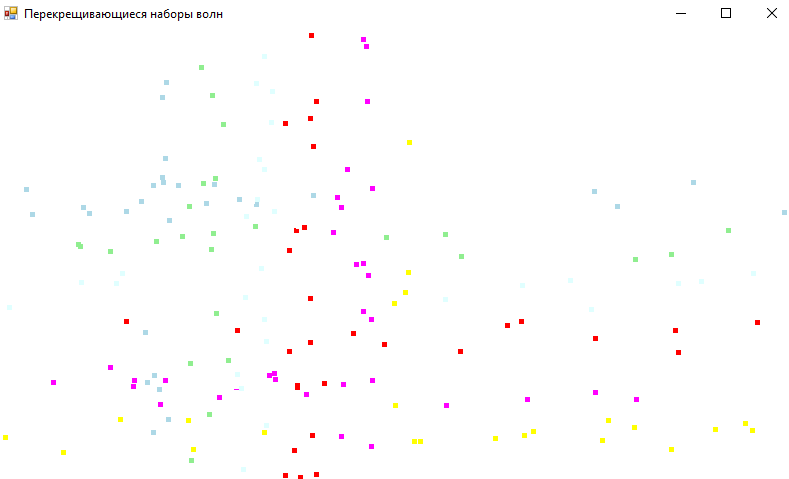
Лабораторная работа №5

Фиб-2301

Задание 3

1)





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 \_3.\_1

{

public partial class Form1 : Form

{

Graphics graph;

Random rand;

Point[] obj;

Point[] vector;

Brush[] color;

int n = 50;

int xMax, yMax;

List<Brush[]> c;

List<Point[]> v;

List<Point[]> objected;

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

{

for (int z = 0; z < 2; z++)

{

obj = new Point[n];

vector = new Point[n];

color = new Brush[n];

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

{

if (z == 0)

obj[i] = new Point(rand.Next(xMax), rand.Next(40));

else

obj[i] = new Point(rand.Next(40), rand.Next(yMax));

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

switch (tmp)

{

case 9:

color[i] = Brushes.LightBlue;

break;

case 10:

color[i] = Brushes.LightGreen;

break;

case 11:

color[i] = Brushes.LightCyan;

break;

case 12:

color[i] = Brushes.Red;

break;

case 13:

color[i] = Brushes.Magenta;

break;

case 14:

color[i] = Brushes.Yellow;

break;

case 15:

color[i] = Brushes.Black;

break;

}

if (z == 0)

vector[i] = new Point(-2 + rand.Next(5), tmp - 6);

else

vector[i] = new Point(tmp - 6, -2 + rand.Next(5));

}

c.Add(color);

objected.Add(obj);

v.Add(vector);

}

timer1.Start();

}

private void Timer1\_Tick(object sender, EventArgs e)

{

for (int z = 0; z < 2; z++)

{

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

{

if (objected[z][i].X + v[z][i].X < 0 || objected[z][i].X + v[z][i].X > xMax) v[z][i].X = -v[z][i].X;

if (objected[z][i].Y + v[z][i].Y < 0 || objected[z][i].Y + v[z][i].Y > yMax) v[z][i].Y = -v[z][i].Y;

graph.FillRectangle(Brushes.White, objected[z][i].X, objected[z][i].Y, 5, 5);

objected[z][i].X += v[z][i].X;

objected[z][i].Y += v[z][i].Y;

graph.FillRectangle(c[z][i], objected[z][i].X, objected[z][i].Y, 5, 5);

}

}

}

public Form1()

{

InitializeComponent();

c = new List<Brush[]>();

v = new List<Point[]>();

objected = new List<Point[]>();

graph = CreateGraphics();

xMax = ClientSize.Width;

yMax = ClientSize.Height;

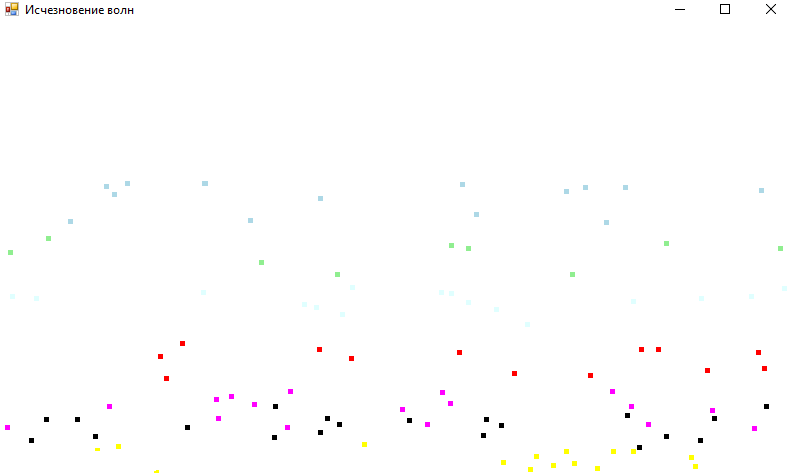
rand = new Random();

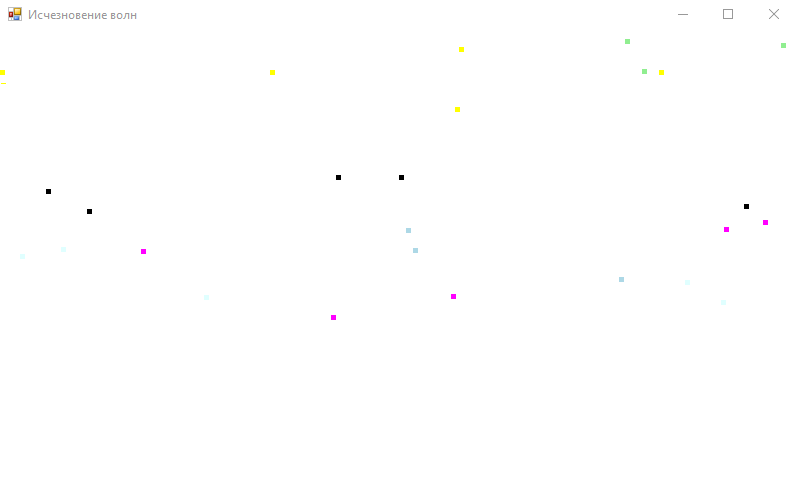
}

}

}

3)





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 \_3.\_3

{

public partial class Form1 : Form

{

Bitmap raster;

Random rand;

Point[] objected;

Point[] vector;

Brush[] color;

int n = 200;

int Max\_x, Max\_y;

bool[] clear;

private bool CheckTouch(int j)

{

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

if (!clear[i] && i != j && (objected[i].Y < objected[j].Y + 5) && (objected[i].Y > objected[j].Y) && (objected[i].X > objected[j].X) && (objected[i].X < objected[j].X + 5)) return true;

return false;

}

private void Timer1\_Tick(object sender, EventArgs e)

{

using (Graphics graph = Graphics.FromImage(raster))

{

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

{

if (objected[i].X + vector[i].X < 0 || objected[i].X + vector[i].X > Max\_x) vector[i].X = -vector[i].X;

if (objected[i].Y + vector[i].Y < 0 || objected[i].Y + vector[i].Y > Max\_y) vector[i].Y = -vector[i].Y;

if (!clear[i])

graph.FillRectangle(Brushes.White, objected[i].X, objected[i].Y, 5, 5);

objected[i].X += vector[i].X;

objected[i].Y += vector[i].Y;

if (!clear[i] && !CheckTouch(i))

{

graph.FillRectangle(color[i], objected[i].X, objected[i].Y, 5, 5);

}

else

clear[i] = true;

}

}

pictureBox1.Image = raster;

}

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

{

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

{

objected[i] = new Point(rand.Next(Max\_x), rand.Next(40));

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

switch (choose)

{

case 9:

color[i] = Brushes.LightBlue;

break;

case 10:

color[i] = Brushes.LightGreen;

break;

case 11:

color[i] = Brushes.LightCyan;

break;

case 12:

color[i] = Brushes.Red;

break;

case 13:

color[i] = Brushes.Magenta;

break;

case 14:

color[i] = Brushes.Yellow;

break;

case 15:

color[i] = Brushes.Black;

break;

}

vector[i] = new Point(-2 + rand.Next(5), choose - 6);

}

timer1.Start();

}

public Form1()

{

InitializeComponent();

objected = new Point[n];

clear = new bool[n];

vector = new Point[n];

color = new Brush[n];

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

Max\_x = ClientSize.Width;

Max\_y = ClientSize.Height;

rand = new Random();

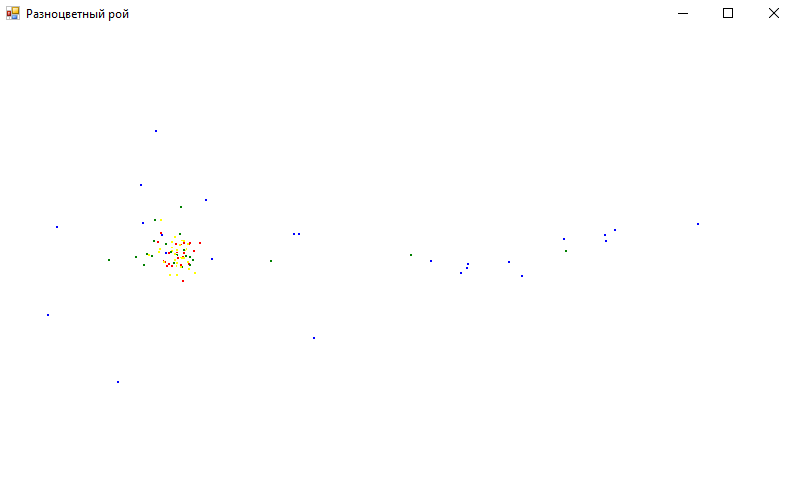
}

}

}

Задание 5

1)



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 \_5.\_1

{

public partial class Form1 : Form

{

Graphics graph;

Pen myPen;

Point[] objects;

int n = 100;

int Max\_x, Max\_y;

Random rand;

Point center;

int speed;

int dx, dy;

int[] color;

int shift;

private void Timer1\_Tick(object sender, EventArgs e)

{

if (center.X > Max\_x || center.X < 0) speed = -speed;

center.X += speed;

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

{

Brush brush = Brushes.Red;

dx = -25 + rand.Next(51);

dy = -25 + rand.Next(51);

if (color[i] == 0)

{

brush = Brushes.Red;

shift = 13;

}

else if (color[i] == 1)

{

brush = Brushes.Green;

shift = 7;

}

else if (color[i] == 2)

{

brush = Brushes.Blue;

shift = 3;

}

else if (color[i] == 3)

{

brush = Brushes.Yellow;

shift = 10;

}

graph.FillRectangle(Brushes.White, objects[i].X, objects[i].Y, 2, 2);

if (objects[i].X + dx > 0 && objects[i].X + dx < Max\_x)

{

if (objects[i].X + dx < center.X)

dx += shift;

else dx -= shift;

objects[i].X += dx;

}

if (objects[i].Y + dy > 0 && objects[i].Y + dy < Max\_y)

{

if (objects[i].Y + dy < center.Y)

dy += shift;

else dy -= shift;

objects[i].Y += dy;

}

graph.FillRectangle(brush, objects[i].X, objects[i].Y, 2, 2);

}

}

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

{

Max\_x = ClientSize.Width;

Max\_y = ClientSize.Height;

objects = new Point[n];

color = new int[n];

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

{

objects[i] = new Point(rand.Next(Max\_x), rand.Next(Max\_y));

color[i] = rand.Next(4);

}

center.X = 1;

center.Y = Max\_y / 2;

speed = 3;

timer1.Start();

}

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

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

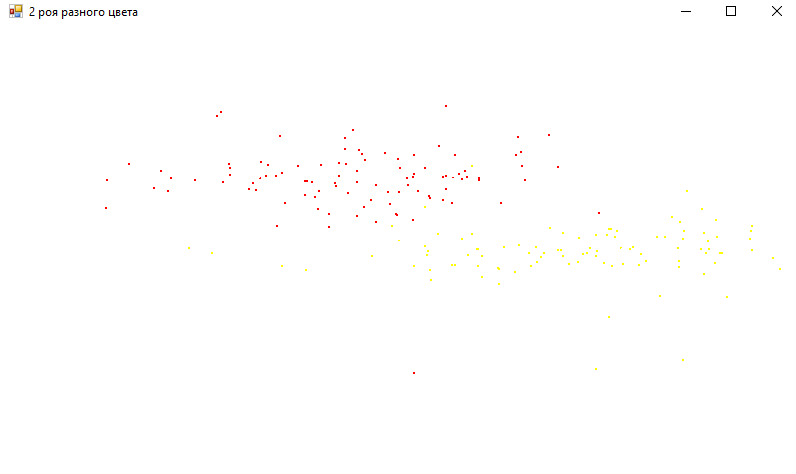
rand = new Random();

}

}

}

2)



Код:

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

{

Graphics graph;

Pen MyPen;

Point[] object\_1, object\_2;

int Max\_x, Max\_y, dx, dy, n = 100;

Random rand;

Point center\_1, center\_2;

int speed\_1 = 10, speed\_2 = 20;

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

{

Max\_x = ClientSize.Width;

Max\_y = ClientSize.Height;

object\_1 = new Point[n];

object\_2 = new Point[n];

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

{

object\_1[i] = new Point(rand.Next(Max\_x), rand.Next(Max\_y));

object\_2[i] = new Point(rand.Next(Max\_x), rand.Next(Max\_y));

}

center\_1.X = 1;

center\_1.Y = Max\_y / 2;

center\_2.X = Max\_x / 2;

center\_2.Y = Max\_y / 3;

timer1.Start();

}

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

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

rand = new Random();

}

private void Timer1\_Tick(object sender, EventArgs e)

{

Roy(ref center\_1, object\_1, Brushes.Yellow, ref speed\_1);

Roy(ref center\_2, object\_2, Brushes.Red, ref speed\_2);

}

private void Roy(ref Point center, Point[] objects, Brush brush, ref int speed)

{

if (center.X > Max\_x || center.X < 0) speed = -speed;

center.X += speed;

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

{

dx = -25 + rand.Next(51);

dy = -25 + rand.Next(51);

graph.FillRectangle(Brushes.White, objects[i].X, objects[i].Y, 2, 2);

if (objects[i].X + dx > 0 && objects[i].X + dx < Max\_x)

{

if (objects[i].X + dx < center.X)

dx += 5;

else dx -= 5;

objects[i].X += dx;

}

if (objects[i].Y + dy > 0 && objects[i].Y + dy < Max\_y)

{

if (objects[i].Y + dy < center.Y)

dy += 5;

else dy -= 5;

objects[i].Y += dy;

}

graph.FillRectangle(brush, objects[i].X, objects[i].Y, 2, 2);

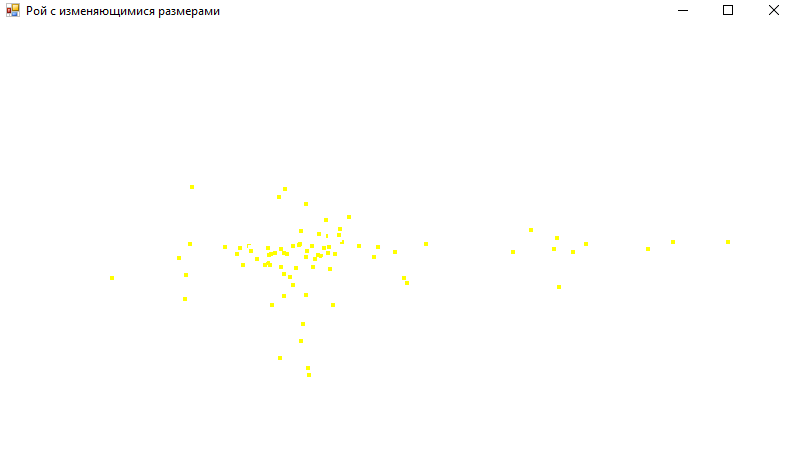
}

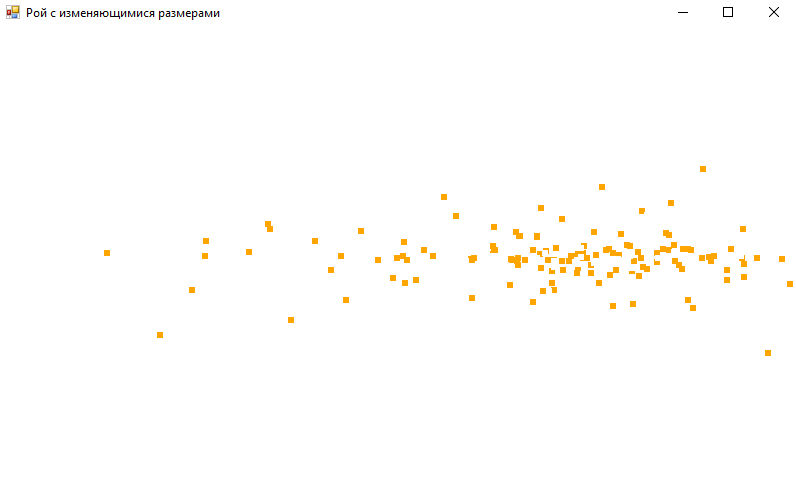
}

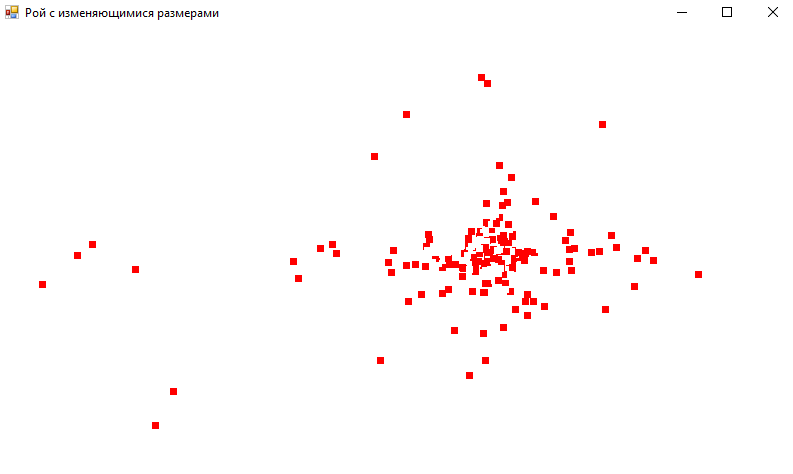
}

3)









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 \_5.\_3

{

public partial class Form1 : Form

{

Graphics graph;

Pen MyPen;

Point[] objected;

Random rand;

Brush brush;

Point center;

int speed = 3, dx, dy, Max\_x, Max\_y, shift = 7, seconds, size = 1, n = 30;

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

{

timer1.Start();

Max\_x = ClientSize.Width;

Max\_y = ClientSize.Height;

objected = new Point[1000];

for (int i = 0; i < 1000; i++)

{

objected[i] = new Point(rand.Next(Max\_x), rand.Next(Max\_y));

}

center.X = 1;

center.Y = Max\_y / 2;

brush = Brushes.Black;

}

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

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

rand = new Random();

}

private void Timer1\_Tick(object sender, EventArgs e)

{

seconds++;

if (center.X > Max\_x || center.X < 0) speed = -speed;

center.X += speed;

if (seconds == 50)

{

brush = Brushes.Yellow;

n += 50;

size += 3;

speed += 4;

}

else if (seconds == 100)

{

brush = Brushes.Orange;

n += 50;

size += 2;

speed += 7;

}

else if (seconds == 150)

{

brush = Brushes.Red;

size += 1;

n += 50;

speed += 20;

}

else if (seconds == 200)

{

brush = Brushes.Green;

n += 50;

size += 2;

speed -= 30;

}

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

{

dx = -25 + rand.Next(51);

dy = -25 + rand.Next(51);

graph.FillRectangle(Brushes.White, objected[i].X, objected[i].Y, size, size);

if (objected[i].X + dx > 0 && objected[i].X + dx < Max\_x)

{

if (objected[i].X + dx < center.X)

dx += shift;

else dx -= shift;

objected[i].X += dx;

}

if (objected[i].Y + dy > 0 && objected[i].Y + dy < Max\_y)

{

if (objected[i].Y + dy < center.Y)

dy += shift;

else dy -= shift;

objected[i].Y += dy;

}

graph.FillRectangle(brush, objected[i].X, objected[i].Y, size, size);

}

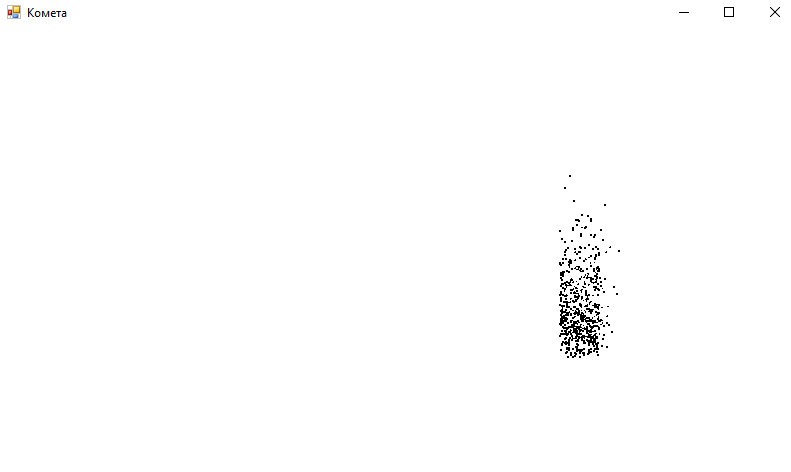
}

}

}

4)





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 \_5.\_4

{

public partial class Form1 : Form

{

Graphics Graph;

Pen MyPen;

Point[] object\_;

int n = 800, r, shift;

double t = 0;

int Max\_x, Max\_y, x0, y0, dx, dy;

Random rand;

Point center;

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

{

x0 = ClientSize.Width / 2;

y0 = ClientSize.Height / 2;

Max\_x = ClientSize.Width;

Max\_y = ClientSize.Height;

object\_ = new Point[n];

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

{

object\_[i] = new Point(rand.Next(Max\_x), rand.Next(Max\_y));

}

r = 200;

shift = 20;

center.X = 1;

center.Y = Max\_y / 2;

timer1.Start();

}

private void Timer1\_Tick(object sender, EventArgs e)

{

center.X = (int)(x0 + r \* Math.Cos(t));

center.Y = (int)(y0 + r \* Math.Sin(t));

t += 0.1;

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

{

dx = -25 + rand.Next(51);

dy = -25 + rand.Next(51);

Graph.FillRectangle(Brushes.White, object\_[i].X, object\_[i].Y, 2, 2);

if (object\_[i].X + dx > 0 && object\_[i].X + dx < Max\_x)

{

if (object\_[i].X + dx < center.X)

dx += shift;

else dx -= shift;

object\_[i].X += dx;

}

if (object\_[i].Y + dy > 0 && object\_[i].Y + dy < Max\_y)

{

if (object\_[i].Y + dy < center.Y)

dy += shift;

else dy -= shift;

object\_[i].Y += dy;

}

Graph.FillRectangle(Brushes.Black, object\_[i].X, object\_[i].Y, 2, 2);

}

}

public Form1()

{

InitializeComponent();

Graph = CreateGraphics();

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

rand = new Random();

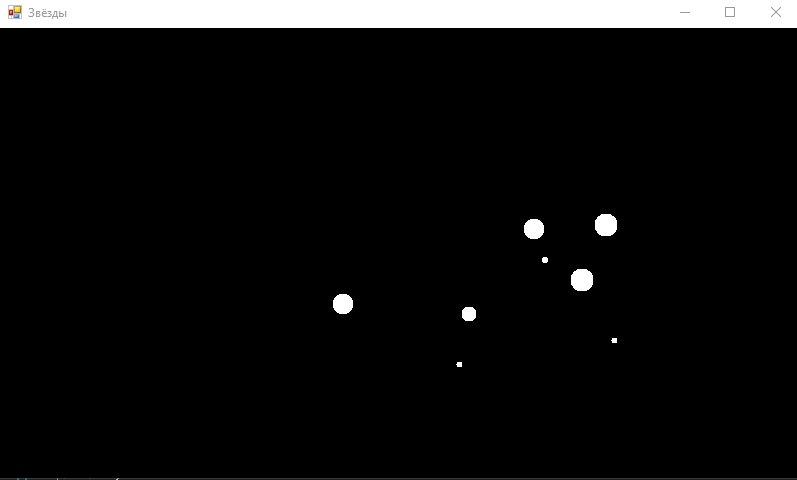
}

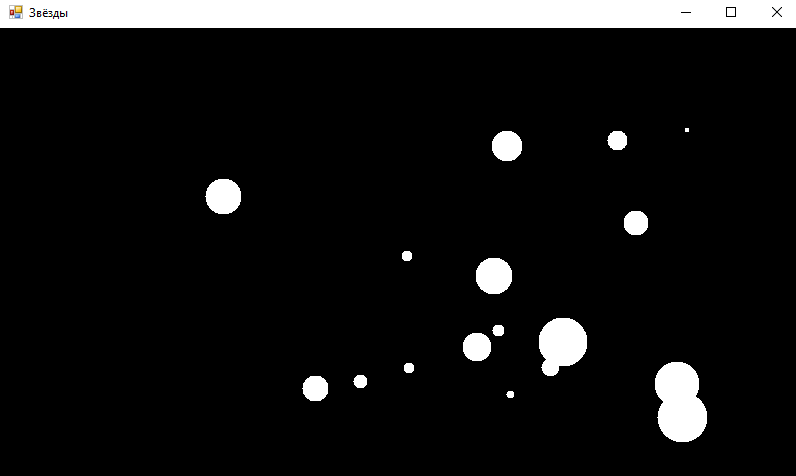
}

}

Задание 7

3)





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;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_7.\_3

{

public partial class Form1 : Form

{

Graphics graph;

Random rand;

Pen myPen;

int[] x, y, z, size, x1, y1;

int center\_X, center\_Y, n = 15, view = 200, speed = 50;

private void Timer1\_Tick(object sender, EventArgs e)

{

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

{

if (z[i] / (Math.Abs(x[i] + 1)) > view / center\_X && z[i] / (Math.Abs(y[i] + 1)) > view / center\_Y)

{

x1[i] = center\_X + (int)Math.Round(x[i] \* ((double)view / (double)z[i]));

y1[i] = center\_Y + (int)Math.Round(y[i] \* ((double)view / (double)z[i]));

graph.FillEllipse(Brushes.White, x1[i], y1[i], size[i], size[i]);

}

else

{

z[i] = view + rand.Next(20 \* view);

size[i] = 3;

}

}

Thread.Sleep(160);

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

{

z[i] -= speed;

size[i]++;

}

graph.Clear(BackColor);

}

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

{

timer1.Start();

x = new int[n];

y = new int[n];

x1 = new int[n];

y1 = new int[n];

size = new int[n];

z = new int[n];

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

{

x[i] = -10 \* view + rand.Next(20 \* view);

y[i] = -10 \* view + rand.Next(20 \* view);

z[i] = view + rand.Next(20 \* view);

size[i] = 3;

}

center\_X = ClientSize.Width / 2;

center\_Y = ClientSize.Height / 2;

}

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

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

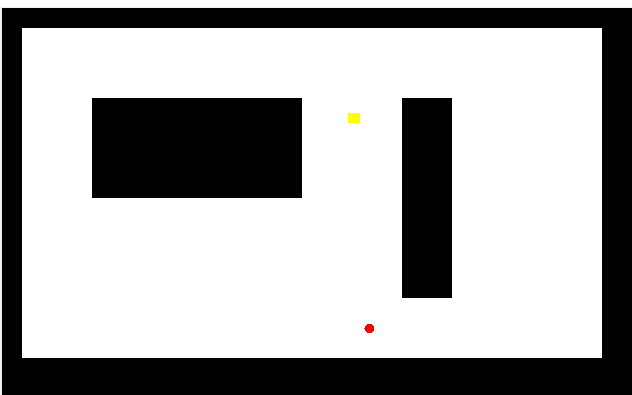
rand = new Random();

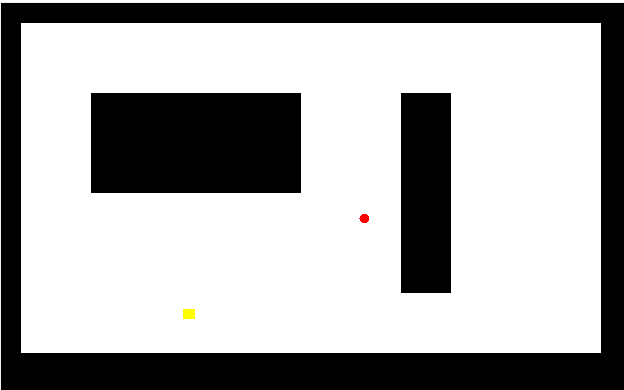
}

}

Задание 9

1)





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;

using System.Threading;

namespace \_9.\_1

{

public partial class Form1 : Form

{

Bitmap raster;

Pen myPen;

Point oldSize;

List<Objects> objects;

bool iy, ix;

List<Point[]> figures;

private void Timer1\_Tick(object sender, EventArgs e)

{

using (Graphics graph = Graphics.FromImage(raster))

{

graph.Clear(Color.FromArgb(255, 0, 0, 0));

DrawFigures(graph);

pictureBox1.Image = raster;

for (int z = 0; z < objects.Count; z++)

{

oldSize.X = objects[z].rx;

oldSize.Y = objects[z].ry;

objects[z].vector.X = objects[z].dx / Math.Abs(objects[z].dx);

objects[z].vector.Y = objects[z].dy / Math.Abs(objects[z].dy);

iy = false;

ix = false;

for (int i = 0; i < objects[z].lx; i++)

{

if (raster.GetPixel(objects[z].location.X + (objects[z].rx + i) \* objects[z].vector.X, objects[z].location.Y) == Color.FromArgb(255, 0, 0, 0))

{

objects[z].dx = -objects[z].dx;

ix = true;

break;

}

}

for (int i = 0; i < objects[z].ly; i++)

{

if (objects[z].location.Y + (objects[z].ry + i) \* objects[z].vector.Y < 0 || raster.GetPixel(objects[z].location.X, objects[z].location.Y + (objects[z].ry + i) \* objects[z].vector.Y) == Color.FromArgb(255, 0, 0, 0))

{

objects[z].dy = -objects[z].dy;

iy = true;

break;

}

}

if (!ix && !iy)

for (int i = 0; i < objects[z].lx; i++)

for (int j = 0; j < objects[z].ly; j++)

if (raster.GetPixel(objects[z].location.X + (objects[z].rx + i) \* objects[z].vector.X, objects[z].location.Y + (objects[z].ry + i) \* objects[z].vector.Y) == Color.FromArgb(255, 0, 0, 0))

{

objects[z].dy = -objects[z].dy;

objects[z].dx = -objects[z].dx;

break;

}

objects[z].location.X += objects[z].dx;

objects[z].location.Y += objects[z].dy;

if (z % 2 != 0)

graph.FillEllipse(objects[z].brush, objects[z].location.X, objects[z].location.Y, objects[z].rx, objects[z].ry);

else

graph.FillRectangle(objects[z].brush, objects[z].location.X, objects[z].location.Y, objects[z].rx + 2, objects[z].ry);

}

}

pictureBox1.Image = raster;

}

class Objects

{

public int dx, dy;

public int rx, ry;

public Point location;

public int lx, ly;

public Point vector;

public Brush brush;

public Objects(Brush brsh)

{

dx = 4; dy = 3;

rx = 10; ry = 10;

location.X = 150; location.Y = 30;

lx = Math.Abs(dx);

ly = Math.Abs(dy);

vector.X = dx / Math.Abs(dx);

vector.Y = dy / Math.Abs(dy);

brush = brsh;

}

}

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

{

timer1.Start();

Objects a = new Objects(Brushes.Yellow);

Objects b = new Objects(Brushes.Red);

a.dx = -b.dx;

objects = new List<Objects>();

objects.Add(a);

objects.Add(b);

figures = new List<Point[]>();

figures.Add(new Point[4] { new Point(20, 20), new Point(600, 20), new Point(600, 350), new Point(20, 350) });

figures.Add(new Point[4] { new Point(90, 90), new Point(300, 90), new Point(300, 190), new Point(90, 190) });

figures.Add(new Point[4] { new Point(400, 90), new Point(450, 90), new Point(450, 290), new Point(400, 290) });

pictureBox1.BackColor = Color.FromArgb(255, 0, 0, 0);

}

private void DrawFigures(Graphics graph)

{

graph.FillPolygon(Brushes.White, figures[0]);

for (int i = 1; i < figures.Count; i++)

graph.FillPolygon(Brushes.Black, figures[i]);

}

public Form1()

{

InitializeComponent();

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

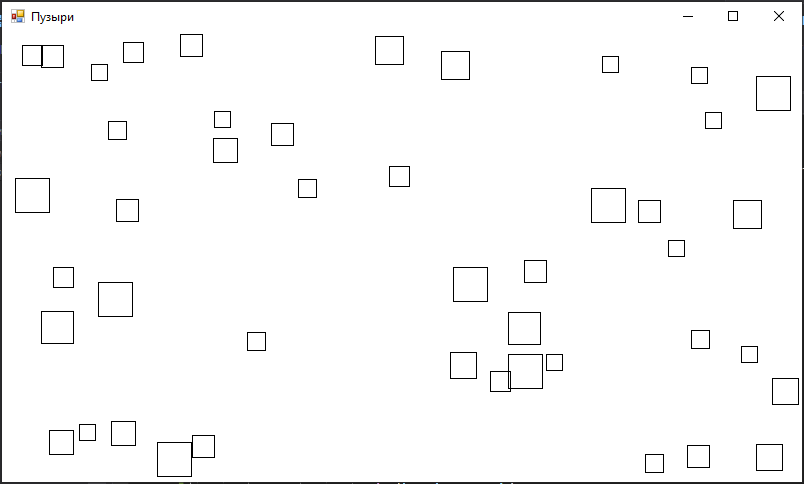
raster = new Bitmap(pictureBox1.Width, pictureBox1.Height);

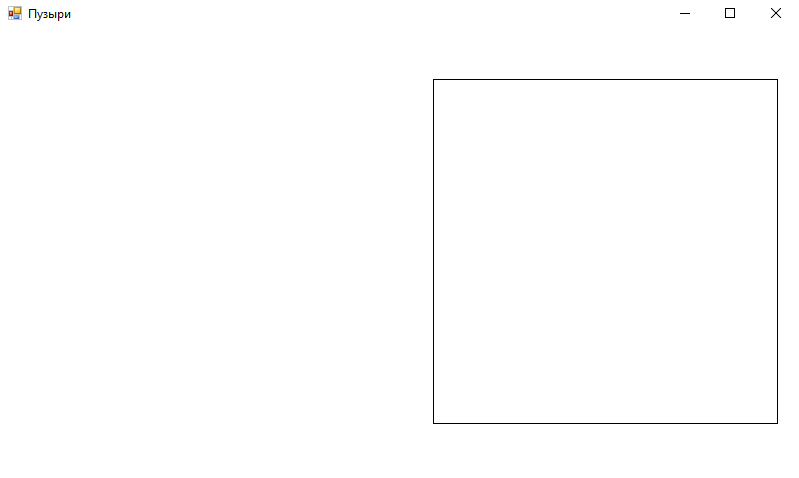
}

}

}

2)





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 \_9.\_2

{

public partial class Form1 : Form

{

Graphics graph;

Pen MyPen;

const int n = 50;

int Max\_x, Max\_y;

int[] r, dr, k;

Point[] center;

bool[] good, check;

Random rand;

bool stop = false;

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

{

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

{

r[i] = 5 + rand.Next(10);

dr[i] = 1;

center[i] = new Point((int)(r[i] + rand.Next(Max\_x - 2 \* r[i])), (int)(r[i] + rand.Next(Max\_y - 2 \* r[i])));

good[i] = true;

check[i] = false;

k[i] = 0;

}

timer1.Start();

}

private void Timer1\_Tick(object sender, EventArgs e)

{

graph.Clear(BackColor);

if (!stop)

{

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

{

if (good[i])

{

graph.DrawRectangle(MyPen, center[i].X - r[i], center[i].Y - r[i], 2 \* r[i], 2 \* r[i]);

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

{

if (i != j && good[j])

{

if (CheckTouch(i, j) && r[j] >= r[i])

{

System.Threading.Thread.Sleep(30);

good[i] = false;

break;

}

}

}

if (center[i].X - r[i] - dr[i] <= 0)

{

center[i].X += dr[i];

k[i] = 1;

check[i] = true;

}

if (center[i].Y - r[i] - dr[i] <= 0)

{

center[i].Y += dr[i];

k[i] = 2;

check[i] = true;

}

if (center[i].X + r[i] + dr[i] >= Max\_x)

{

center[i].X -= dr[i];

k[i] = -1;

check[i] = true;

}

if (center[i].Y + r[i] + dr[i] >= Max\_y)

{

center[i].Y -= dr[i];

k[i] = -2;

check[i] = true;

}

}

}

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

{

if (check[i])

{

switch (k[i])

{

case 1:

center[i].X += dr[i];

break;

case 2:

center[i].Y += dr[i];

break;

case -1:

center[i].X -= dr[i];

break;

case -2:

center[i].Y -= dr[i];

break;

}

}

}

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

{

if (good[i] && !check[i]) r[i] += dr[i];

if (2 \* r[i] > Max\_y) stop = true;

}

}

else timer1.Stop();

}

private bool CheckTouch(int i, int j)

{

if ((center[i].Y - r[i] < center[j].Y + 2 \* r[j]) && (center[i].Y - r[i] + 2 \* r[i] > center[j].Y - r[j]) && (center[i].X - r[i] + 2 \* r[i] > center[j].X - r[j]) && (center[i].X - r[i] < center[j].X - r[j] + 2 \* r[j])) return true;

else return false;

}

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

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

r = new int[n];

dr = new int[n];

check = new bool[n];

k = new int[n];

center = new Point[n];

good = new bool[n];

rand = new Random();

Max\_x = ClientSize.Width;

Max\_y = ClientSize.Height;

}

}

}