**29 АНИМАЦИЯ**

Задание 1. Работа с таймером. Создать анимацию секундной стрелки.

Листинг программы:

public partial class Form1 : Form

{

private int x1, y1, x2, y2, R;

private double a;

private Pen pen = new Pen(Color.LightSkyBlue, 2);

public Form1()

{

InitializeComponent();

}

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

{

Graphics grap = e.Graphics;

grap.DrawLine(pen, x1, y1, x2, y2);

}

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

{

R = 150;

a = 0;

x1 = ClientSize.Width / 2;

y1 = ClientSize.Height / 2;

x2 = x1 + (int)(R \* Math.Cos(a));

y2 = y1 - (int)(R \* Math.Sin(a));

}

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

{

a -= 0.1;

x2 = x1 + (int)(R \* Math.Cos(a));

y2 = y1 - (int)(R \* Math.Sin(a));

Invalidate();

} }

Таблица 29.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 150, 0, x1, x2, y1, y2 | Animation |

Источник: собственная разработка

Анализ результатов:

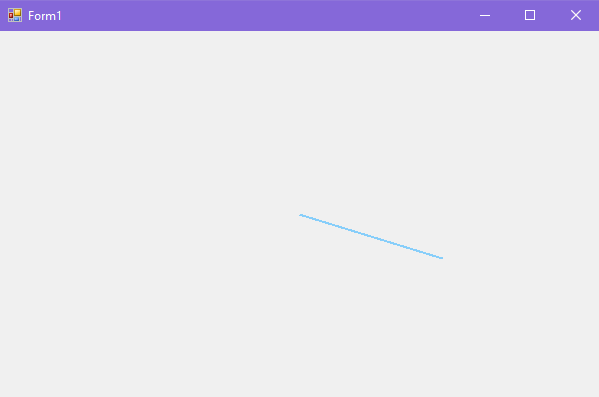


Рисунок 29.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Создать анимацию бегущей строки.

Листинг программы:

public partial class Form1 : Form

{

Graphics grap;

Bitmap baner;

Rectangle rct;

public Form1()

{

InitializeComponent();

try

{

baner = new Bitmap("banner.jpg");

}

catch (Exception e)

{

MessageBox.Show("Ошибка загрузки файла баннера\n" + e.ToString(), "Баннер");

this.Close();

return;

}

rct.X = 0;

rct.Y = 0;

rct.Width = baner.Width;

rct.Height = baner.Height;

timer1.Interval = 50;

timer1.Enabled = true;

}

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

{

rct.X -= 1;

if (Math.Abs(rct.X) > rct.Width)

rct.X += rct.Width;

Invalidate();

}

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

if ((e.Y < rct.Y + rct.Height) && (e.Y > rct.Y))

{

if (timer1.Enabled != false)

timer1.Enabled = false;

}

else

{

if (timer1.Enabled != true)

timer1.Enabled = true;

}

}

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

{

grap = e.Graphics;

for (int i = 0; i < Convert.ToInt16(this.ClientSize.Width / rct.Width) + 1; i++)

grap.DrawImage(baner, rct.X + i \* rct.Width, rct.Y);

}

}

Таблица 29.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 0, 0, 50 | Animation |

Источник: собственная разработка

Анализ результатов:



Рисунок 29.2 – Результат работы программы

Источник: собственная разработка