientSize.Width;

yMax = ClientSize.Height;

graph = CreateGraphics();

rand = new Random();

pen = new Pen(Color.Black, 2);

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Escape)

Close();

}

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

{

graph.Clear(BackColor);

var = int.Parse(VarTextBox.Text);

if (timer1.Enabled) timer1.Stop();

timer1.Start();

}

private void FirstVariant()//рисуем звезды в прямоугольнике

{

x = rand.Next(xMax);

y = rand.Next(yMax);

graph.DrawRectangle(new Pen(Color.FromArgb((int)(255.0 \* ((double)x / (double)xMax)), 0, 0, 0), 2), x, y, 1, 1);

for (int i=0;i<rand.Next(900);i++)

{

graph.FillRectangle(Brushes.White, x - 15 + rand.Next(31), y - 15 + rand.Next(31), 2, 2);

}

}

private void SecondVariant()

{

x = rand.Next(xMax);

y = rand.Next(yMax/2);

graph.DrawRectangle(new Pen(Color.FromArgb((int)(255.0 \* ((double)(-y+yMax/2) / (double)(yMax/2))), 0, 0, 0), 2), x, y, 1, 1);

graph.DrawRectangle(new Pen(Color.FromArgb((int)(255.0 \* ((double)(-y + yMax / 2) / (double)(yMax / 2))), 0, 0, 0), 2), x, y+2\*(yMax/2-y), 1, 1);//отзеркаливаем относительно yMax/2

x = rand.Next(xMax/2);

y = rand.Next(yMax);

graph.DrawRectangle(new Pen(Color.FromArgb((int)(255.0 \* ((double)(xMax/2-x) / (double)(xMax / 2))), 0, 0, 0), 2), x, y, 1, 1);

graph.DrawRectangle(new Pen(Color.FromArgb((int)(255.0 \* ((double)(xMax/2-x) / (double)(xMax / 2))), 0, 0, 0), 2), x+2\*(xMax/2-x), y, 1, 1);//отзеркаливаем относительно xMax/2

for (int i = 0; i < rand.Next(900); i++)

{

graph.FillRectangle(Brushes.White, x - 15 + rand.Next(31), y - 15 + rand.Next(31), 2, 2);

}

}

private void ThirdVariant()//рисуем в нижнем треуголнике

{

//левый верхний угол

x = rand.Next(xMax/2);

y = rand.Next(yMax / 2);

graph.DrawRectangle(new Pen(Color.FromArgb((int)(250.0 \* ((double)Math.Min(x,y) / (double)Math.Max(x,y))), 0, 0, 0), 2), x, y, 1, 1);

for (int i = 0; i < rand.Next(900); i++)

{

graph.FillRectangle(Brushes.White, x - 15 + rand.Next(31), y - 15 + rand.Next(31), 2, 2);

}

//нижний правый угол

x = rand.Next(xMax / 2);

y = rand.Next(yMax/2);

graph.DrawRectangle(new Pen(Color.FromArgb((int)(250.0 \* ((double)Math.Min(x, y) / (double)Math.Max(x, y))), 0, 0, 0), 2), x +2 \* (xMax / 2 - x), y + 2 \* (yMax / 2 - y), 1, 1);

//graph.DrawRectangle(new Pen(Color.FromArgb((int)(255.0 \* ((double)(xMax / 2 - x) / (double)(xMax / 2))), 0, 0, 0), 2), x + 2 \* (xMax / 2 - x), y, 1, 1);

for (int i = 0; i < rand.Next(900); i++)

{

graph.FillRectangle(Brushes.White, x + 2 \* (xMax / 2 - x) - 15 + rand.Next(31), y + 2 \* (yMax / 2 - y) - 15 + rand.Next(31), 2, 2);

}

}

private void DrawSkyInDiamonds()

{

brush = Brushes.Black;

switch (var)

{

case 1:

FirstVariant();

break;

case 2:

SecondVariant();

break;

case 3:

ThirdVariant();

break;

default:

break;

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

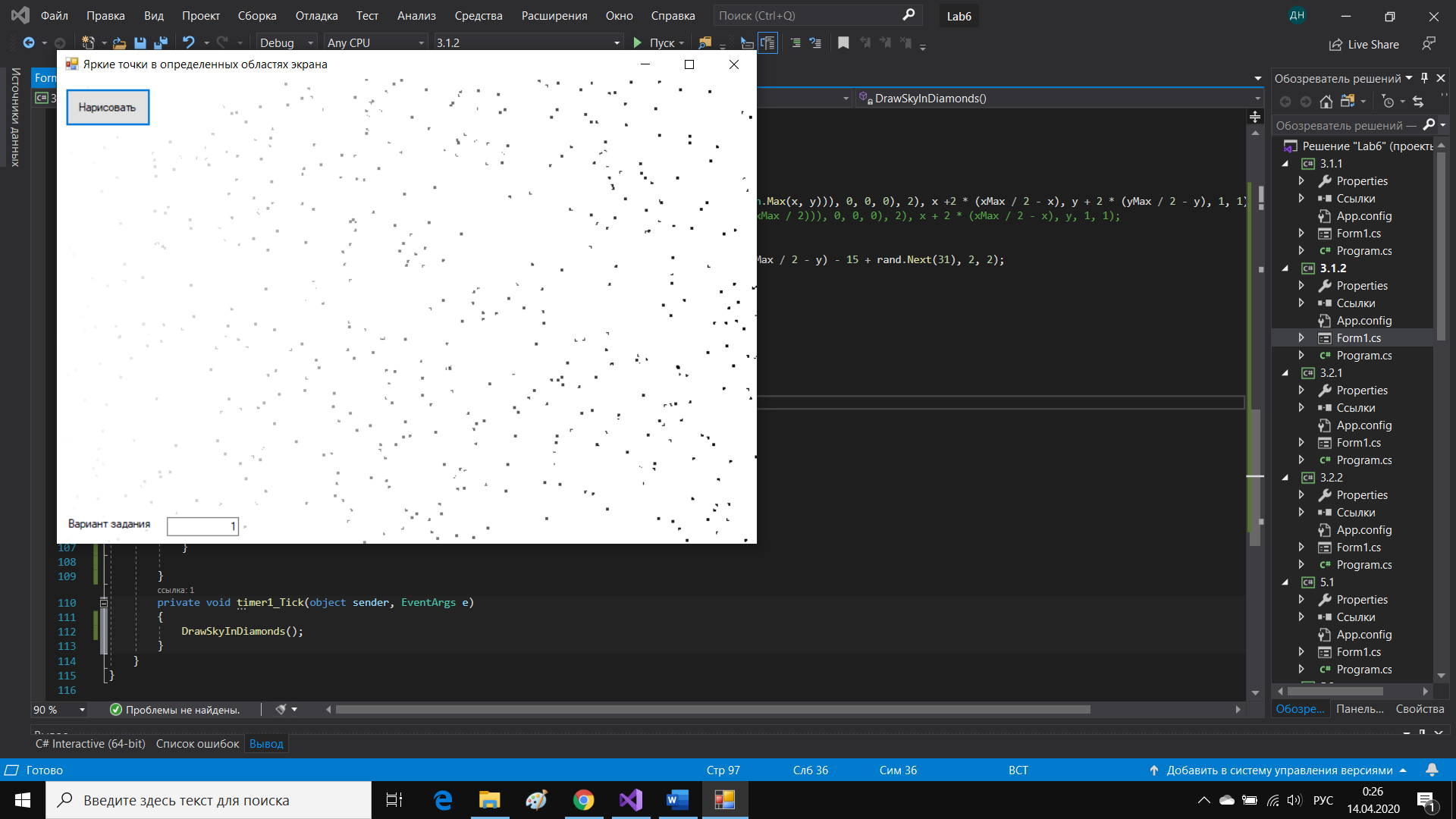
DrawSkyInDiamonds();

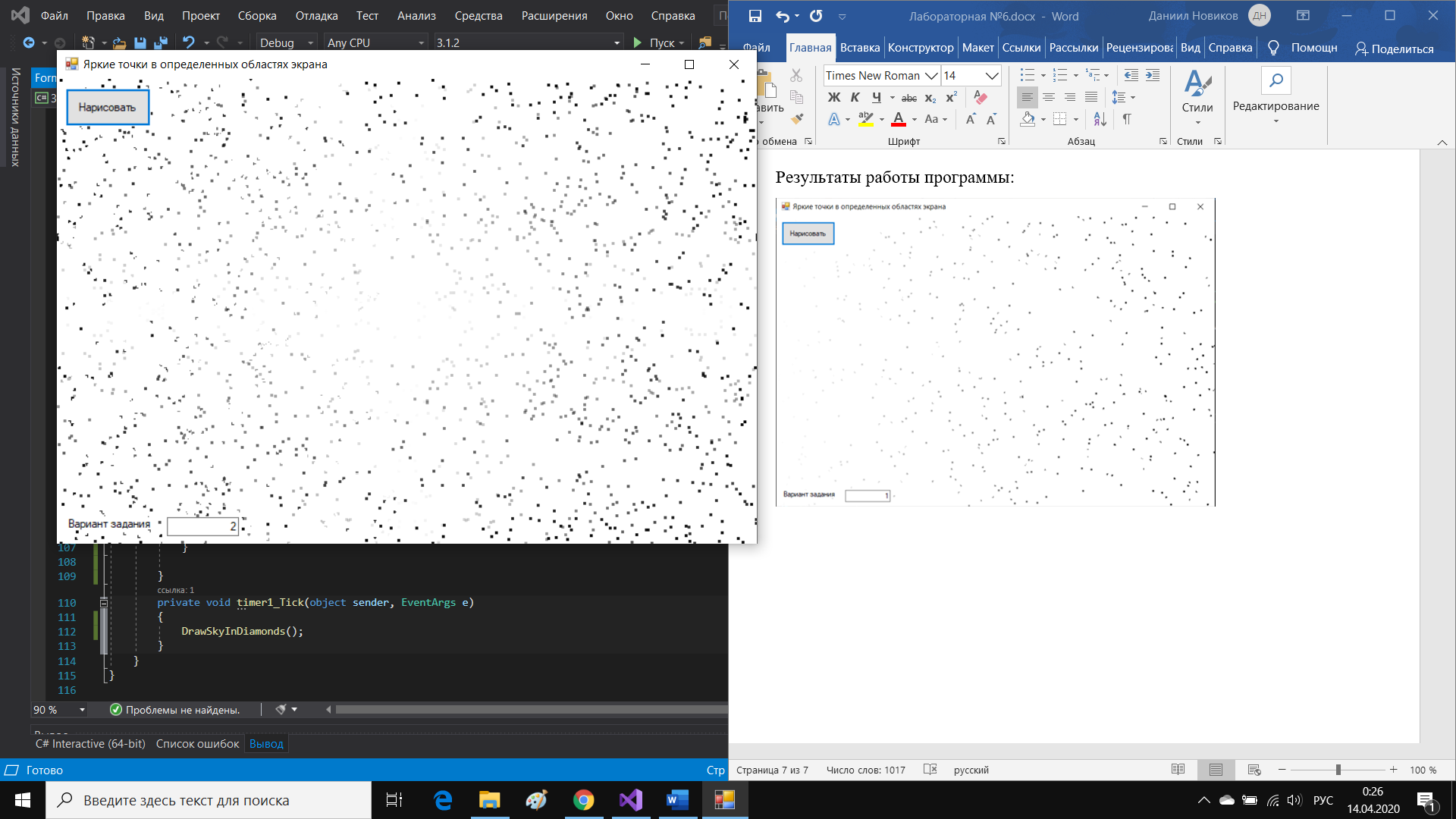
}

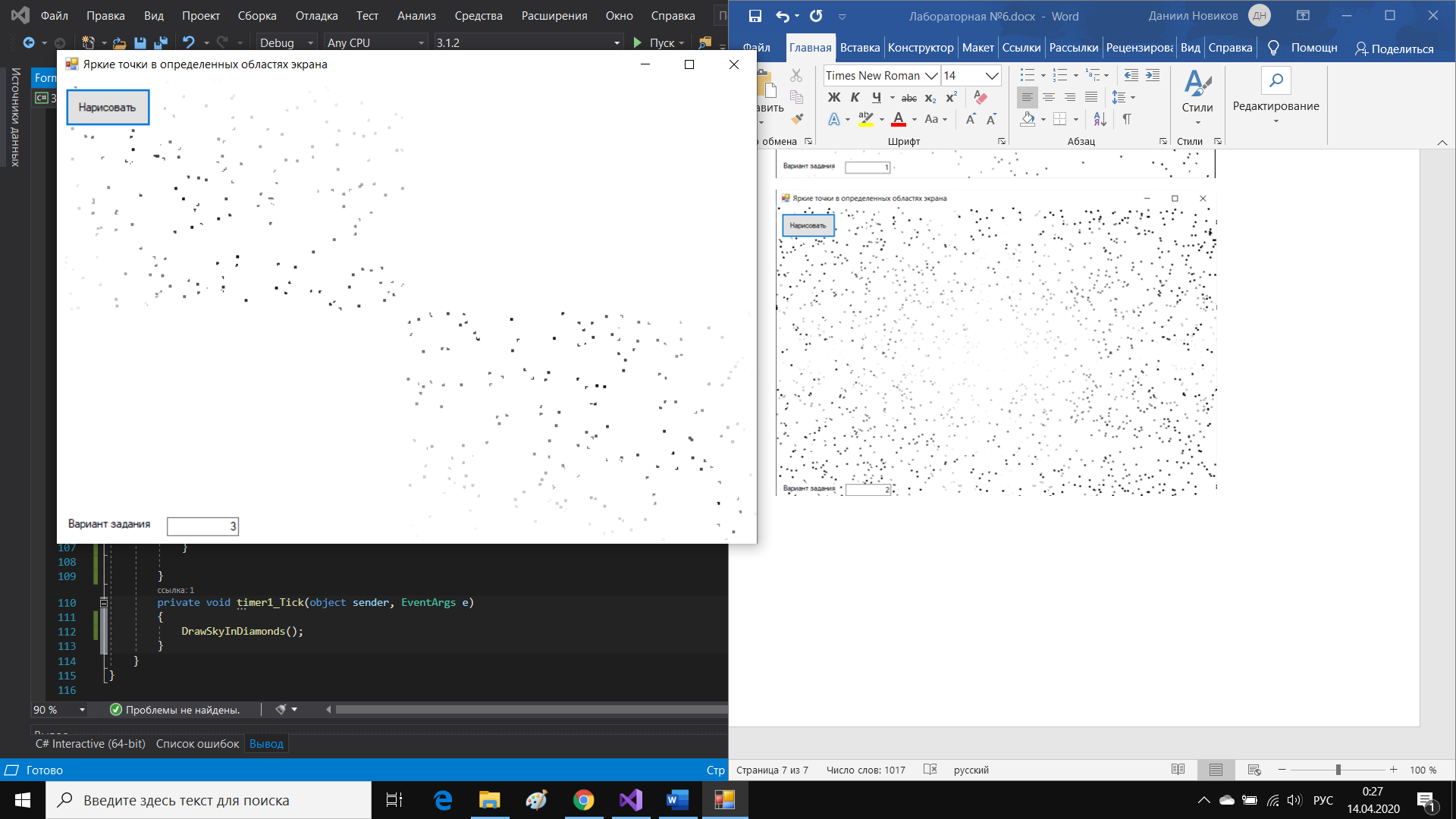
}

}

Результаты работы программы:







Задание 3.2.1

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;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_3.\_2.\_1

{

public partial class Form1 : Form

{

Graphics graph;

Pen pen;

Brush brush;

Random rand;

Point[] point;

Point[] prevPoint;

int x, y, prevx, prevy;

int xMax, yMax;

int n = 100;

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

pen = new Pen(Color.Black, 1);

rand = new Random();

point = new Point[n];

prevPoint = new Point[n];

x = 0; y = 0;

xMax = ClientSize.Width;

yMax = ClientSize.Height;

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

timer1.Start();

else if (e.KeyCode == Keys.Escape)

{

timer1.Stop();

Close();

}

}

private void DrawFlicker()

{

int i = rand.Next(7) + 9;

switch (i)

{

case 9:

brush = Brushes.LightBlue;

break;

case 10:

brush = Brushes.LightGreen;

break;

case 11:

brush = Brushes.Purple;

break;

case 12:

brush = Brushes.Red;

break;

case 13:

brush = Brushes.Magenta;

break;

case 14:

brush = Brushes.Yellow;

break;

case 15:

brush = Brushes.White;

break;

}

point = Enumerable.Range(0, n).Select(s => new Point(rand.Next(xMax), rand.Next(yMax))).ToArray();

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

graph.FillRectangle(brush, point[j].X, point[j].Y, 2, 2);

Thread.Sleep(100);

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

graph.FillRectangle(Brushes.White, point[j].X, point[j].Y, 2, 2);

}

private void timer1\_Tick(object sender, EventArgs e)

{

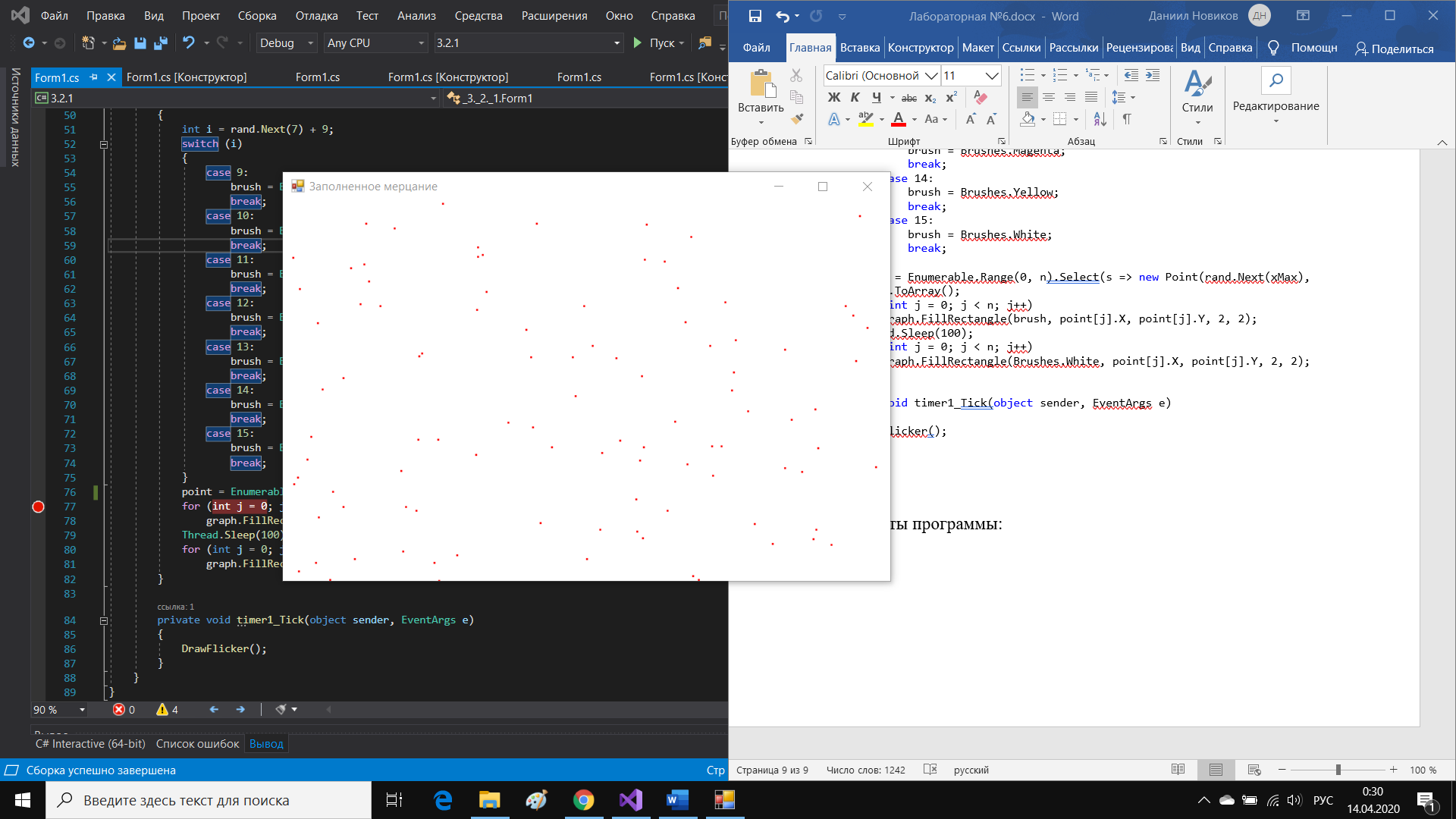
DrawFlicker();

}

}

}

Результат работы программы:



Задание 5.1

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;

namespace \_5.\_1

{

public partial class Form1 : Form

{

Graphics graph;

int xMax, yMax;

Random rand;

const int n = 500;

struct Molecule

{

public Point location;

public Point speed;

public Brush color;

}

Molecule[] molecules;

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

molecules = new Molecule[n];

xMax = ClientSize.Width;

yMax = ClientSize.Height;

rand = new Random();

}

private void timer1\_Tick(object sender, EventArgs e)

{

graph.Clear(BackColor);

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

{

if (molecules[i].color == Brushes.Red || molecules[i].color == Brushes.Magenta)

{

molecules[i].speed = new Point(-20 + rand.Next(41), -10 + rand.Next(11));

}

else

{

molecules[i].speed = new Point(-20 + rand.Next(41), -10 + rand.Next(41));

}

if (molecules[i].location.X + molecules[i].speed.X > 0 && molecules[i].location.X + molecules[i].speed.X < xMax) molecules[i].location.X += molecules[i].speed.X;

if (molecules[i].location.Y + molecules[i].speed.Y > 10 && molecules[i].location.Y + molecules[i].speed.Y < yMax) molecules[i].location.Y += molecules[i].speed.Y;

graph.FillRectangle(molecules[i].color, molecules[i].location.X, molecules[i].location.Y, 2, 2);

}

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

timer1.Start();

}

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

{

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

{

molecules[i].location = new Point(rand.Next(xMax), rand.Next(yMax));

molecules[i].speed = new Point(-20 + rand.Next(41), -10 + rand.Next(41));

int j = rand.Next(6) + 9;

switch (j)

{

case 9:

molecules[i].color = Brushes.LightBlue;

break;

case 10:

molecules[i].color = Brushes.LightGreen;

break;

case 11:

molecules[i].color = Brushes.LightCyan;

break;

case 12:

molecules[i].color = Brushes.Red;

break;

case 13:

molecules[i].color = Brushes.Magenta;

break;

case 14:

molecules[i].color = Brushes.Yellow;

break;

}

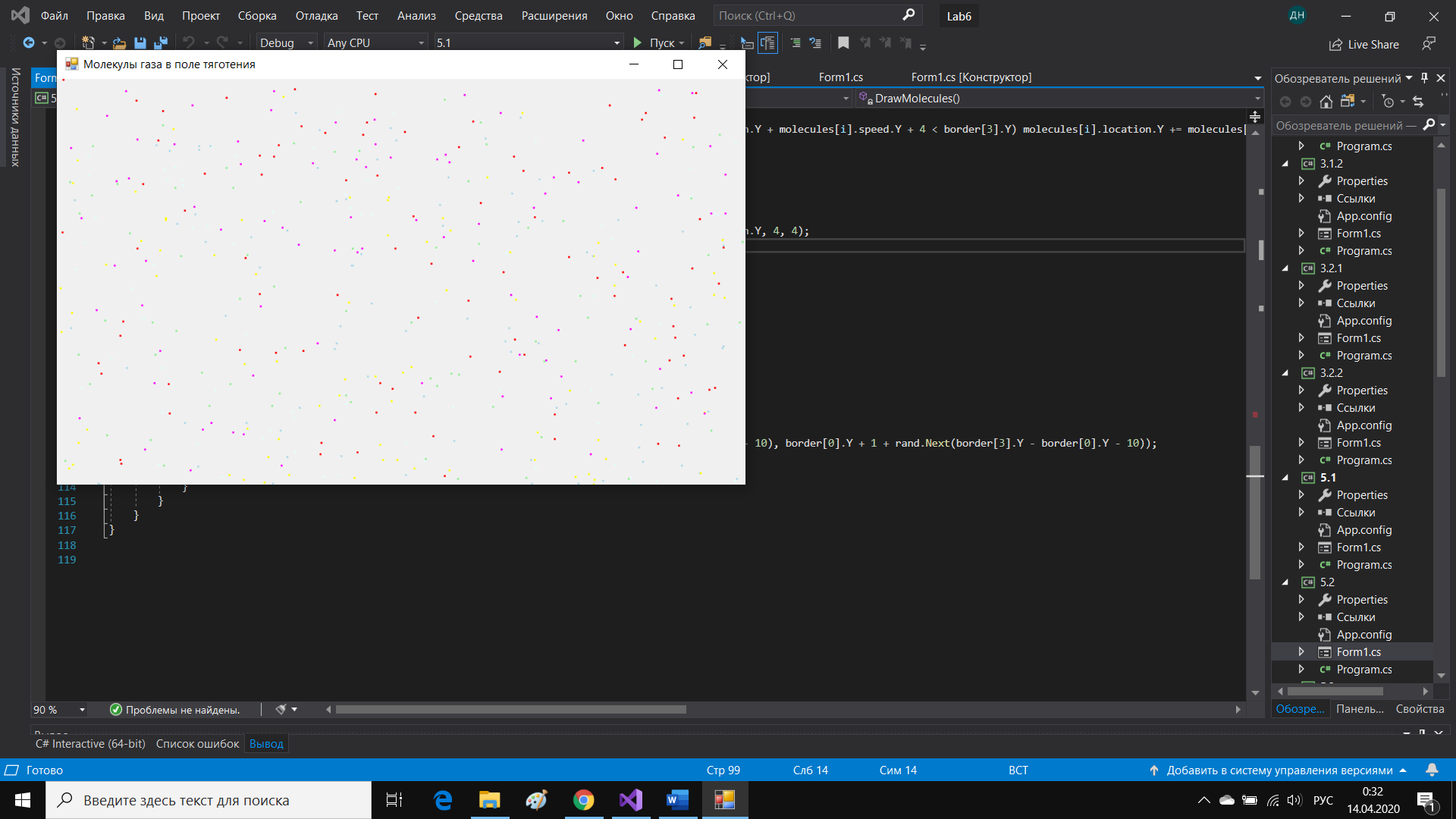
}

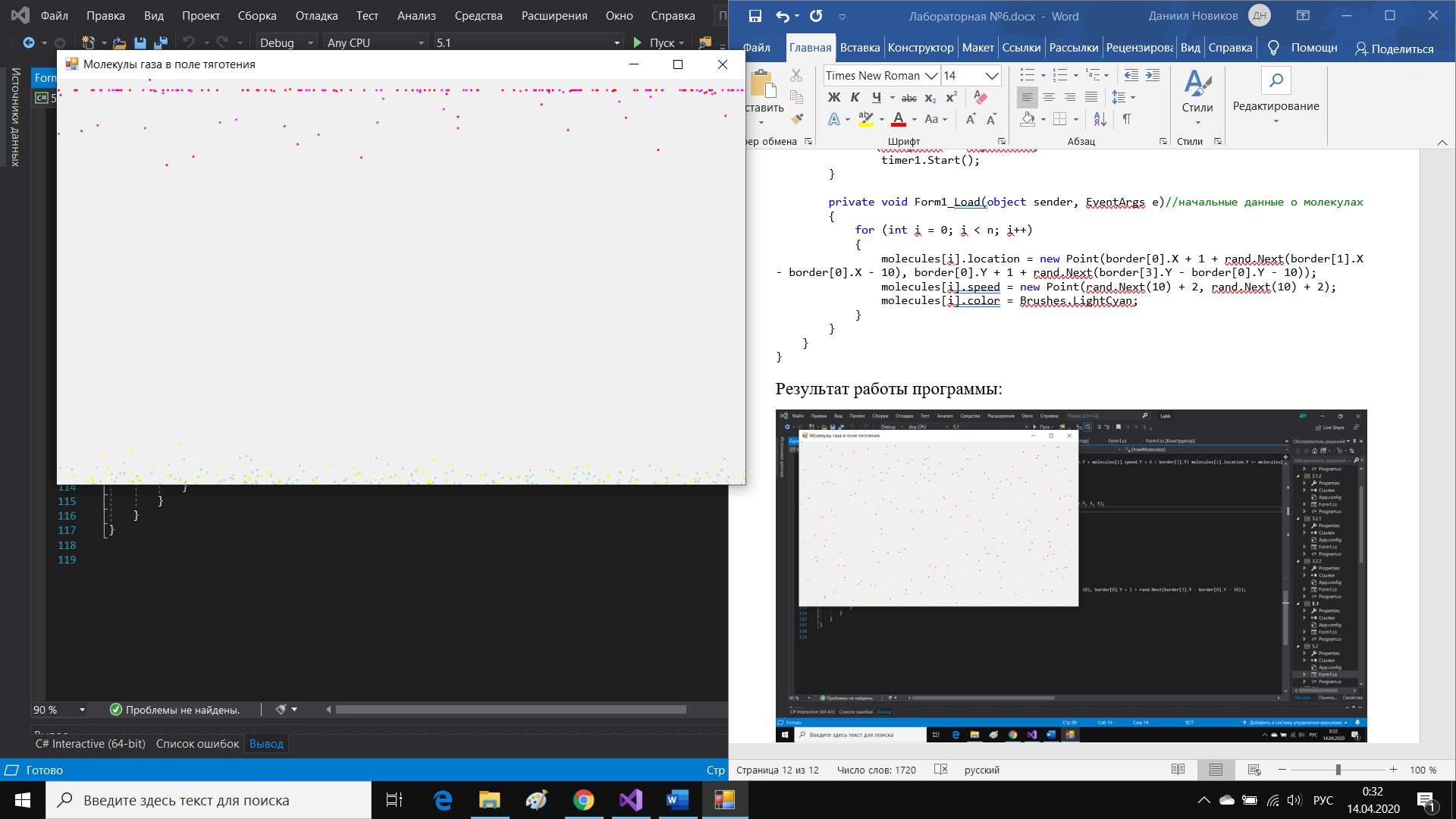
}

}

}

Результат работы программы:





Задание 5.2

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;

namespace \_5.\_2

{

public partial class Form1 : Form

{

Bitmap bmp;

struct Cylinder

{

public Point[] firstPart;

public Point[] secondPart;

public Point[] thirdPart;

}

Cylinder cylinder;

const int n = 200;

struct Molecule

{

public Point location;

public Point speed;

public Brush color;

}

Molecule[] molecules;

Graphics graph;

int dx = -2;//смещение элментов цилиндра

Point[] border;//в этом полигоне молекулы могут двигаться

Random rand;

public Form1()

{

InitializeComponent();

rand = new Random();

cylinder = new Cylinder();

molecules = new Molecule[n];

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

graph = Graphics.FromImage(bmp);

cylinder.firstPart = new Point[4] { new Point(30, 80), new Point(ClientSize.Width / 2 + 90, 80), new Point(ClientSize.Width / 2 + 90, ClientSize.Height - 100), new Point(30, ClientSize.Height - 100) };

cylinder.secondPart = new Point[4] { new Point(cylinder.firstPart[1].X - 30, cylinder.firstPart[1].Y), cylinder.firstPart[1], cylinder.firstPart[2], new Point(cylinder.firstPart[1].X - 30, cylinder.firstPart[2].Y) };

cylinder.thirdPart = new Point[4] { new Point(cylinder.secondPart[1].X, (cylinder.secondPart[1].Y + cylinder.secondPart[2].Y) / 2 - 10), new Point(ClientSize.Width - 10, (cylinder.secondPart[1].Y + cylinder.secondPart[2].Y) / 2 - 10), new Point(ClientSize.Width - 10, (cylinder.secondPart[1].Y + cylinder.secondPart[2].Y) / 2 + 10), new Point(cylinder.secondPart[1].X, (cylinder.secondPart[1].Y + cylinder.secondPart[2].Y) / 2 + 10) };

border = new Point[4] { new Point(30, 80), new Point(cylinder.firstPart[1].X - 30, cylinder.firstPart[1].Y), new Point(cylinder.firstPart[1].X - 30, cylinder.firstPart[2].Y), new Point(30, ClientSize.Height - 100) };

}

private void timer1\_Tick(object sender, EventArgs e)

{

DrawCylinder();

DrawMolecules();

ChangeLocationCylinder();

pictureBox1.Image = bmp;

}

private void ChangeLocationCylinder()

{

if (cylinder.secondPart[0].X + dx < cylinder.firstPart[0].X + 150)

dx = -dx;

else if (cylinder.secondPart[1].X + dx > cylinder.firstPart[1].X)

dx = -dx;

for (int i = 0; i < 4; i++)

cylinder.secondPart[i].X += dx;

for (int i = 0; i < 4; i++)

cylinder.thirdPart[i].X += dx;

for (int i = 1; i < 3; i++)//меняем границы движения молекул

border[i].X += dx;

}

private void DrawCylinder()

{

graph.Clear(BackColor);

graph.DrawPolygon(new Pen(Color.Black, 2), cylinder.firstPart);

graph.DrawPolygon(new Pen(Color.Black, 2), cylinder.secondPart);

graph.DrawPolygon(new Pen(Color.Black, 2), cylinder.thirdPart);

graph.FillPolygon(Brushes.Gray, cylinder.secondPart);

graph.FillPolygon(Brushes.Gray, cylinder.thirdPart);

}

private void DrawMolecules()

{

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

{

if (molecules[i].location.X + molecules[i].speed.X > border[0].X && molecules[i].location.X + molecules[i].speed.X + 4 < border[1].X) molecules[i].location.X += molecules[i].speed.X;

else

{

molecules[i].speed.X = -molecules[i].speed.X;

molecules[i].location.X += molecules[i].speed.X;

}

if (molecules[i].location.Y + molecules[i].speed.Y > border[0].Y && molecules[i].location.Y + molecules[i].speed.Y + 4 < border[3].Y) molecules[i].location.Y += molecules[i].speed.Y;

else

{

molecules[i].speed.Y = -molecules[i].speed.Y;

molecules[i].location.Y += molecules[i].speed.Y;

}

graph.DrawEllipse(new Pen(Color.Black, 1), molecules[i].location.X, molecules[i].location.Y, 4, 4);

}

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

timer1.Start();

}

private void Form1\_Load(object sender, EventArgs e)//начальные данные о молекулах

{

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

{

molecules[i].location = new Point(border[0].X + 1 + rand.Next(border[1].X - border[0].X - 10), border[0].Y + 1 + rand.Next(border[3].Y - border[0].Y - 10));

molecules[i].speed = new Point(rand.Next(10) + 2, rand.Next(10) + 2);

molecules[i].color = Brushes.LightCyan;

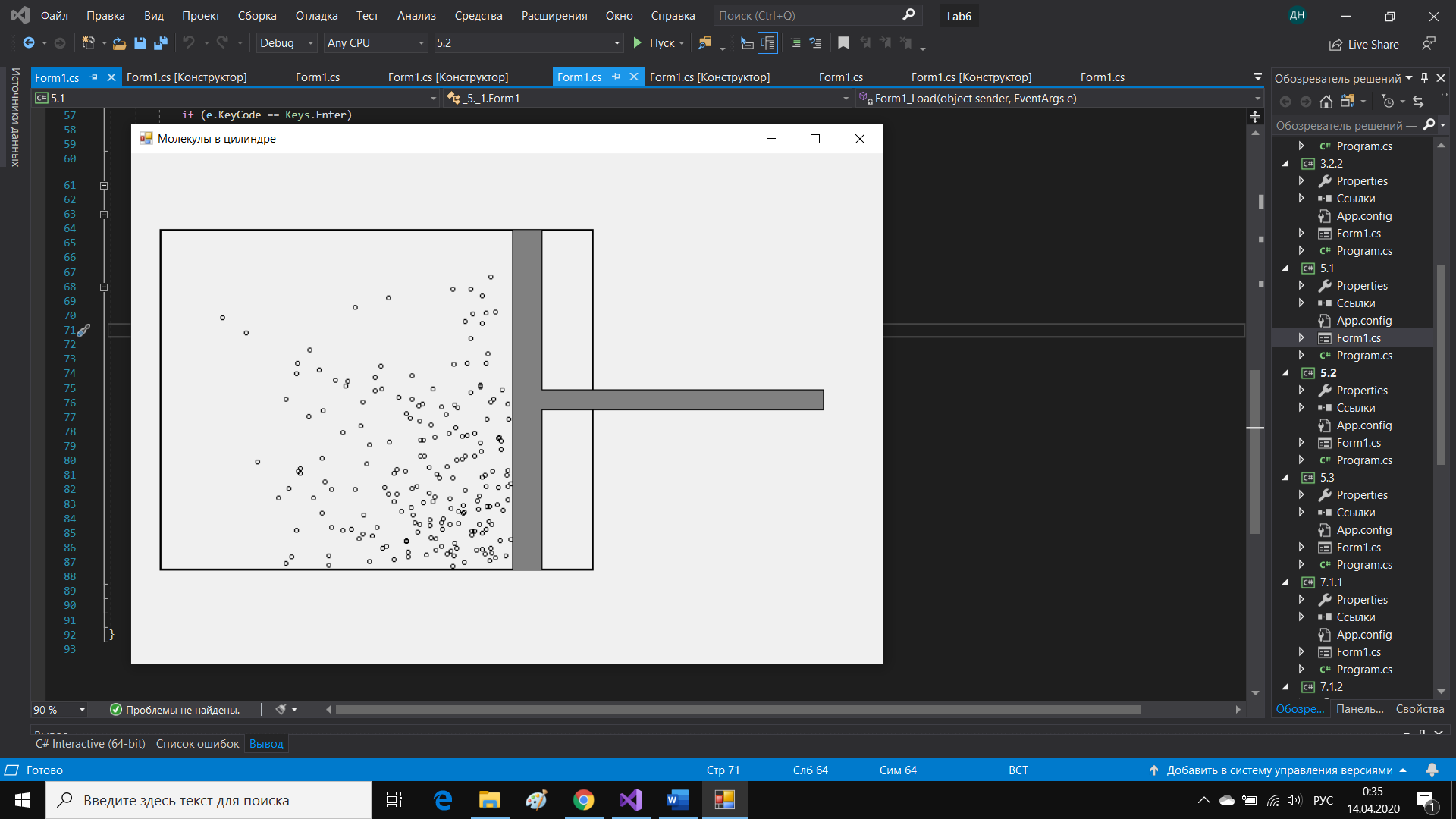
}

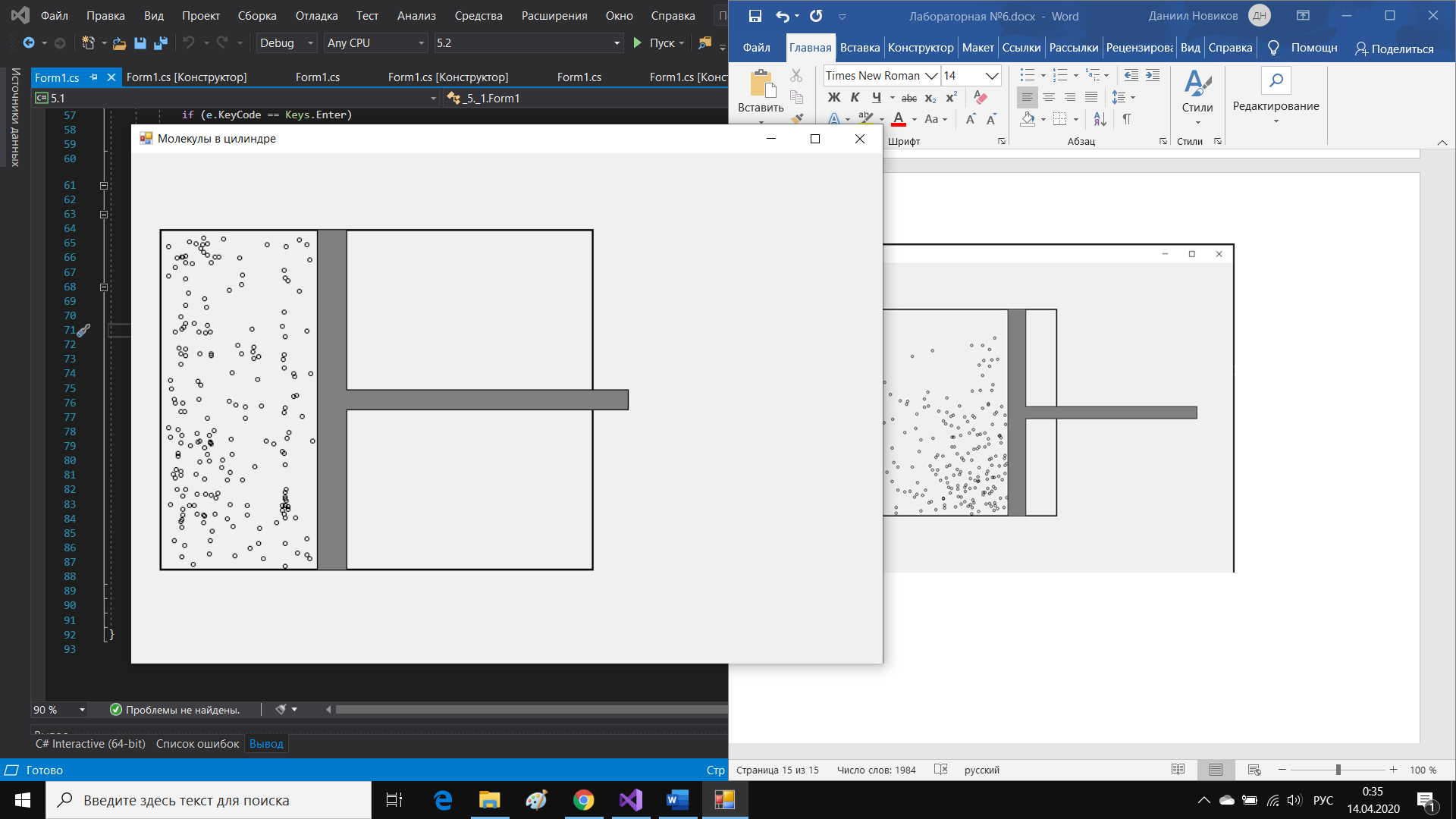
}

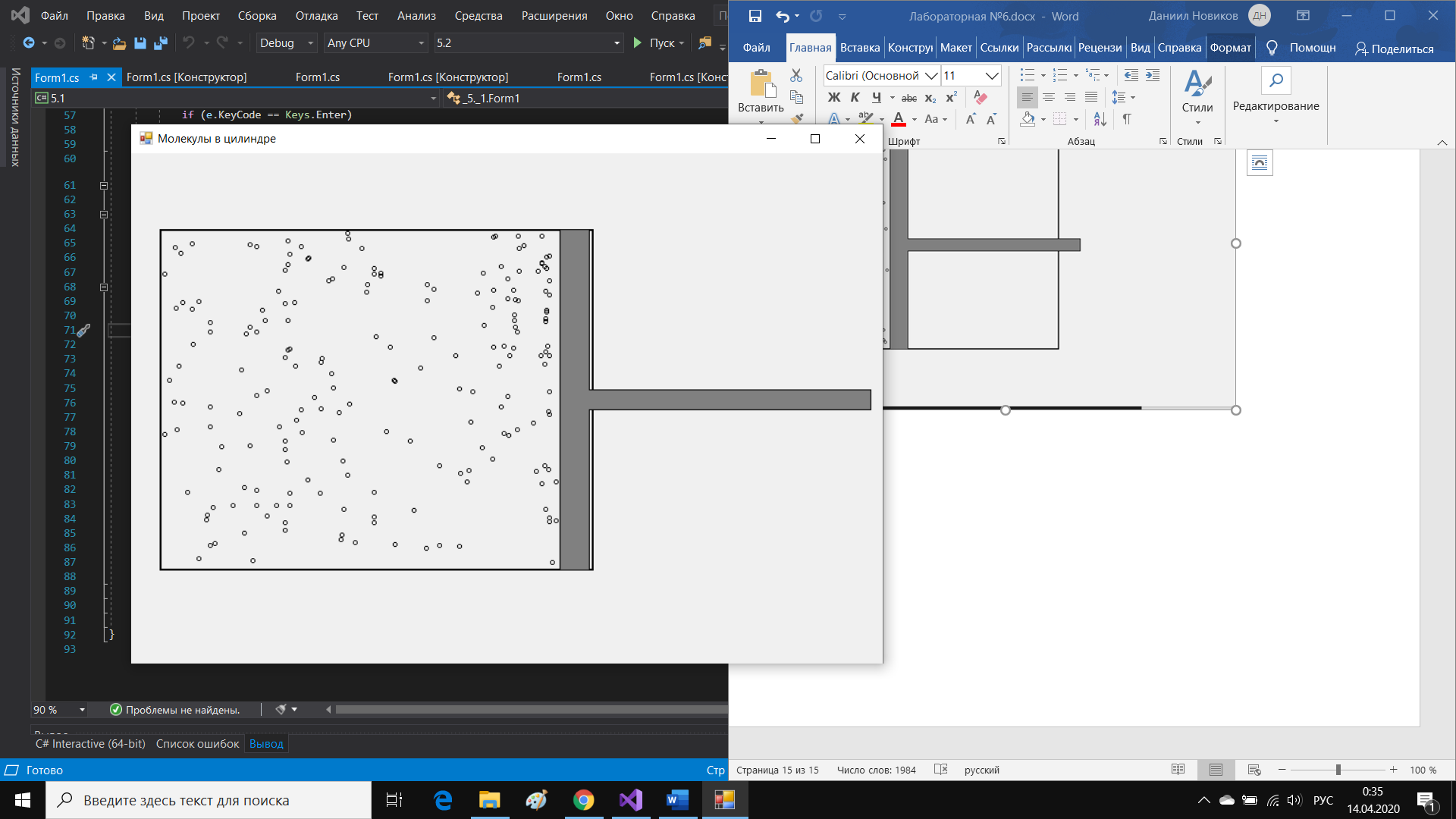
}

}

Результат работы программы:







Задание 5.3

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;

namespace \_5.\_3

{

public partial class Form1 : Form

{

Graphics graph;

int xMax, yMax;

Random rand;

int seconds;

Bitmap bmp;

struct Aqua//структура для жидкости

{

public struct Molecule

{

public Point location;

public Point speed;

}

public Molecule[] molecules;

public int n;

public Brush color;

}

Aqua first;

Aqua second;

Aqua third;

public Form1()

{

InitializeComponent();

seconds = 0;

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

graph = Graphics.FromImage(bmp);

first = new Aqua();

second = new Aqua();

third = new Aqua();

xMax = ClientSize.Width;

yMax = ClientSize.Height;

rand = new Random();

}

private void timer1\_Tick(object sender, EventArgs e)

{

++seconds;

graph.Clear(BackColor);

if (seconds > 10)

DrawAqua(first);

else

{

Initialization(ref first, 7 \* yMax / 10, 3);

FirstDraw(first);

}

if (seconds >= 15)

DrawAqua(second);

else

{

Initialization(ref second, 8 \* yMax / 10, 6);

FirstDraw(second);

}

if (seconds > 30)

DrawAqua(third);

else

{

Initialization(ref third, 9 \* yMax / 10 ,6);

FirstDraw(third);

}

pictureBox1.Image = bmp;

}

private void DrawAqua(Aqua tmp)

{

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

{

if (tmp.molecules[i].location.X + tmp.molecules[i].speed.X > 0 && tmp.molecules[i].location.X + tmp.molecules[i].speed.X < xMax) tmp.molecules[i].location.X += tmp.molecules[i].speed.X;

if (tmp.molecules[i].location.Y + tmp.molecules[i].speed.Y > 10 && tmp.molecules[i].location.Y + tmp.molecules[i].speed.Y < yMax) tmp.molecules[i].location.Y += tmp.molecules[i].speed.Y;

graph.FillRectangle(tmp.color, tmp.molecules[i].location.X, tmp.molecules[i].location.Y, 2, 2);

}

}

private void FirstDraw(Aqua tmp)

{

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

{

graph.FillRectangle(tmp.color, tmp.molecules[i].location.X, tmp.molecules[i].location.Y, 2, 2);

}

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

timer1.Start();

}

private void Initialization(ref Aqua tmp, int y, int speed)

{

tmp.n = rand.Next(100, 200);

tmp.molecules = new Aqua.Molecule[tmp.n];

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

{

tmp.molecules[i].location = new Point(rand.Next(xMax), y-10+rand.Next(10));

tmp.molecules[i].speed = new Point(-20 + rand.Next(41), -(rand.Next(12)+1));

}

}

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

{

first.color = Brushes.Blue;

second.color = Brushes.Orange;

third.color = Brushes.Red;

Initialization(ref first, 7 \* yMax / 10, 6);

Initialization(ref second, 8 \* yMax / 10, 4);

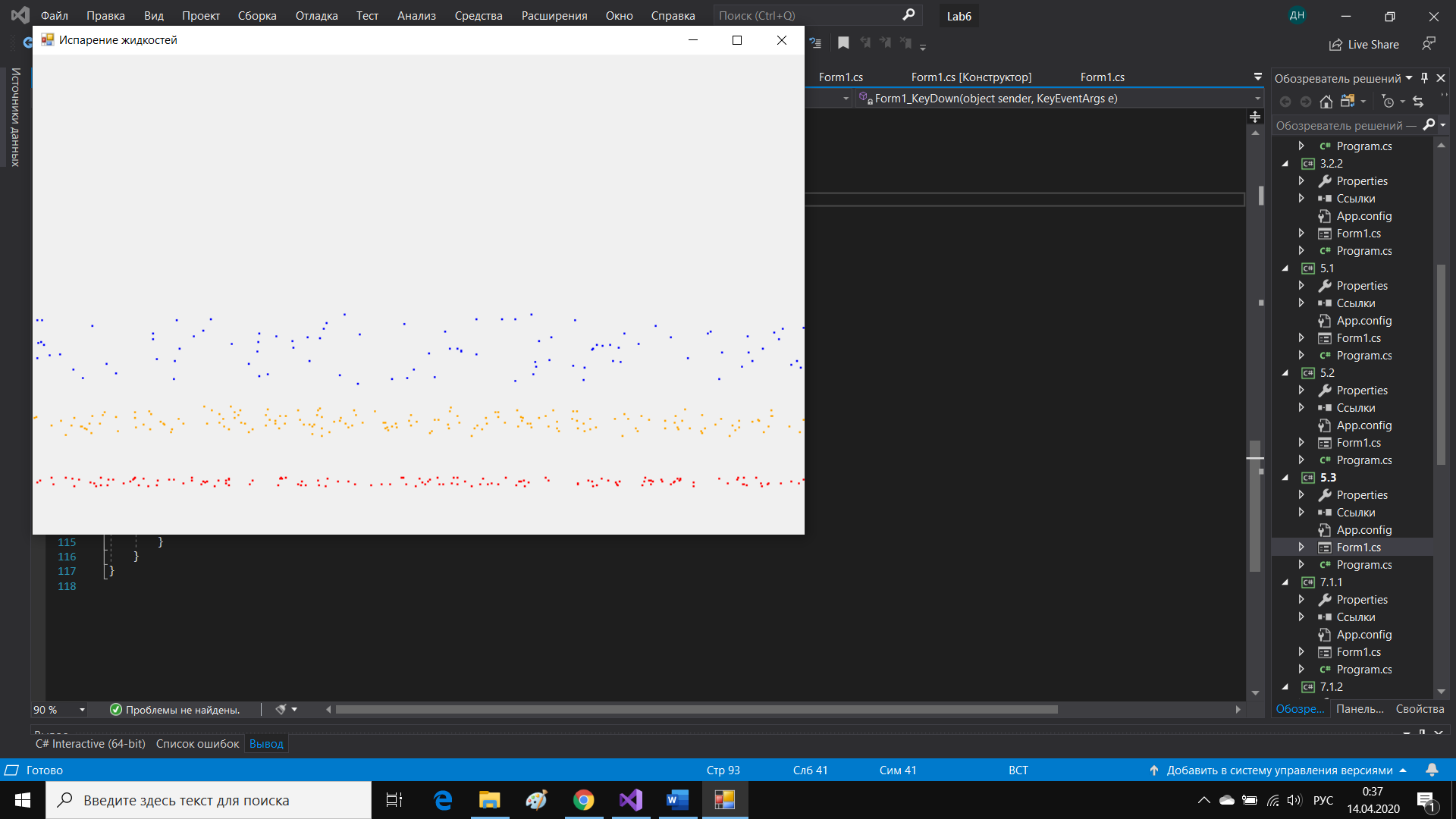
Initialization(ref third, 9 \* yMax / 10, 3);

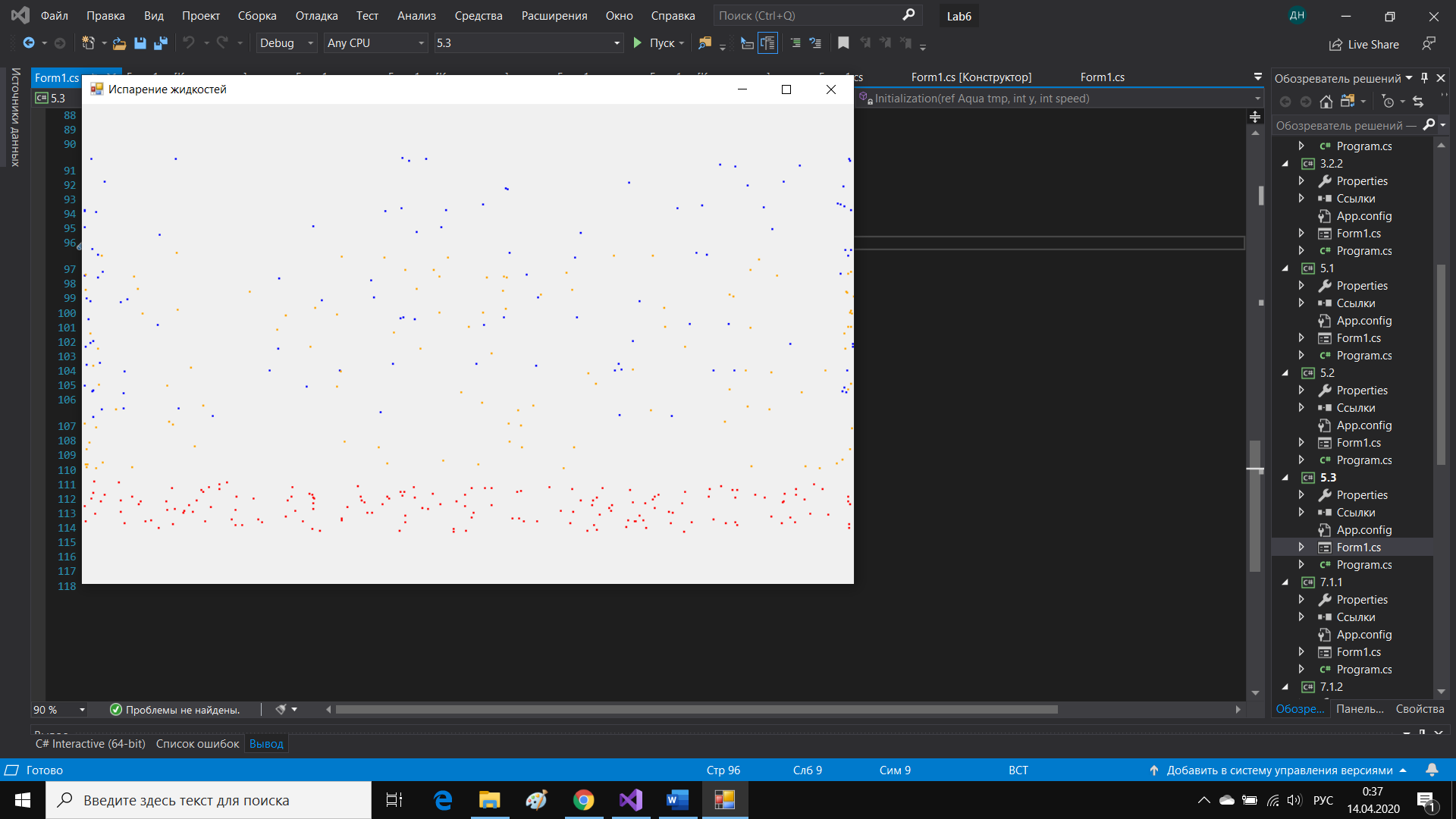
}

}

}

Результат работы программы:





Задание 7.1.1

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;

namespace \_7.\_1.\_1

{

public partial class Form1 : Form

{

Bitmap bmp;

Graphics graph;

Point[] snows;

int n = 300;

Random rand;

Point[] border;//преграда

int xMax, yMax;

int wind;//ветер

int r;

int seconds = 0;

public Form1()

{

InitializeComponent();

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

graph = Graphics.FromImage(bmp);

snows = new Point[n];

rand = new Random();

wind = 1;

r = 2;//размер снежинки

}

private void DrawSnows()

{

graph.FillPolygon(Brushes.Black, border);

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

{

if (snows[i].Y+r + Math.Abs(wind) < yMax&&snows[i].X+Math.Abs(wind)+r<xMax &&snows[i].X+wind>0&& bmp.GetPixel(snows[i].X+r+1,snows[i].Y+r+1) != Color.FromArgb(0,0,0)&& bmp.GetPixel(snows[i].X+r+1, snows[i].Y + r+1) != Color.FromArgb(255,255,255))

{

snows[i].Y++;

snows[i].X += wind;

graph.FillRectangle(Brushes.White, snows[i].X, snows[i].Y, r ,r);

}

else

{

snows[i] = new Point(rand.Next(xMax), rand.Next(3\*yMax/4));

graph.FillRectangle(Brushes.White, snows[i].X, snows[i].Y, r, r);

}

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (seconds % 57== 0)

wind = 1;

else if (seconds % 63 == 0)

wind = -1;

else if (seconds %121==0)

wind = 3;

++seconds;

graph.Clear(BackColor);

DrawSnows();

pictureBox1.Image = bmp;

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

timer1.Start();

}

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

{

xMax = pictureBox1.Width;

yMax =pictureBox1.Height;

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

{

snows[i] = new Point(rand.Next(xMax), rand.Next(yMax));

}

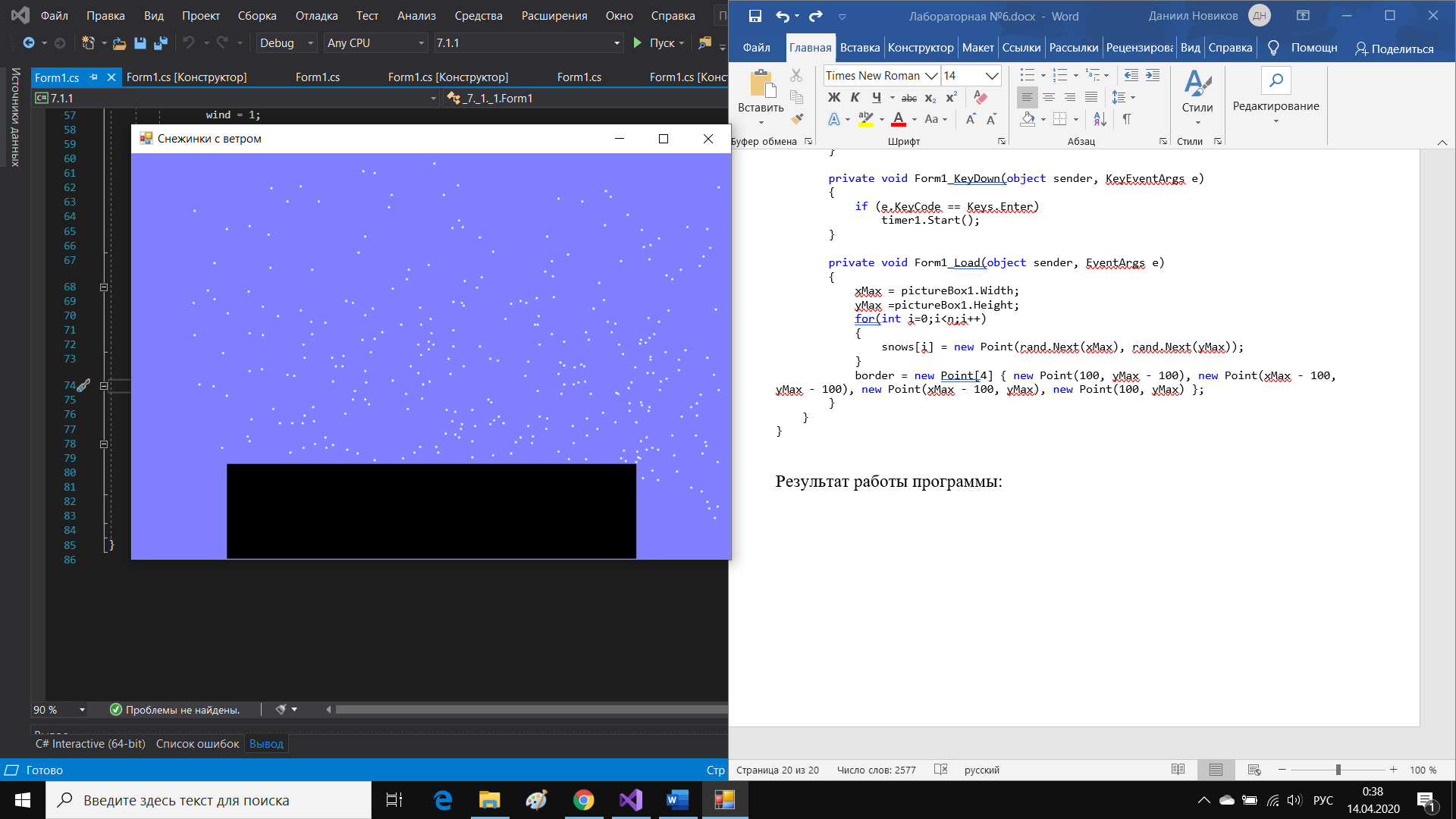
border = new Point[4] { new Point(100, yMax - 100), new Point(xMax - 100, yMax - 100), new Point(xMax - 100, yMax), new Point(100, yMax) };

}

}

}

Результат работы программы:



Задание 7.1.2

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;

namespace \_7.\_1.\_2

{

public partial class Form1 : Form

{

Bitmap bmp;

Graphics graph;

Point[] snows;

int n = 300;

Random rand;

Point[] border;//преграда

int xMax, yMax;

int r;

public Form1()

{

InitializeComponent();

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

graph = Graphics.FromImage(bmp);

snows = new Point[n];

rand = new Random();

r = 2;//размер снежинки

}

private bool CheckBorderOrSnow(int i)//если на пути препятствие или снежника, то вернуть true

{

for (int j = 0; j < r; j++)//проверяем по каждому пикселю по x

if (bmp.GetPixel(snows[i].X + j, snows[i].Y + r) == Color.FromArgb(0, 0, 0) || bmp.GetPixel(snows[i].X + j, snows[i].Y + r) == Color.FromArgb(255, 255, 255)) return true;

return false;

}

private void DrawSnows()

{

graph.FillPolygon(Brushes.Black, border);

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

{

if (snows[i].Y + r < yMax &&snows[i].X+r<xMax&& !CheckBorderOrSnow(i))

{

graph.FillRectangle(Brushes.Blue, snows[i].X, snows[i].Y, r, r);

snows[i].Y++;

graph.FillRectangle(Brushes.White, snows[i].X, snows[i].Y, r, r);

}

else

{

snows[i] = new Point(rand.Next(xMax), rand.Next(3 \* yMax / 4));

graph.FillRectangle(Brushes.White, snows[i].X, snows[i].Y, r, r);

}

}

}

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

{

xMax = pictureBox1.Width;

yMax = pictureBox1.Height;

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

{

snows[i] = new Point(rand.Next(xMax), rand.Next(yMax));

}

border = new Point[4] { new Point(100, yMax - 100), new Point(xMax - 100, yMax - 100), new Point(xMax - 100, yMax), new Point(100, yMax) };

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

timer1.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

DrawSnows();

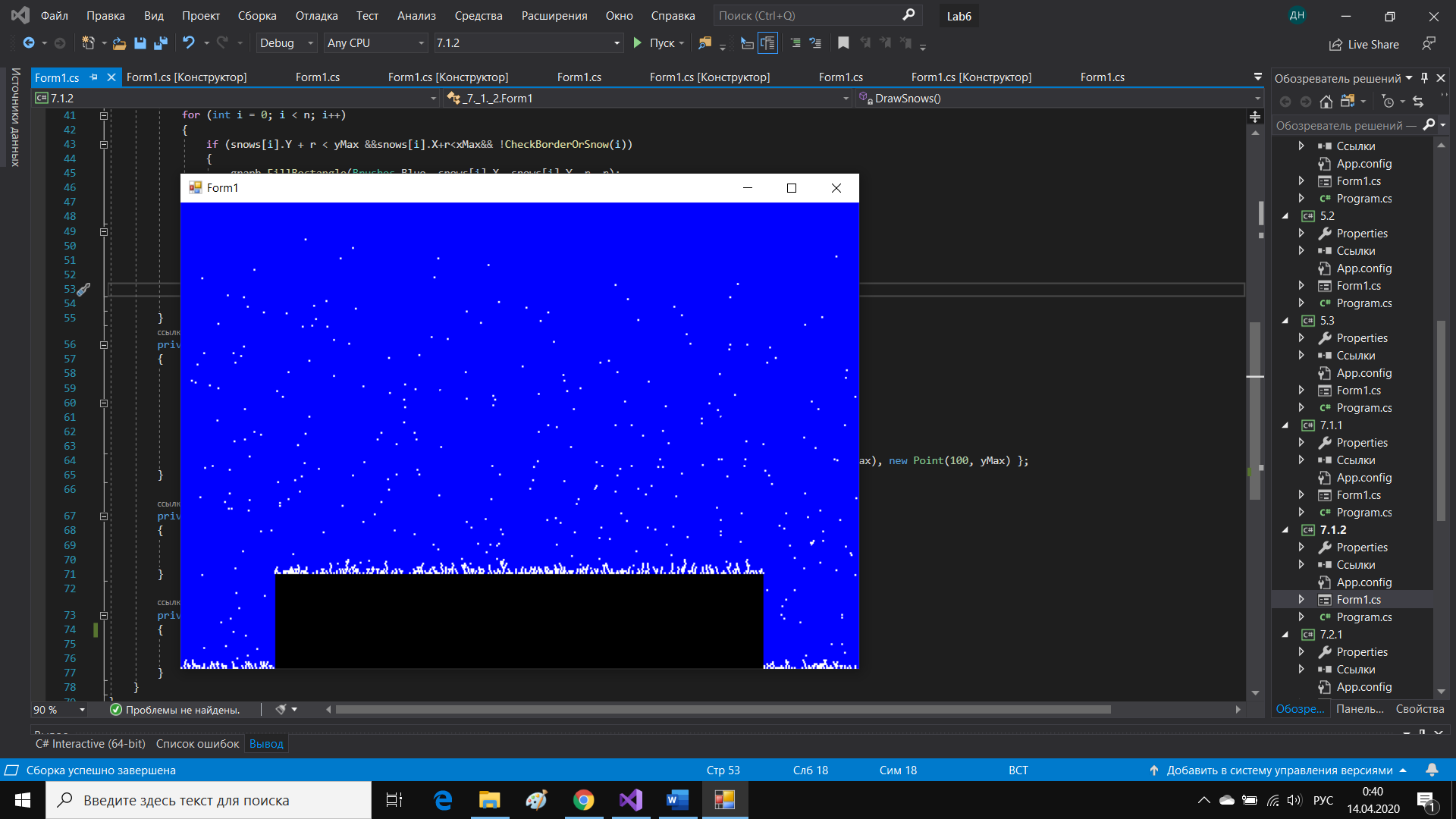
pictureBox1.Image = bmp;

}

}

}

Результат работы программы:



Задание 7.2.1

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;

namespace \_7.\_2.\_1

{

public partial class Form1 : Form

{

struct Shard//свойства осколка

{

public Point location;

public Point speed;

public int size;

}

int n = 100;

Shard[] shards;

Graphics graph;

Bitmap bmp;

Random rand;

int xMax, yMax;

public Form1()

{

InitializeComponent();

shards = new Shard[n];

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

graph = Graphics.FromImage(bmp);

rand = new Random();

}

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

{

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

{

shards[i].location = new Point(300 + i % 25, 200 + i / 25);

shards[i].speed = new Point(-20 + rand.Next(41), -20 + rand.Next(41));

shards[i].size = rand.Next(4) + 1;

}

xMax = bmp.Width;

yMax = bmp.Height;

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

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

graph.FillRectangle(Brushes.Black, shards[i].location.X, shards[i].location.Y, 1, 1);

pictureBox1.Image = bmp;

if (timer1.Enabled) timer1.Stop();

timer1.Start();

}

}

private void DrawShards()

{

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

{

if (shards[i].location.X + shards[i].speed.X > 0 && shards[i].location.X + shards[i].speed.X < xMax &&

shards[i].location.Y + shards[i].speed.Y > 0 && shards[i].location.Y + shards[i].speed.Y < yMax)

{

shards[i].location.X+= shards[i].speed.X;

shards[i].location.Y+= shards[i].speed.Y;

graph.FillEllipse(Brushes.Black, shards[i].location.X, shards[i].location.Y, shards[i].size, shards[i].size);

if (shards[i].speed.X != 0) shards[i].speed.X = (shards[i].speed.X >= 0) ? shards[i].speed.X -= 1 : -1 \* (Math.Abs(shards[i].speed.X) - 1);

if (shards[i].speed.Y != 0) shards[i].speed.Y = (shards[i].speed.Y >= 0) ? shards[i].speed.Y-=1 : -1 \* (Math.Abs(shards[i].speed.Y) - 1);

}

else

{

shards[i].speed.X = 0;

shards[i].speed.Y = 0;

}

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

graph.Clear(BackColor);

DrawShards();

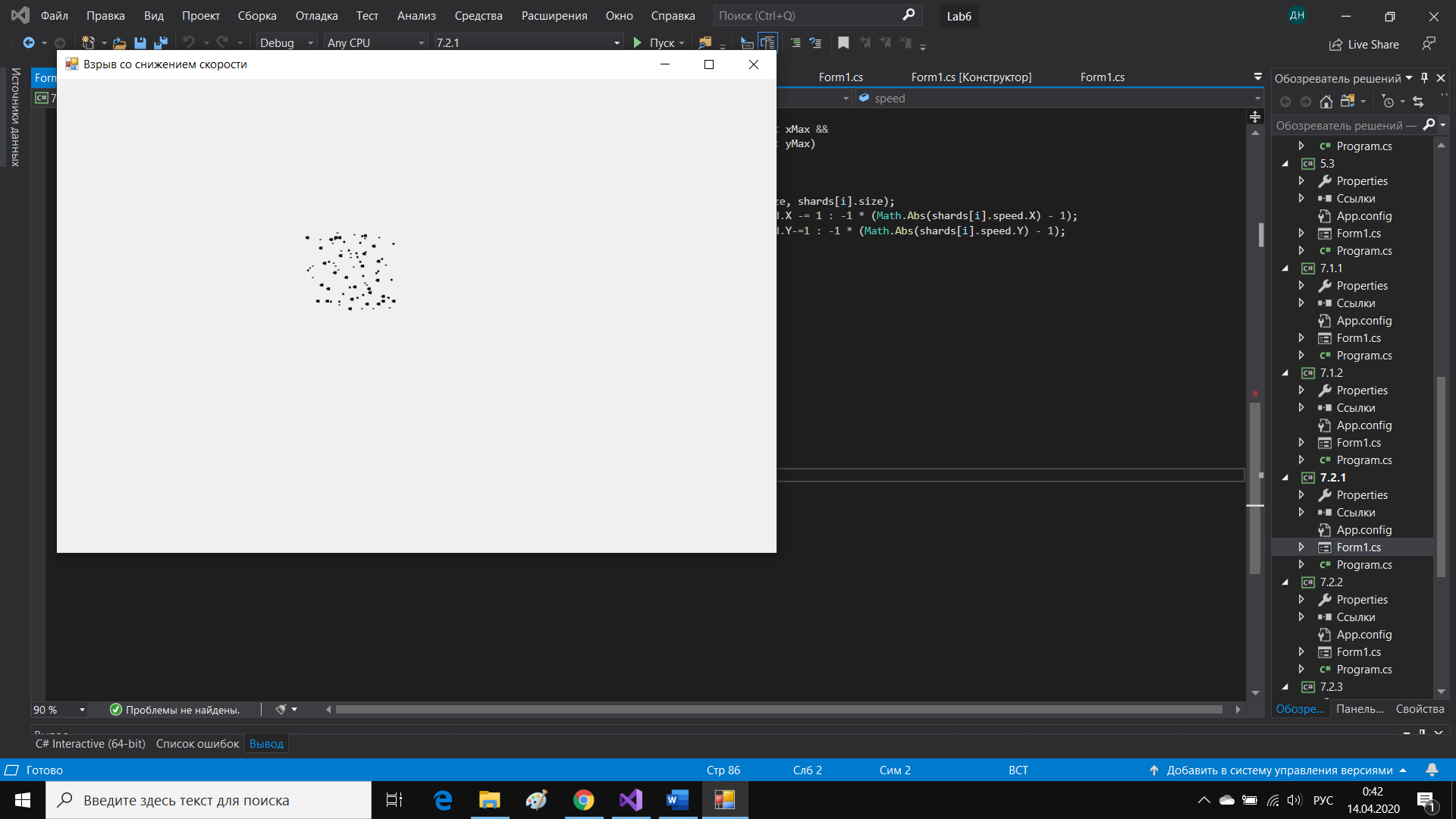
pictureBox1.Image = bmp;

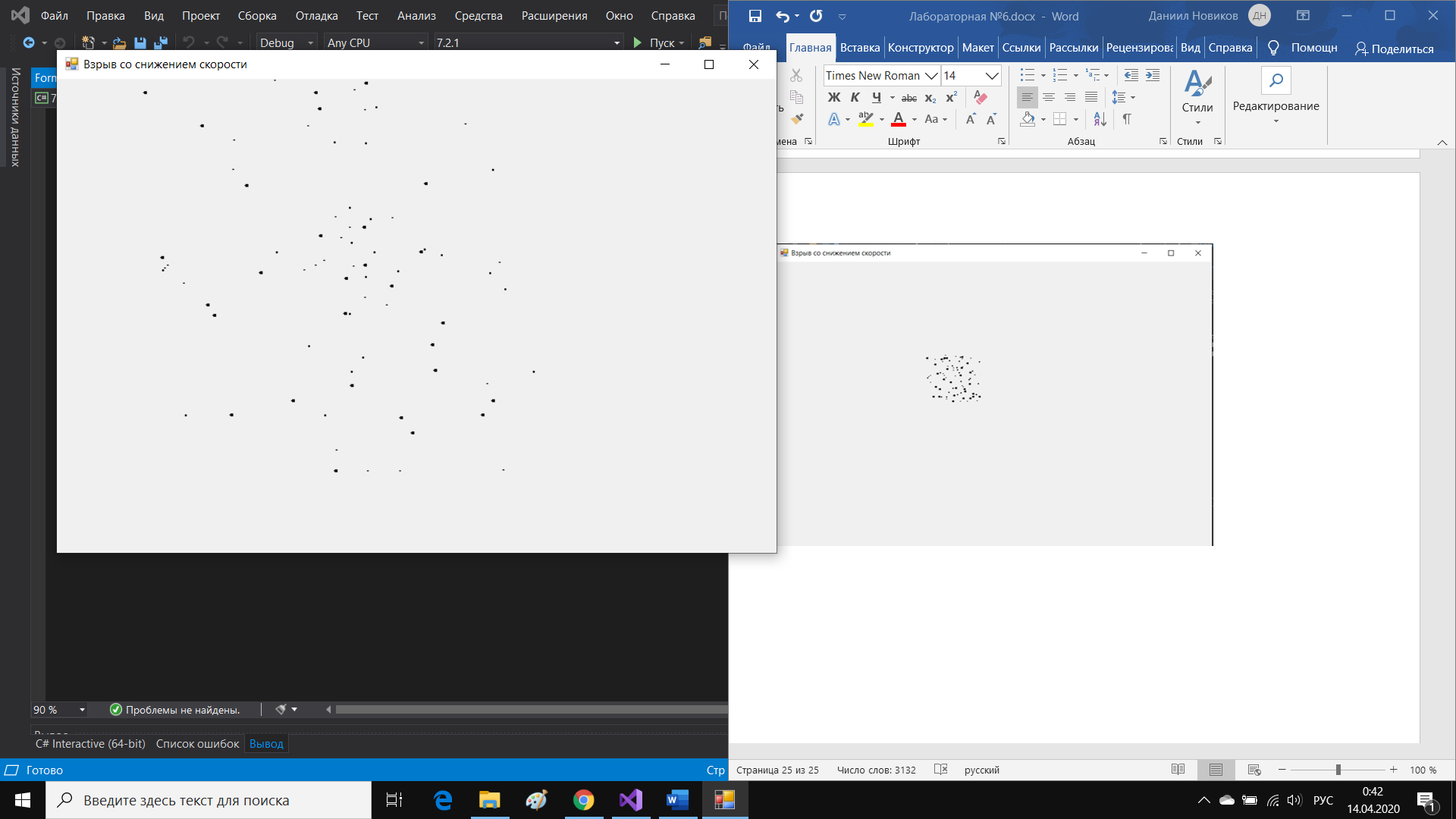
}

}

}

Результат работы программы:





Задание 7.2.2

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;

namespace \_7.\_2.\_2

{

public partial class Form1 : Form

{

struct Shard//свойства осколка

{

public Point location;

public Point speed;

public Brush color;

public int size;

}

struct Rocket//структура ракеты

{

public Point[] middlePart;

public Point[] topPart;

public Point[] downPart;

public int speed;

}

int n = 400;

Pen pen;

Point[] let;//препятствие

Shard[] shards;

Rocket rocket;

Graphics graph;

Bitmap bmp;

Random rand;

int xMax, yMax;

bool detonation;//чек на то, что начинается взрыв

bool detonationIsHere;//чек на то, что взрыв уже идет

public Form1()

{

InitializeComponent();

detonation = false;

detonationIsHere = false;

pen = new Pen(Color.Black, 2);

shards = new Shard[n];

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

graph = Graphics.FromImage(bmp);

rand = new Random();

}

private void timer1\_Tick(object sender, EventArgs e)

{

graph.Clear(BackColor);

detonation = CheckTouch();

graph.FillPolygon(Brushes.Black, let);

if (!detonation)

{

ChangeLocationRocket();

DrawRocket();

}

else

{

if(!detonationIsHere) StartLocationShards();//начальные позиции осколков

detonationIsHere = true;

DrawShards();

}

pictureBox1.Image = bmp;

}

private bool CheckTouch()//проверка на касание ракетой препятствия

{

if (rocket.topPart[1].Y < let[2].Y)

return true;

else return false;

}

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

{

xMax = bmp.Width;

yMax = bmp.Height;

let = new Point[4] { new Point(100, 100), new Point(xMax - 100, 100), new Point(xMax - 100, yMax / 4), new Point(100, yMax / 4) };

rocket.downPart = new Point[4] { new Point(xMax / 2 - 30, yMax - 30), new Point(xMax / 2 + 30, yMax - 30), new Point(xMax / 2 + 50, yMax), new Point(xMax / 2 - 50, yMax) };

rocket.middlePart = new Point[4] { new Point(rocket.downPart[0].X, rocket.downPart[0].Y - 40), new Point(rocket.downPart[1].X, rocket.downPart[1].Y - 40), rocket.downPart[1], rocket.downPart[0] };

rocket.topPart = new Point[3] { rocket.middlePart[0],new Point((rocket.middlePart[0].X+ rocket.middlePart[1].X)/2, rocket.middlePart[1].Y-20), rocket.middlePart[1] };

rocket.speed = 4;

}

private void StartLocationShards()//начальные положения точек при взрыве

{

for (int i = 0; i < n; i++)//осколки

{

shards[i].location = new Point(rocket.middlePart[0].X + i % 25, rocket.middlePart[0].Y + i / 25);

shards[i].speed = new Point(-20 + rand.Next(41), rand.Next(21)+1);

shards[i].size = rand.Next(5) + 1;

int col = rand.Next(3);

switch (col)

{

case 0:shards[i].color = Brushes.Red;

break;

case 1:

shards[i].color = Brushes.Orange;

break;

case 2:

shards[i].color = Brushes.Blue;

break;

default:

break;

}

}

}

private void DrawRocket()

{

graph.DrawPolygon(pen, rocket.topPart);

graph.DrawPolygon(pen, rocket.downPart);

graph.DrawPolygon(pen, rocket.middlePart);

graph.FillPolygon(Brushes.Blue, rocket.topPart);

graph.FillPolygon(Brushes.Orange, rocket.downPart);

graph.FillPolygon(Brushes.Red, rocket.middlePart);

}

private void ChangeLocationRocket()

{

for(int i=0;i<4;i++)

{

if(i<3)

{

rocket.downPart[i].Y-= rocket.speed;

rocket.middlePart[i].Y-= rocket.speed;

rocket.topPart[i].Y-= rocket.speed;

}

else

{

rocket.downPart[i].Y-= rocket.speed;

rocket.middlePart[i].Y-= rocket.speed;

}

}

}

private void DrawShards()

{

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

{

if (shards[i].location.X + shards[i].speed.X > 0 && shards[i].location.X + shards[i].speed.X < xMax &&

shards[i].location.Y + shards[i].speed.Y > 0 && shards[i].location.Y + shards[i].speed.Y < yMax)

{

shards[i].location.X += shards[i].speed.X;

shards[i].location.Y += shards[i].speed.Y;

graph.FillRectangle(shards[i].color, shards[i].location.X, shards[i].location.Y, shards[i].size, shards[i].size);

if (shards[i].speed.X != 0) shards[i].speed.X = (shards[i].speed.X >= 0) ? shards[i].speed.X -= 1 : -1 \* (Math.Abs(shards[i].speed.X) - 1);

if (shards[i].speed.Y != 0) shards[i].speed.Y = (shards[i].speed.Y >= 0) ? shards[i].speed.Y -= 1 : -1 \* (Math.Abs(shards[i].speed.Y) - 1);

}

else

{

shards[i].speed.X = 0;

shards[i].speed.Y = 0;

}

}

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

if (timer1.Enabled) timer1.Stop();

timer1.Start();

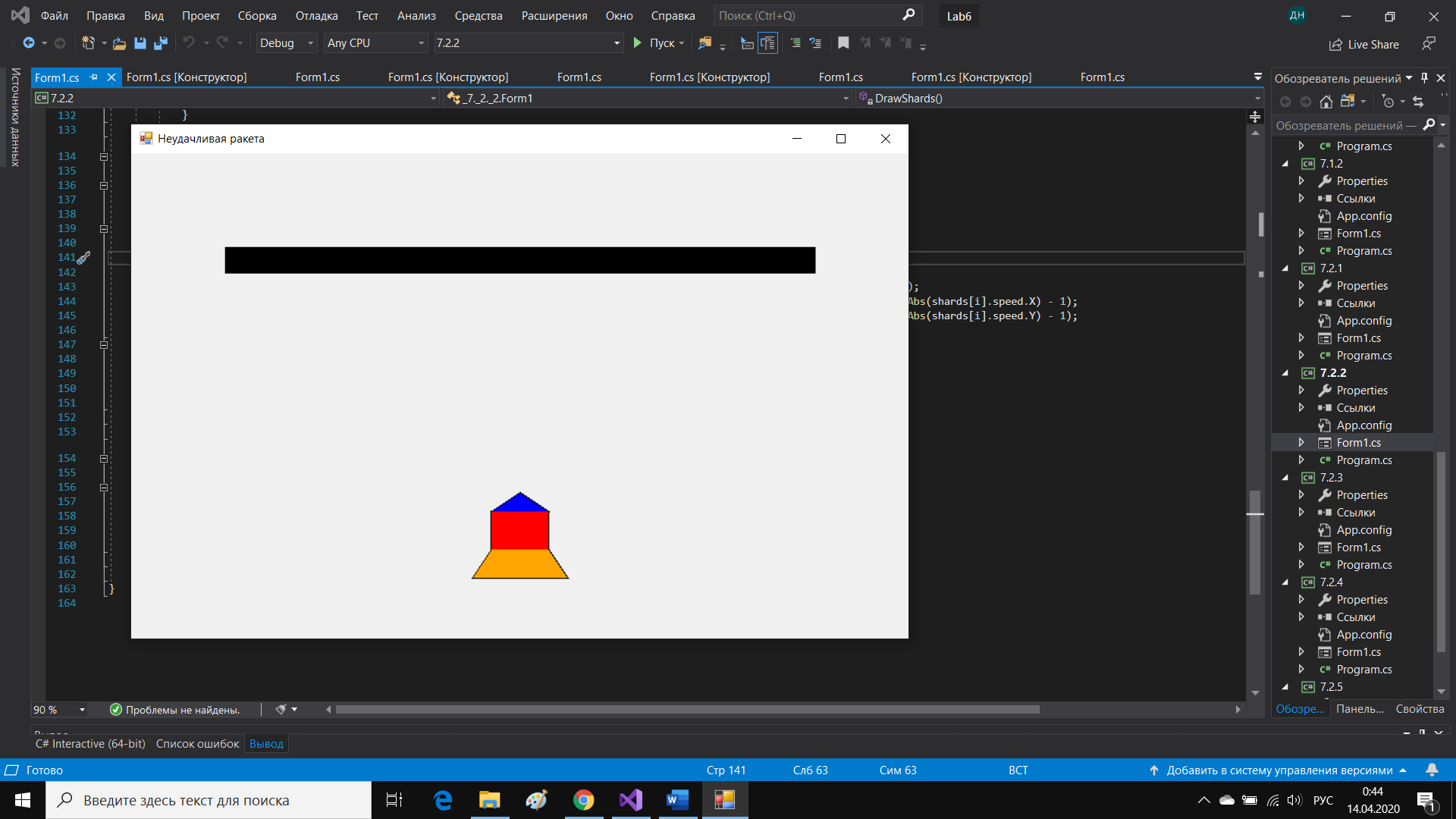
}

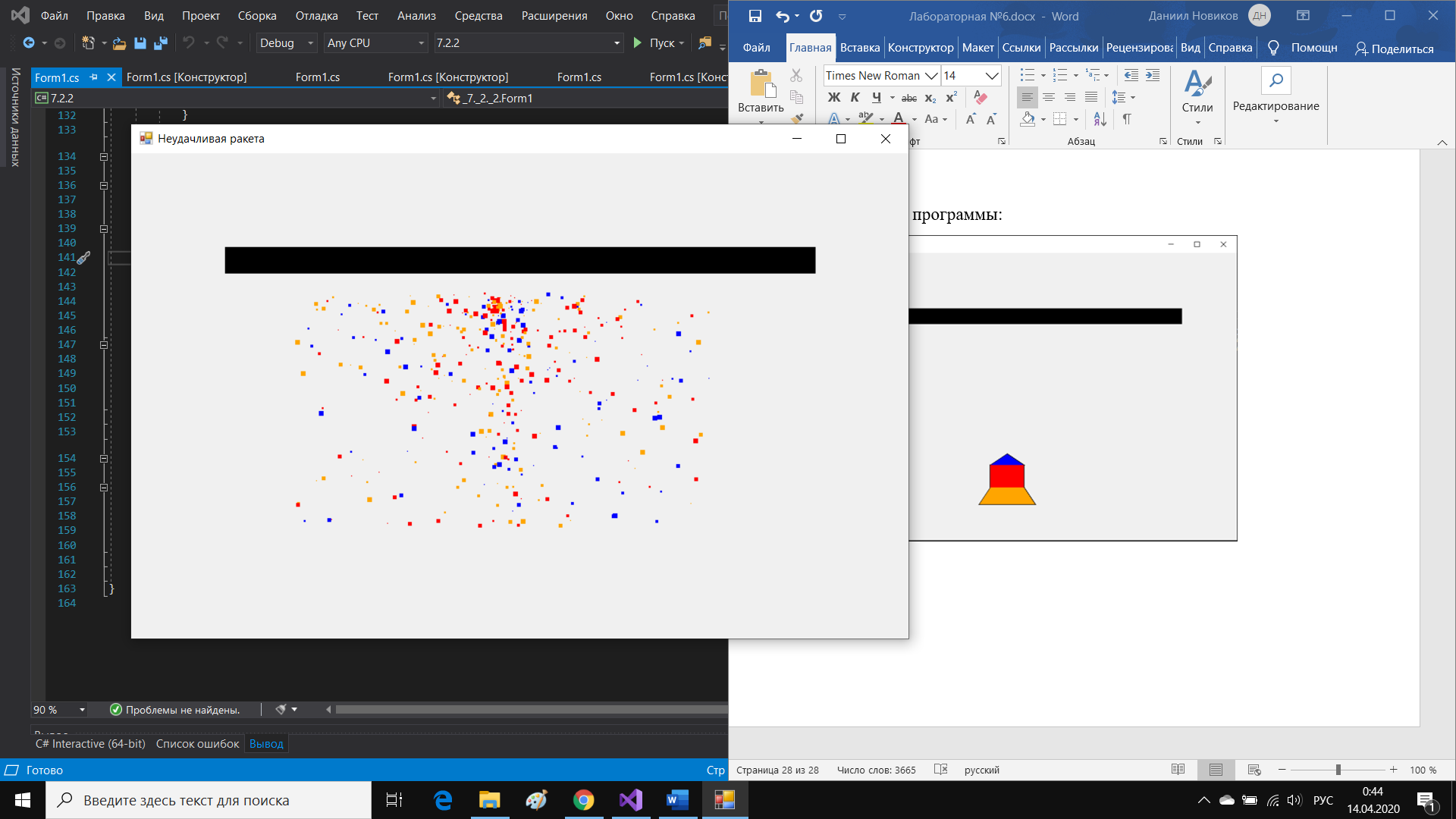
}

}

}

Результат работы программы:





Задание 7.2.3

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;

namespace \_7.\_2.\_3

{

public partial class Form1 : Form

{

struct Shard//свойства осколка

{

public Point location;

public Point speed;

public int size;

}

struct Bomb

{

public Point location;//местоположение

public int r;//размер бомбы

public int n;//количество осколков

public Shard[] shards;//осколки

}

Bomb bomb1;//первая бомба

Bomb bomb2;//вторая бомба

Graphics graph;

Bitmap bmp;

Random rand;

int xMax, yMax;

int seconds = 50;//с 3-ей секунды начинаем взрывать

bool checkTouch;//чек на касание оскольком второй бомбы

bool firstRun;//чек на то, чтобы не было после поиска на наличие касаний после того как уже нашли касание

public Form1()

{

InitializeComponent();

checkTouch = false;

firstRun = false;

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

graph = Graphics.FromImage(bmp);

rand = new Random();

}

private void FillingInTheInitialData(ref Bomb bomb)

{

bomb.r = rand.Next(50, 60);

bomb.n = rand.Next(200, 300);

bomb.shards = new Shard[bomb.n];

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

{

bomb.shards[i].location = new Point(bomb.location.X + i % bomb.r, bomb.location.Y + i / bomb.r);

bomb.shards[i].speed = new Point(-25 + rand.Next(51), -25 + rand.Next(51));

bomb.shards[i].size = rand.Next(4) + 1;

}

}

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

{

xMax = bmp.Width;

yMax = bmp.Height;

bomb1 = new Bomb();

bomb2 = new Bomb();

bomb1.location = new Point(xMax / 3, yMax / 2);

bomb2.location = new Point(2 \* xMax / 3, yMax / 2);

FillingInTheInitialData(ref bomb1);

FillingInTheInitialData(ref bomb2);

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

if (timerFirstBomb.Enabled) timerFirstBomb.Stop();

timerFirstBomb.Start();

}

}

private void timerFirstBomb\_Tick(object sender, EventArgs e)

{

--seconds;

graph.Clear(BackColor);

if (seconds < 0)

{

if(!firstRun)checkTouch = CheckTouch();

DrawShards(bomb1);

if (checkTouch)

{

firstRun = true;

DrawShards(bomb2);

}

}

else

{

if (seconds - 1 > 0)

graph.FillEllipse(Brushes.Black, bomb1.location.X, bomb1.location.Y, bomb1.r, bomb1.r);

else

graph.FillEllipse(Brushes.White, bomb1.location.X, bomb1.location.Y, bomb1.r, bomb1.r);

}

if (!checkTouch) graph.FillEllipse(Brushes.Black, bomb2.location.X, bomb2.location.Y, bomb2.r, bomb2.r);

pictureBox1.Image = bmp;

}

private void DrawShards(Bomb bomb)

{

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

{

if (bomb.shards[i].location.X + bomb.shards[i].speed.X > 0 && bomb.shards[i].location.X + bomb.shards[i].speed.X < xMax &&

bomb.shards[i].location.Y + bomb.shards[i].speed.Y > 0 && bomb.shards[i].location.Y + bomb.shards[i].speed.Y < yMax)

{

bomb.shards[i].location.X += bomb.shards[i].speed.X;

bomb.shards[i].location.Y += bomb.shards[i].speed.Y;

graph.FillEllipse(Brushes.Black, bomb.shards[i].location.X, bomb.shards[i].location.Y, bomb.shards[i].size, bomb.shards[i].size);

if (bomb.shards[i].speed.X != 0) bomb.shards[i].speed.X = (bomb.shards[i].speed.X >= 0) ? bomb.shards[i].speed.X -= 1 : -1 \* (Math.Abs(bomb.shards[i].speed.X) - 1);

if (bomb.shards[i].speed.Y != 0) bomb.shards[i].speed.Y = (bomb.shards[i].speed.Y >= 0) ? bomb.shards[i].speed.Y -= 1 : -1 \* (Math.Abs(bomb.shards[i].speed.Y) - 1);

}

else

{

bomb.shards[i].speed.X = 0;

bomb.shards[i].speed.Y = 0;

}

}

}

private bool CheckTouch()

{

for (int i = 0; i < bomb1.n; i++)

{

if (bomb1.shards[i].location.X + bomb1.shards[i].size > bomb2.location.X && bomb1.shards[i].location.X < bomb2.location.X + bomb2.r &&

bomb1.shards[i].location.Y + bomb1.shards[i].size > bomb2.location.Y && bomb1.shards[i].location.Y < bomb2.location.Y + bomb2.r) return true;

}

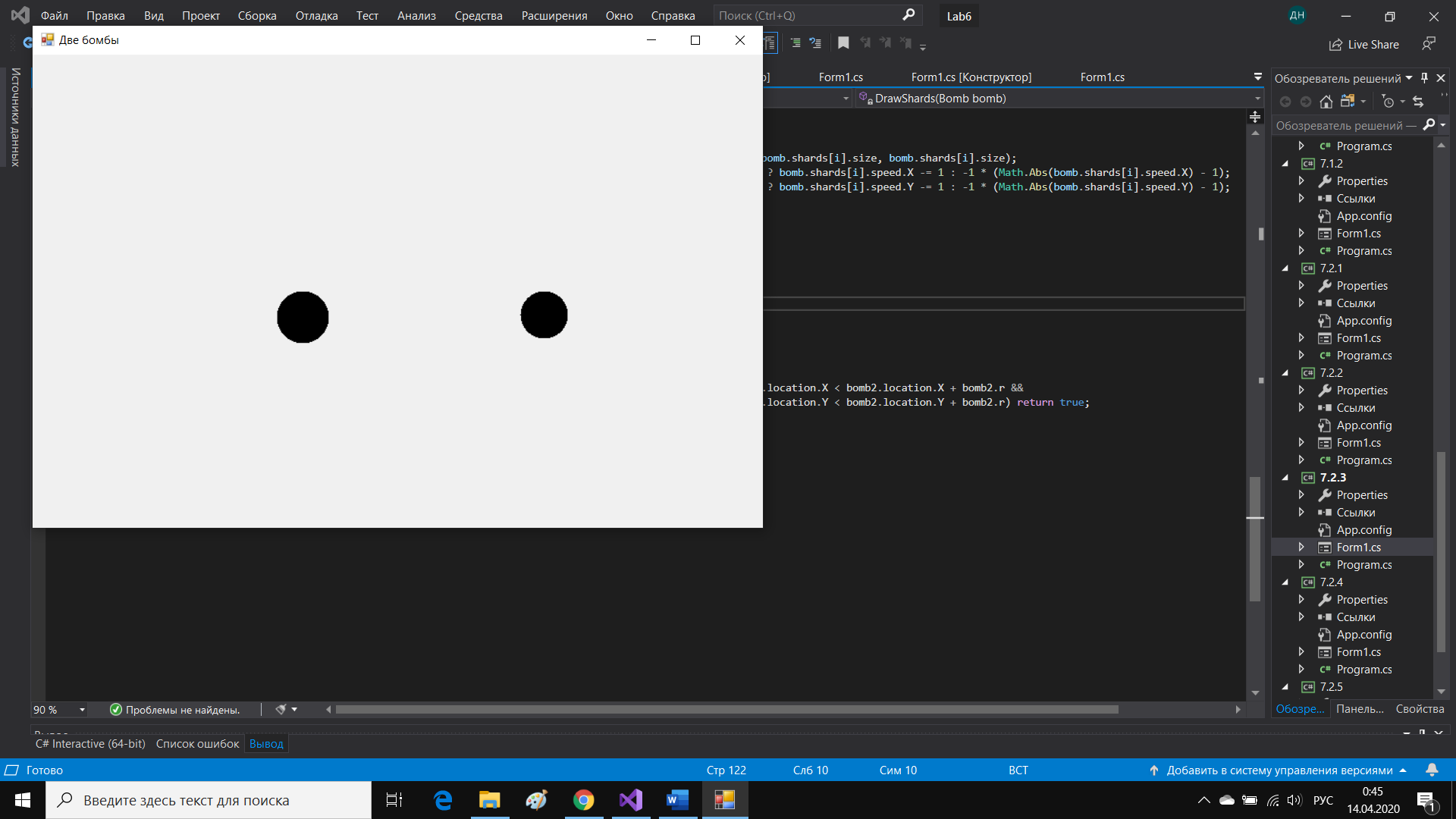
return false;

}

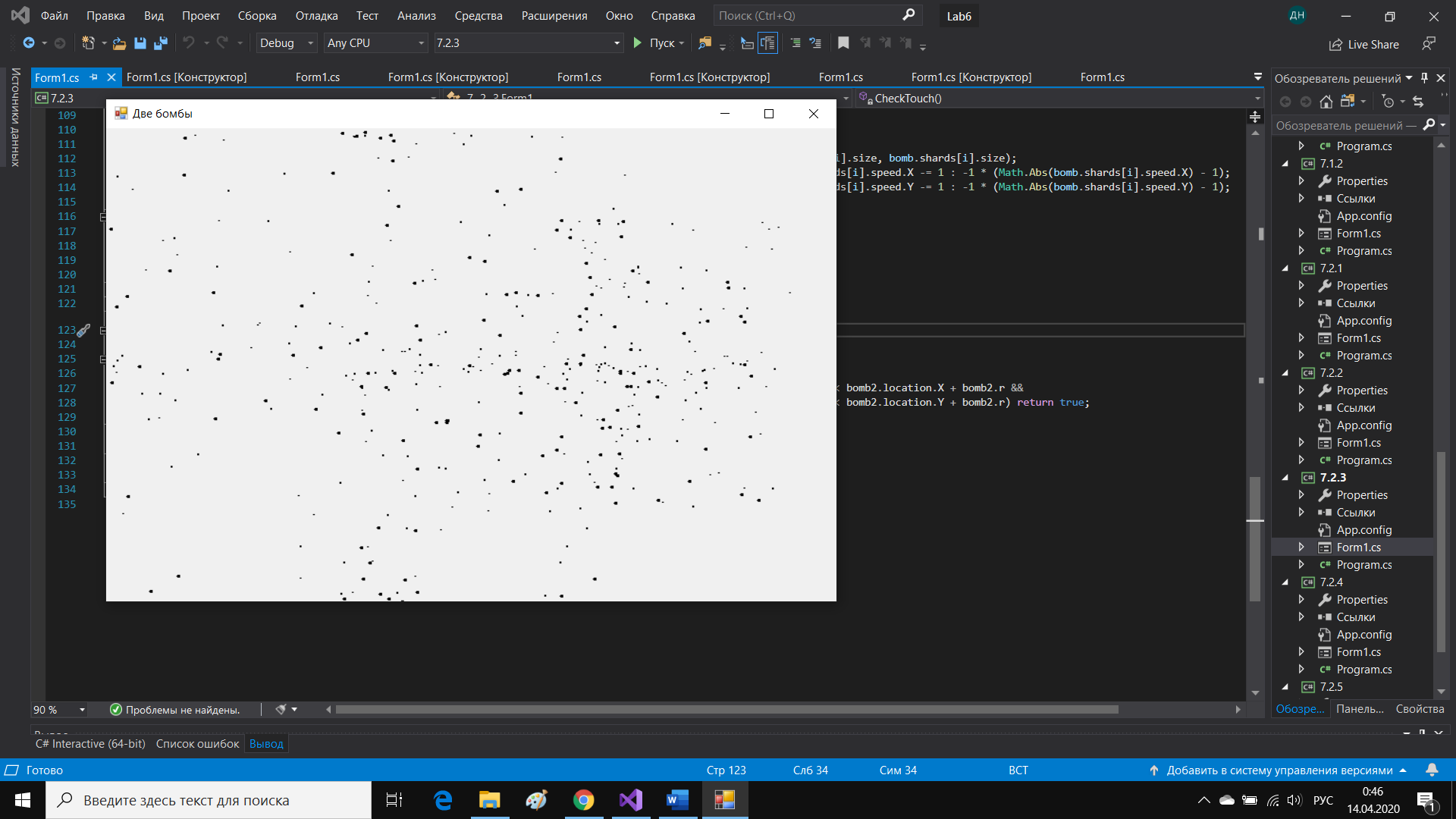
}

}

Результат работы программы:







Задание 7.2.4

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;

namespace \_7.\_2.\_4

{

public partial class Form1 : Form

{

struct Shard//свойства осколка

{

public Point location;

public Point speed;

public int size;

}

int n = 400;

Shard[] shards;

Graphics graph;

Bitmap bmp;

Random rand;

int xMax, yMax;

public Form1()

{

InitializeComponent();

shards = new Shard[n];

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

graph = Graphics.FromImage(bmp);

rand = new Random();

}

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

{

xMax = bmp.Width;

yMax = bmp.Height;

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

{

shards[i].location = new Point(xMax/2 + i % 25, yMax/2 + i / 25);

shards[i].speed = new Point(-19 + rand.Next(38), -19 + rand.Next(38));

shards[i].size = rand.Next(4) + 1;

}

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

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

graph.FillRectangle(Brushes.Black, shards[i].location.X, shards[i].location.Y, 1, 1);

pictureBox1.Image = bmp;

if (timer1.Enabled) timer1.Stop();

timer1.Start();

}

}

private void DrawShards()

{

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

{

if (shards[i].location.X + shards[i].speed.X > 0 && shards[i].location.X + shards[i].speed.X < xMax &&

shards[i].location.Y + shards[i].speed.Y > 0 && shards[i].location.Y + shards[i].speed.Y < yMax)

{

shards[i].location.X += shards[i].speed.X;

shards[i].location.Y += shards[i].speed.Y;

graph.FillEllipse(Brushes.Black, shards[i].location.X, shards[i].location.Y, shards[i].size, shards[i].size);

if (shards[i].speed.Y>0) shards[i].speed.X = (shards[i].speed.X >= 0) ? shards[i].speed.X -= 1 : -1 \* (Math.Abs(shards[i].speed.X) - 1);

shards[i].speed.Y++;

}

else

{

shards[i].speed.X = 0;

shards[i].speed.Y++;

}

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

graph.Clear(BackColor);

DrawShards();

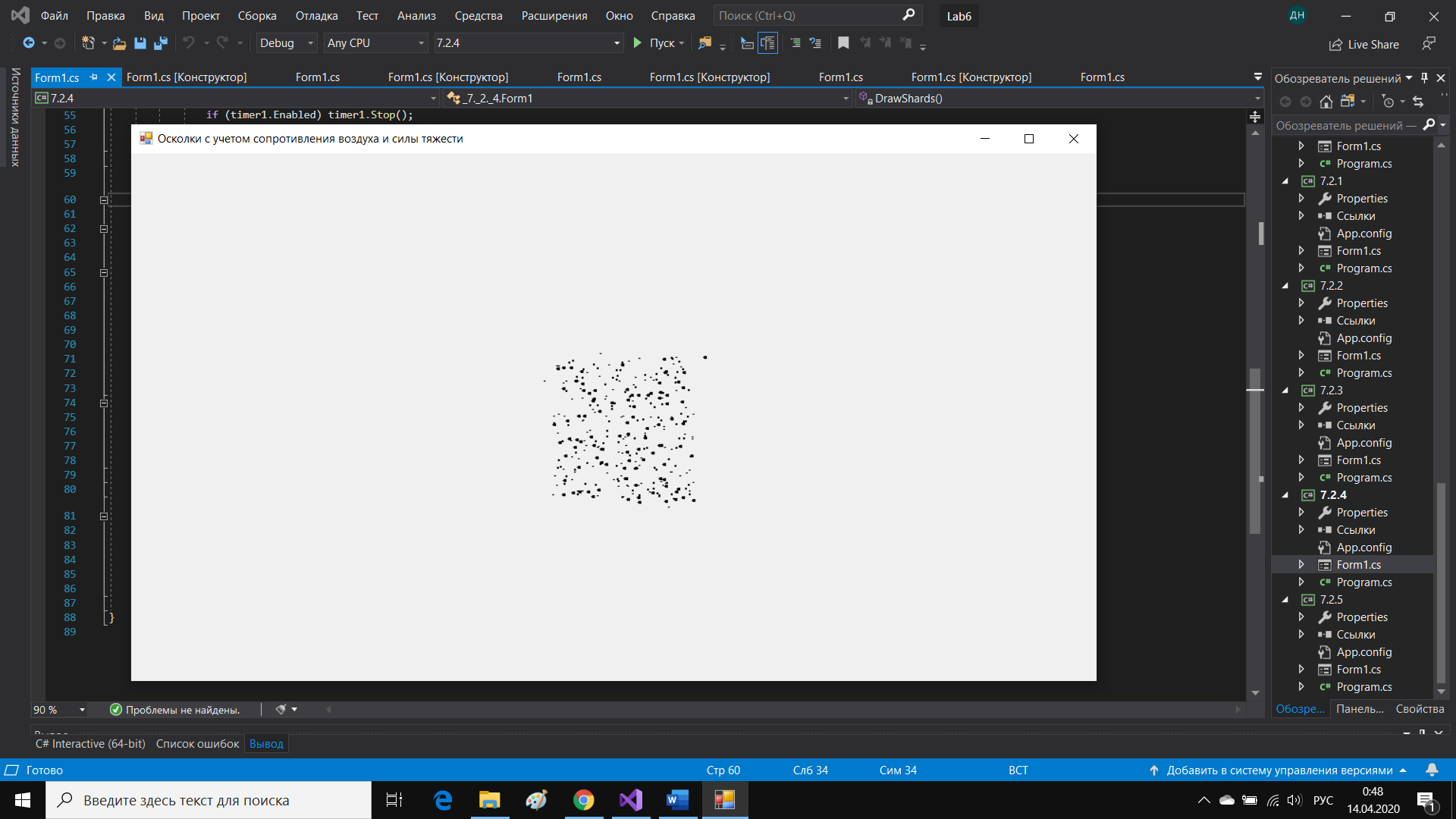
pictureBox1.Image = bmp;

}

}

}

Результат работы программы:





Задание 7.2.5

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 \_7.\_2.\_5

{

public partial class Form1 : Form

{

struct Shard//свойства осколка

{

public Point location;

public Point speed;

public int size;

}

struct FireWork

{

public int n;

public Point location;

public Shard[] shards;

public Brush color;

}

FireWork fire;

Graphics graph;

Bitmap bmp;

Random rand;

int xMax, yMax;

int sec1,sec2;//sec1-время для огоньков , sec2-время между салютами

public Form1()

{

InitializeComponent();

sec1 = 12;//время действия одного салюта

sec2 = 20;

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

graph = Graphics.FromImage(bmp);

rand = new Random();

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

if (timer1.Enabled) timer1.Stop();

timer1.Start();

}

}

private void Initialization()

{

fire.n = rand.Next(50, 60);

fire.location = new Point(rand.Next(50, xMax - 50), rand.Next(30, yMax / 2));

fire.shards = new Shard[fire.n];

int tmp = rand.Next(7) + 9;

switch (tmp)

{

case 9:

fire.color = Brushes.Blue;

break;

case 10:

fire.color = Brushes.Purple;

break;

case 11:

fire.color = Brushes.Orange;

break;

case 12:

fire.color = Brushes.Red;

break;

case 13:

fire.color = Brushes.Magenta;

break;

case 14:

fire.color = Brushes.Yellow;

break;

case 15:

fire.color = Brushes.Green;

break;

}

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

{

fire.shards[i].location = new Point(fire.location.X + i % 25, fire.location.Y + i / 25);

fire.shards[i].speed = new Point(-10 + rand.Next(21), -10 + rand.Next(21));

fire.shards[i].size = rand.Next(4) + 1;

}

}

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

{

xMax = bmp.Width;

yMax = bmp.Height;

fire = new FireWork();

Initialization();

}

private void timer1\_Tick(object sender, EventArgs e)

{

graph.Clear(BackColor);

--sec2;

if (sec2 >= 0)

{

--sec1;

if (sec1 >= 0)

DrawShards();

}

else

{

Initialization();

sec1 = 12;

sec2 = 20;

}

pictureBox1.Image = bmp;

}

private void DrawShards()

{

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

{

if (fire.shards[i].location.X + fire.shards[i].speed.X > 0 && fire.shards[i].location.X + fire.shards[i].speed.X < xMax &&

fire.shards[i].location.Y + fire.shards[i].speed.Y > 0 && fire.shards[i].location.Y + fire.shards[i].speed.Y < yMax)

{

fire.shards[i].location.X += fire.shards[i].speed.X;

fire.shards[i].location.Y += fire.shards[i].speed.Y;

graph.FillEllipse(fire.color, fire.shards[i].location.X, fire.shards[i].location.Y, fire.shards[i].size, fire.shards[i].size);

if (fire.shards[i].speed.Y > 0) fire.shards[i].speed.X = (fire.shards[i].speed.X >= 0) ? fire.shards[i].speed.X -= 1 : -1 \* (Math.Abs(fire.shards[i].speed.X) - 1);

fire.shards[i].speed.Y++;

}

else

{

fire.shards[i].speed.X = 0;

fire.shards[i].speed.Y++;

}

}

}

}

}

Результат работы программы:

