Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования «Национальный

исследовательский Нижегородский государственный

университет им. Н.И. Лобачевского»

**Институт информационных технологий, математики и механики**

Лабораторная работа №2 по курсу «Технология визуального программирования»:

«Нестандартные элементы управления:

создание программы «Ручка управления»»

Выполнил: студент группы 381808-1

Оганян Роберт Владимирович

Преподаватель: доцент кафедры ПИнж

Борисов Н. А.

Нижний Новгород

2019 г.

# Содержание

[Содержание 2](#_Toc4101358)

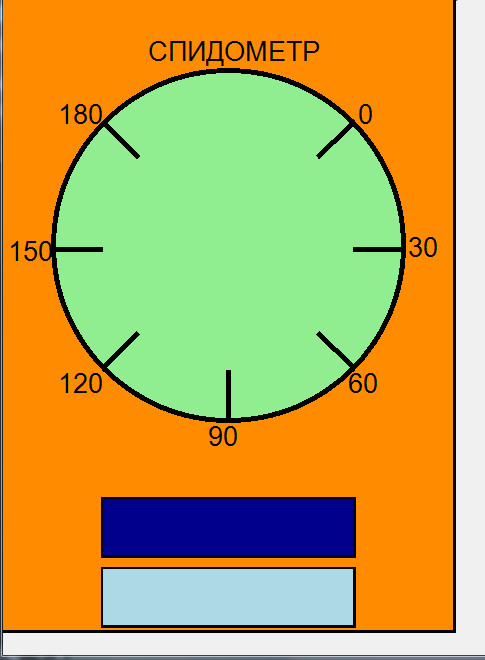
[Описание работы программы 3](#_Toc4101359)

[Приложение 10](#_Toc4101361)

## Описание работы программы

Программа выполнена в среде разработки microsoft visual studio 2017 с помощью Windows Form. Приложение написано для операционной системы Windows.

Начальный экран программы выглядит следующим образом:



Главный экран представляет из себя окно, содержащее:

* Шкалу спидометра
* Синяя кнопка для включения режима 1
* Голубая кнопка для включения режима 2

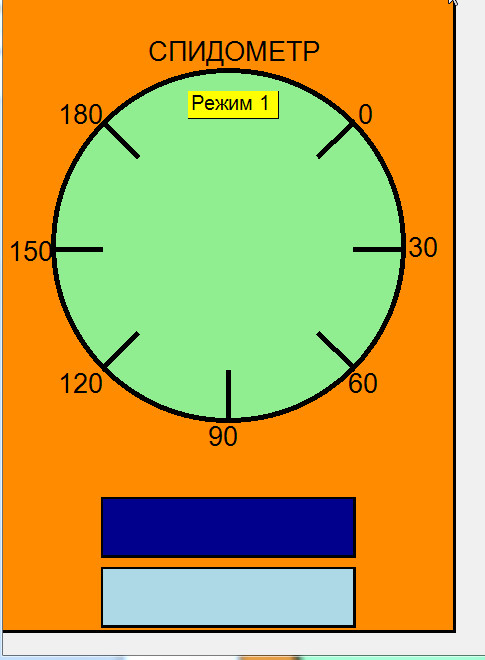
Режим 1:

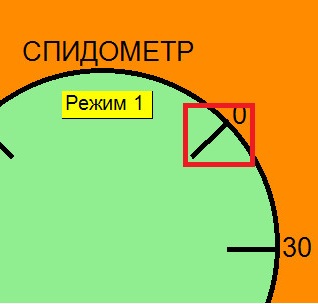
Дискретный режим для ступенчатого изменения положения стрелки. Стрелка рисуется строго только к делениям.

Режим 2:

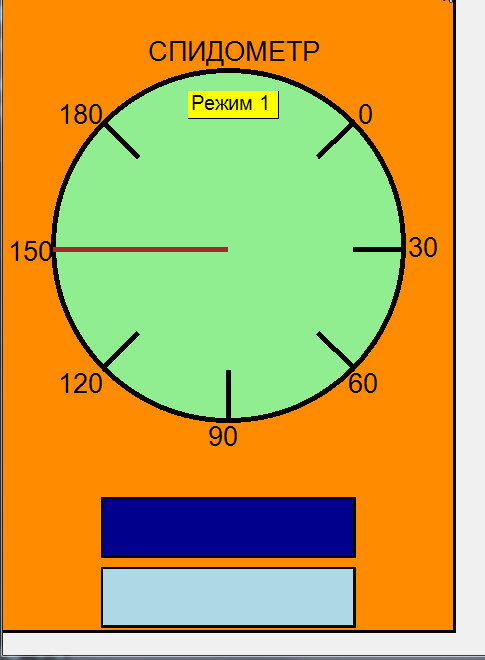
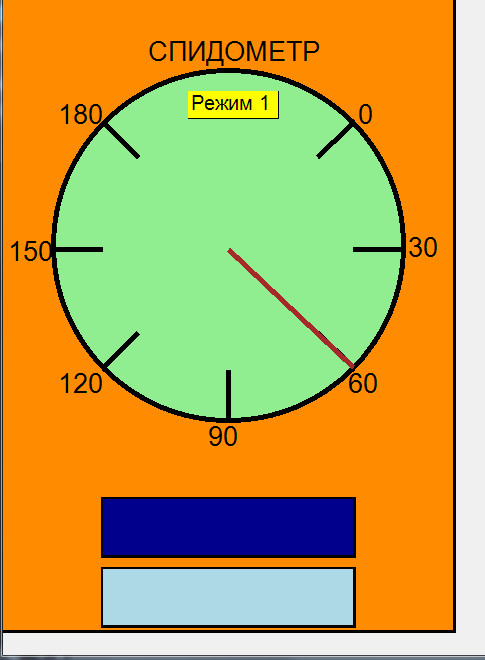
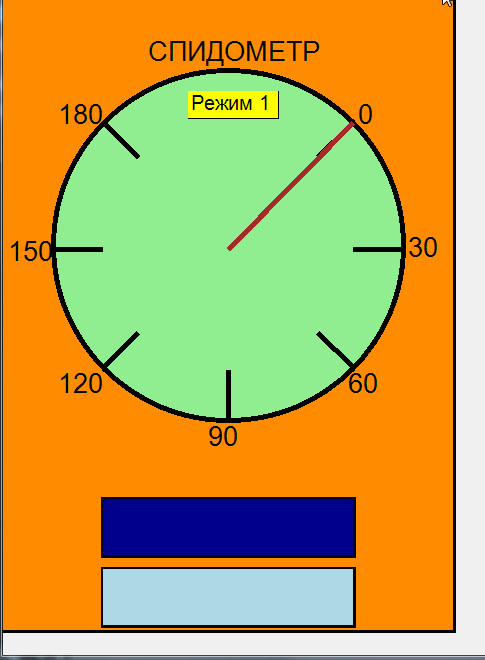
Непрерывный режим для плавного изменения положения стрелки. Стрелка рисуется в любом положении

Активируем режим 1.



Если кликнуть по примерно такой области около деления, 

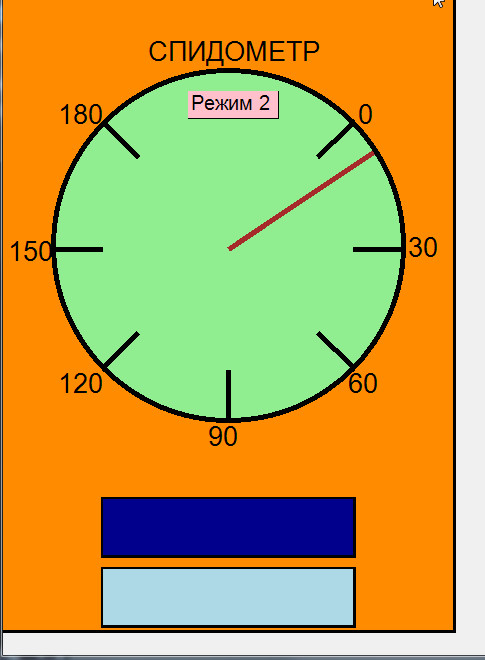
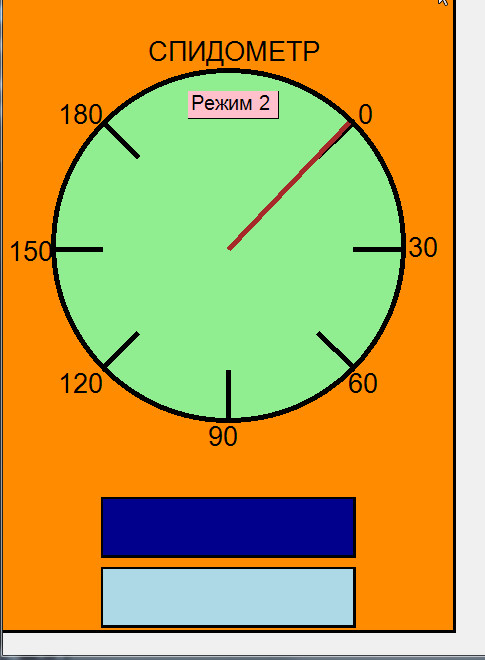
то нарисуется стрелка.

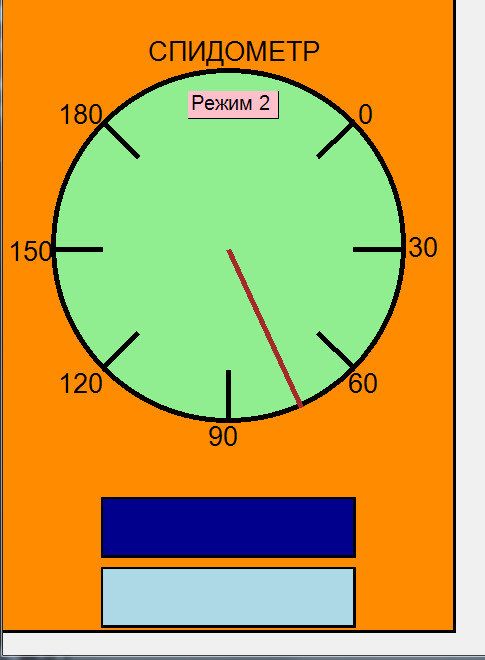
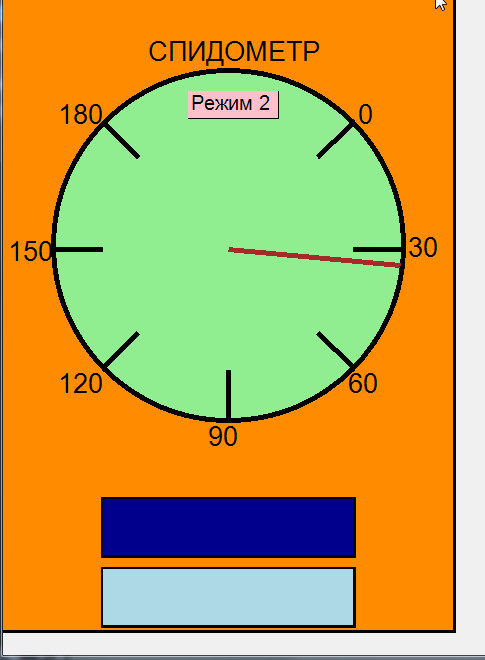


Теперь включим второй режим.

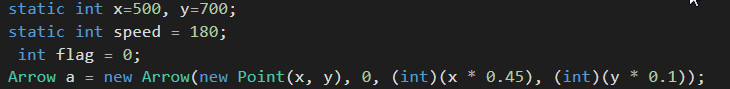
### C:\Users\OgRobertino2\Documents\Bandicam\bandicam 2019-03-21 22-36-22-649.jpg

Кликая вдоль границы окружности мы можем построить стрелку в любом месте.



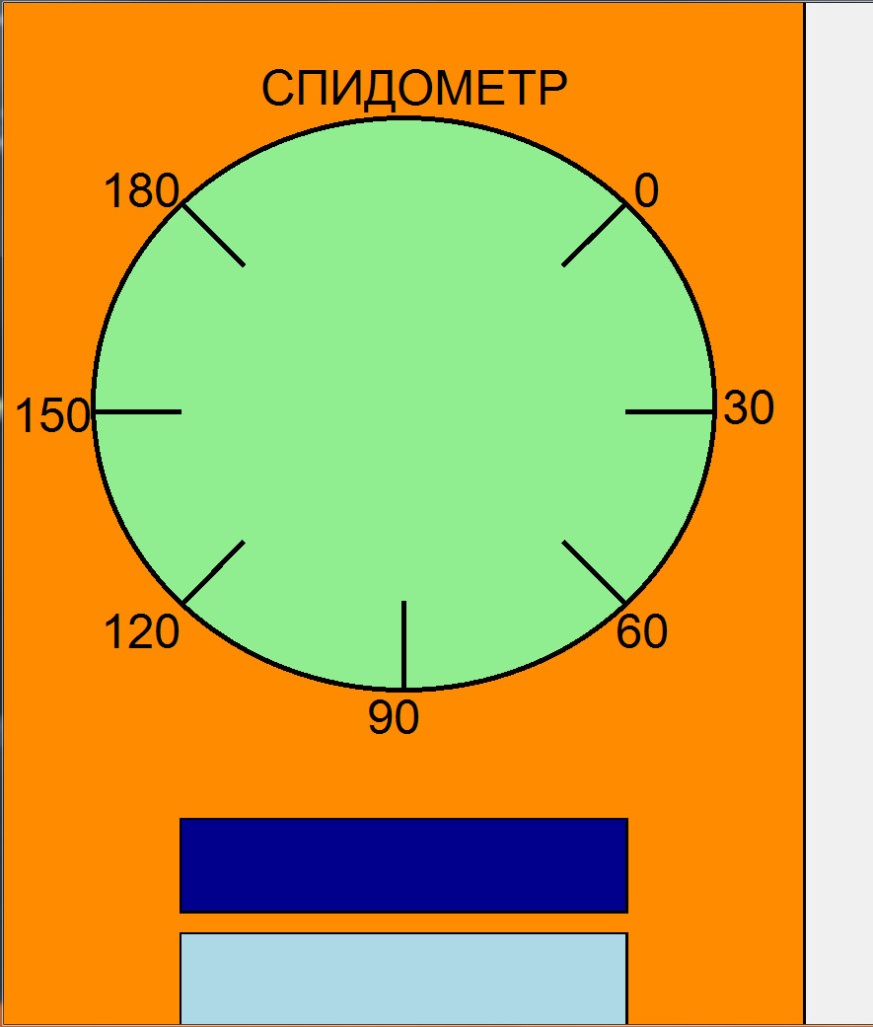
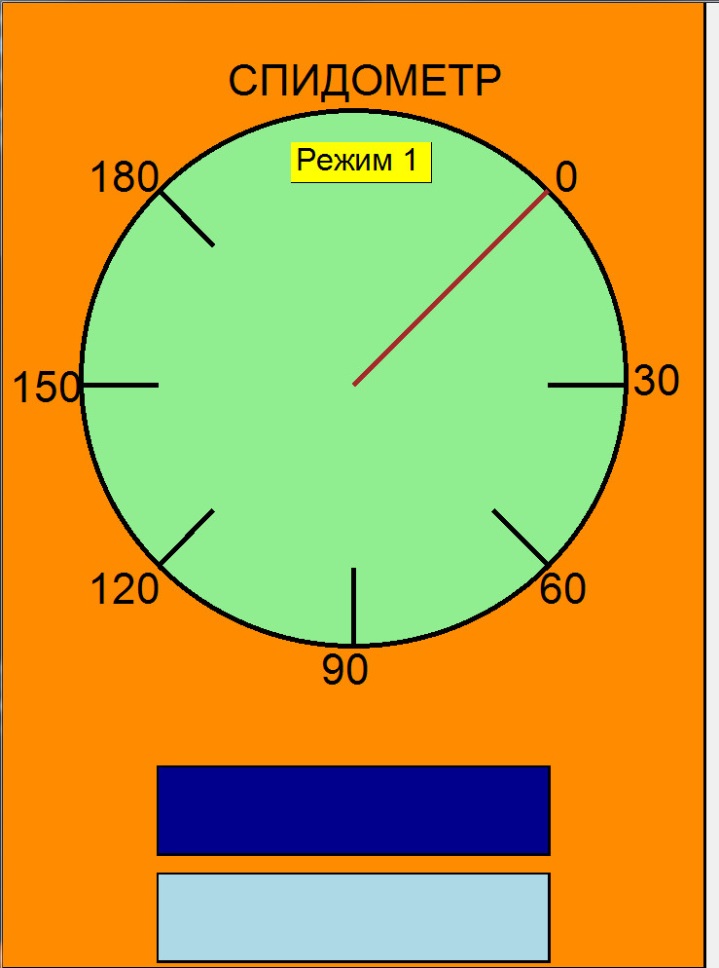


Программа гибкая, то есть при изменении масшатаба программа не ломается и выглядит нормально.

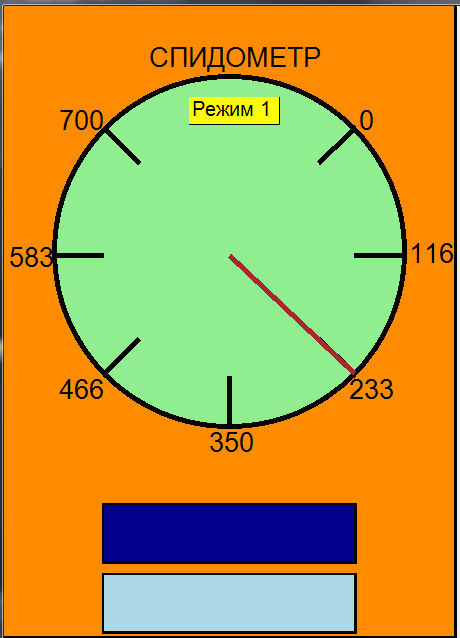


Также можно поменять значения на спидометре, изменив переменную speed.

Например, установим размеры (800,1100) и (900,1160).

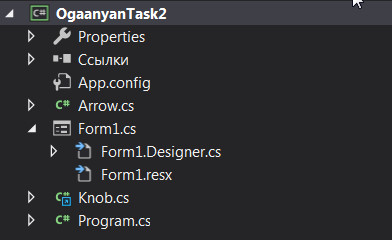


Теперь поменяем speed на 700.



### Приложение

Схема файла OganyanTask2



Файл Form1.cs:

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 OgaanyanTask2

{

public partial class Form1 : Form

{

static int x=800, y=1100;

static int speed = 180;

int flag = 0;

Arrow a = new Arrow(new Point(x, y), 0, (int)(x \* 0.45), (int)(y \* 0.1));

Knob c = new Knob(new Point(x,y), speed);

public Form1()

{

InitializeComponent();

this.Width = x;

this.Height = y;

}

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

{

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Graphics g = this.CreateGraphics();

c.Draw(g);

c.DrawNumbers(g);

}

int dx;

int dy;

private void Form1\_MouseClick(Object sender, MouseEventArgs e)

{

Graphics g = this.CreateGraphics();

if (flag == 2)

{ a.HelpArrowtwo(g, dx, dy); }

if (flag==1)

{ a.HelpArrowone(g, dx, dy, a.AreaReference(g, dx, dy)); }

if (c.isInsideblock1(e.X, e.Y))

{ c.rejimone(g); flag = 1; }

if (c.isInsideblock2(e.X, e.Y))

{ c.rejimtwo(g); flag = 2; }

if (flag == 1)

{a.DrawArrowone(g, a.AreaReference(g, e.X, e.Y)); dx = e.X;

dy = e.Y;

}

if (flag == 2 ) { a.DrawArrowtwo(g, e.X, e.Y); dx = e.X;

dy = e.Y; }

}

}

}

Файл Knob.cs

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 OgaanyanTask2

{

class Knob

{

private

int speed;

Point xy;

Pen p = new Pen(Color.Black, 5);

Pen p2 = new Pen(Color.Brown, 5);

Brush b = new SolidBrush(Color.DarkOrange);

Brush b2 = new SolidBrush(Color.LightGreen);

public Knob(Point xy, int speed)

{

this.xy = xy;

this.speed = speed;

}

internal bool isInsideblock1(int x, int y)

{

if ((x >= (int)(xy.X \* 0.2)) && (y >= (int)(0.714 \* xy.Y)) && (x <= (int)(xy.X \* 0.7)) && (y <= (int)(0.794 \* xy.Y)))

return true;

return false;

}

internal bool isInsideblock2(int x, int y)

{

if ((x >= (int)(xy.X \* 0.2)) && (y >= (int)(0.814 \* xy.Y)) && (x <= (int)(xy.X \* 0.7)) && (y <= (int)(0.894 \* xy.Y)))

return true;

return false;

}

internal void rejimone (Graphics g)

{

int line = (int)(0.1 \* xy.X);

g.DrawRectangle(Pens.Black, (int)(xy.X \* 0.37),

(int)(xy.Y \* 0.13), (int)(xy.X \* 0.18), (int)(xy.Y \* 0.11) - line);

g.FillRectangle(Brushes.Yellow, (int)(xy.X \* 0.37),

(int)(xy.Y \* 0.13), (int)(xy.X \* 0.18), (int)(xy.Y \* 0.11) - line);

g.DrawString("Режим 1", new Font("Arial", (int)(0.04 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.37),

(int)(xy.Y \* 0.13), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6) - line));

}

internal void rejimtwo(Graphics g)

{

int line = (int)(0.1 \* xy.X);

g.DrawRectangle(Pens.Black, (int)(xy.X \* 0.37),

(int)(xy.Y \* 0.13), (int)(xy.X \* 0.18), (int)(xy.Y \* 0.11) - line);

g.FillRectangle(Brushes.Pink, (int)(xy.X \* 0.37),

(int)(xy.Y \* 0.13), (int)(xy.X \* 0.18), (int)(xy.Y \* 0.11) - line);

g.DrawString("Режим 2", new Font("Arial", (int)(0.04 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.37),

(int)(xy.Y \* 0.13), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6) - line));

}

internal void Draw(Graphics g)

{

int line = (int)(0.1 \* xy.X);

g.DrawRectangle(p, 0, 0, (int)(0.9 \* xy.X), (int)(0.9 \* xy.Y)); //Основая рамка

//окружность в центре

g.DrawEllipse(p, (int)(0.1 \* xy.X), (int)(0.1 \* xy.Y), (int)(0.7 \* xy.X), (int)(0.5 \* xy.Y));

g.FillRectangle(b, 0, 0, (int)(0.9 \* xy.X), (int)(0.9 \* xy.Y));

g.FillEllipse(b2, (int)(0.1 \* xy.X), (int)(0.1 \* xy.Y), (int)(0.7 \* xy.X), (int)(0.5 \* xy.Y));

g.DrawEllipse(p, (int)(0.1 \* xy.X), (int)(0.1 \* xy.Y), (int)(0.7 \* xy.X), (int)(0.5 \* xy.Y));

//деления

g.DrawLine(p, (int)(xy.X \* 0.1), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.1) + line, (int)(xy.Y \* 0.357));

g.DrawLine(p, (int)(xy.X \* 0.8), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.8) - line, (int)(xy.Y \* 0.357));

g.DrawLine(p, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6) - line);

g.DrawLine(p, (int)(xy.X \* 0.2), (int)(xy.Y \* 0.175), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.7), (int)(xy.Y \* 0.175), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.2), (int)(xy.Y \* 0.525), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.7), (int)(xy.Y \* 0.525), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398)));

Rectangle block1 = new Rectangle((int)(xy.X \* 0.2), (int)(0.714 \* xy.Y), (int)(xy.X \* 0.5), (int)(0.08 \* xy.Y));

Rectangle block2 = new Rectangle((int)(xy.X \* 0.2), (int)(0.814 \* xy.Y), (int)(xy.X \* 0.5), (int)(0.08 \* xy.Y));

g.DrawRectangle(p, block1);

g.FillRectangle(Brushes.DarkBlue, block1);

g.DrawRectangle(p, block2);

g.FillRectangle(Brushes.LightBlue, block2);

}

internal void DrawNumbers(Graphics g)

{

int line = (int)(0.1 \* xy.X);

g.DrawString(0 + "", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.7),

(int)(xy.Y \* 0.14), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)),

(int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398))));

g.DrawString(speed + "", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.1),

(int)(xy.Y \* 0.14), (int)(xy.X \* 0.35 - line \* Math.Cos(0.785398)),

(int)(xy.Y \* 0.17 + line \* Math.Sin(0.785398))));

g.DrawString(speed / 2 + "", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.4),

(int)(xy.Y \* 0.6), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6) - line));

g.DrawString(speed \* 5 / 6 + "", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(0),

(int)(xy.Y \* 0.337), (int)(xy.X \* 0.1) + line, (int)(xy.Y \* 0.357)));

g.DrawString(speed \* 1 / 6 + "", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.8),

(int)(xy.Y \* 0.33), (int)(xy.X \* 0.1) + line, (int)(xy.Y \* 0.357)));

g.DrawString(speed \* 4 / 6 + "", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.1),

(int)(xy.Y \* 0.525), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)),

(int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398))));

g.DrawString(speed \* 2 / 6 + "", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.68),

(int)(xy.Y \* 0.525), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)),

(int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398))));

g.DrawString("СПИДОМЕТР", new Font("Arial", (int)(0.055 \* xy.X),

GraphicsUnit.Pixel), Brushes.Black, new RectangleF((int)(xy.X \* 0.28),

(int)(xy.Y \* 0.05), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6) - line));

}

}

}

Файл Arrow.cs

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 OgaanyanTask2

{

class Arrow

{

List<int> numbers = new List<int>();

double deg;

Point xy;

int xx, yy;

Pen p = new Pen(Color.Black, 5);

Pen p2 = new Pen(Color.Brown, 5);

Brush b = new SolidBrush(Color.DarkOrange);

Pen kostil= new Pen(Color.LightGreen,5);

public Arrow(Point xy, double deg, int xx, int yy)

{

this.xy = xy;

this.deg = deg;

this.xx = xx;

this.yy = yy;

}

internal int AreaReference(Graphics g, int xx, int yy)

{

if (xx >= (int)(0.62\*xy.X) && xx <= (int)(0.74\*xy.X) && yy >= (int)(0.157\*xy.Y) && yy <= (int)(0.228\*xy.Y))

return 1;

else if (xx >= (int)(0.7 \* xy.X) && xx <= (int)(0.82 \* xy.X) && yy >= (int)(0.314 \* xy.Y) && yy <= (int)(0.392 \* xy.Y))

return 2;

else if (xx >= (int)(0.62 \* xy.X) && xx <= (int)(0.74 \* xy.X) && yy >= (int)(0.465 \* xy.Y) && yy <= (int)(0.542 \* xy.Y))

return 3;

else if (xx >= (int)(0.39 \* xy.X) && xx <= (int)(0.51 \* xy.X) && yy >= (int)(0.364 \* xy.Y) && yy <= (int)(0.444 \* xy.Y))

return 4;

else if (xx >= (int)(0.17 \* xy.X) && xx <= (int)(0.29 \* xy.X) && yy >= (int)(0.465 \* xy.Y) && yy <= (int)(0.542 \* xy.Y))

return 5;

else if (xx >= (int)(0.09 \* xy.X) && xx <= (int)(0.21 \* xy.X) && yy >= (int)(0.314 \* xy.Y) && yy <= (int)(0.392 \* xy.Y))

return 6;

else if (xx >= (int)(0.17 \* xy.X) && xx <= (int)(0.29 \* xy.X) && yy >= (int)(0.157 \* xy.Y) && yy <= (int)(0.228 \* xy.Y))

return 7;

return 0;

}

internal void DrawArrowone(Graphics g, int a)

{

switch(a)

{

case 1:

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.7), (int)(xy.Y \* 0.175));

break;

case 2:

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.8), (int)(xy.Y \* 0.357));

break;

case 3:

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.7), (int)(xy.Y \* 0.525));

break;

case 4:

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6));

break;

case 5:

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.2), (int)(xy.Y \* 0.525));

break;

case 6:

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.1), (int)(xy.Y \* 0.357));

break;

case 7:

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.2), (int)(xy.Y \* 0.175));

break;

default:

break;

}

}

internal void HelpArrowone(Graphics g, int xx, int yy,int a)

{

int line = (int)(0.1 \* xy.X);

switch (a)

{

case 1:

// g.DrawLine(p, (int)(xy.X \* 0.7), (int)(xy.Y \* 0.175), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398)));

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.7), (int)(xy.Y \* 0.175));

break;

case 2:

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.8), (int)(xy.Y \* 0.357));

break;

case 3:

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.7), (int)(xy.Y \* 0.525));

break;

case 4:

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6));

break;

case 5:

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.2), (int)(xy.Y \* 0.525));

break;

case 6:

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.1), (int)(xy.Y \* 0.357));

break;

case 7:

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.2), (int)(xy.Y \* 0.175));

break;

default:

break;

}

g.DrawEllipse(p, (int)(0.1 \* xy.X), (int)(0.1 \* xy.Y), (int)(0.7 \* xy.X), (int)(0.5 \* xy.Y));

g.DrawLine(p, (int)(xy.X \* 0.1), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.1) + line, (int)(xy.Y \* 0.357));

g.DrawLine(p, (int)(xy.X \* 0.8), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.8) - line, (int)(xy.Y \* 0.357));

g.DrawLine(p, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6) - line);

g.DrawLine(p, (int)(xy.X \* 0.2), (int)(xy.Y \* 0.175), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.7), (int)(xy.Y \* 0.175), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.2), (int)(xy.Y \* 0.525), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.7), (int)(xy.Y \* 0.525), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398)));

}

internal void HelpArrowtwo(Graphics g, int xx, int yy)

{

int line = (int)(0.1 \* xy.X);

g.DrawEllipse(p, (int)(0.1 \* xy.X), (int)(0.1 \* xy.Y), (int)(0.7 \* xy.X), (int)(0.5 \* xy.Y));

int r = (int)(0.25 \* xy.Y);

double errormax = Math.Pow((xx - (int)xy.X \* 0.45), 2) + Math.Pow((yy - (int)xy.Y \* 0.357), 2);

if (errormax <= r \* r \* 1.02 && errormax >= 0.95 \* r \* r)

g.DrawLine(kostil, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), xx, yy);

g.DrawLine(p, (int)(xy.X \* 0.1), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.1) + line, (int)(xy.Y \* 0.357));

g.DrawLine(p, (int)(xy.X \* 0.8), (int)(xy.Y \* 0.357), (int)(xy.X \* 0.8) - line, (int)(xy.Y \* 0.357));

g.DrawLine(p, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6), (int)(xy.X \* 0.45), (int)(xy.Y \* 0.6) - line);

g.DrawLine(p, (int)(xy.X \* 0.2), (int)(xy.Y \* 0.175), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.7), (int)(xy.Y \* 0.175), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.175 + line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.2), (int)(xy.Y \* 0.525), (int)(xy.X \* 0.2 + line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398)));

g.DrawLine(p, (int)(xy.X \* 0.7), (int)(xy.Y \* 0.525), (int)(xy.X \* 0.7 - line \* Math.Cos(0.785398)), (int)(xy.Y \* 0.525 - line \* Math.Sin(0.785398)));

}

internal void DrawArrowtwo(Graphics g,int xx,int yy)

{

int d1 = xx;

int d2 = yy;

int r = (int)(0.25 \* xy.Y);

double errormax = Math.Pow((xx - (int)xy.X \* 0.45), 2) + Math.Pow((yy - (int)xy.Y \* 0.357), 2);

if (errormax<=r\*r\*1.02 && errormax>=0.95\*r\*r)

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), xx, yy);

}

internal void DrawArrow(Graphics g)

{

int r = (int)(0.25 \* xy.Y);

int line = (int)(0.1 \* xy.X);

double coss = Math.Cos(deg);

double sinn = Math.Sin(deg);

g.DrawLine(p2, (int)(xy.X \* 0.45), (int)(xy.Y \* 0.357), (int)((xy.X \* 0.45)-(sinn\*r)), (int)((xy.Y \* 0.357)-(coss\*r)));

}

}

}