

29 Анимация

Задание 1. Разработайте программу Clock моделирующую движение секундной стрелки.

Листинг программы:

```
namespace Task3
{
    public partial class Form1 : Form
    {
        private List<Point> points = new List<Point>(); // Список точек для
отображения пути
        private Random random = new Random(); // Генератор случайных чисел
для цветов
        private int x1, y1, x2, y2;
        private double a, t, fi;
        private Pen pen = new Pen(Color.DarkRed, 2);

        public Form1()
        {
            InitializeComponent();
            this.Load += Form1_Load; // Подключаем обработчик события Load
            this.Paint += Form1_Paint; // Подключаем обработчик события Paint

            timer1.Interval = 10; // Устанавливаем интервал времени (в
миллисекундах)
            timer1.Tick += timer1_Tick; // Подключаем обработчик события Tick
            timer1.Start(); // Запускаем таймер

            this.DoubleBuffered = true;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            x1 = ClientSize.Width / 2;
            y1 = ClientSize.Height / 2;
            a = 150;
            fi = -0.5;
            t = Math.Sin(fi);
            x2 = x1 + (int)((3 * a * t) / (1 + t * t * t));
        }
    }
}
```

					УП 2-40 01 01.37ТП.227.23.29		
Изм.	Лист	№ док	Подпись	Дата			
Разраб.		Сорокина Е.А.			Анимация	Лит	Лист
Проверил.		Новик А.И.					127
						Гродненский ГКТТид	
Н.контр.							
Утвердил.							

```

        y2 = y1 - (int)((3 * a * t * t) / (1 + t * t * t));

        this.BackColor = Color.White; // Установка цвета фона формы
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = e.Graphics;

        // Рисуем линии, соединяющие точки в списке points
        for (int i = 1; i < points.Count; i++)
        {
            Color lineColor = Color.FromArgb(random.Next(256),
random.Next(256), random.Next(256));
            Pen linePen = new Pen(lineColor, 2);
            g.DrawLine(linePen, points[i - 1], points[i]);
        }

        // Рисуем текущую позицию в виде круга
        g.DrawEllipse(pen, x2, y2, 20, 20);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        fi += 0.05; // Увеличение шага изменения угла
        t = Math.Tan(fi);
        x2 = x1 + (int)((3 * a * t) / (1 + t * t * t));
        y2 = y1 - (int)((3 * a * t * t) / (1 + t * t * t));

        // Добавляем текущую позицию в список points
        points.Add(new Point(x2 + 10, y2 + 10));

        // Ограничиваем количество точек, чтобы не рисовать слишком
длинный путь
        if (points.Count > 100)
        {
            points.RemoveAt(0);
        }

        Invalidate();
    }
}

```

Анализ результатов:

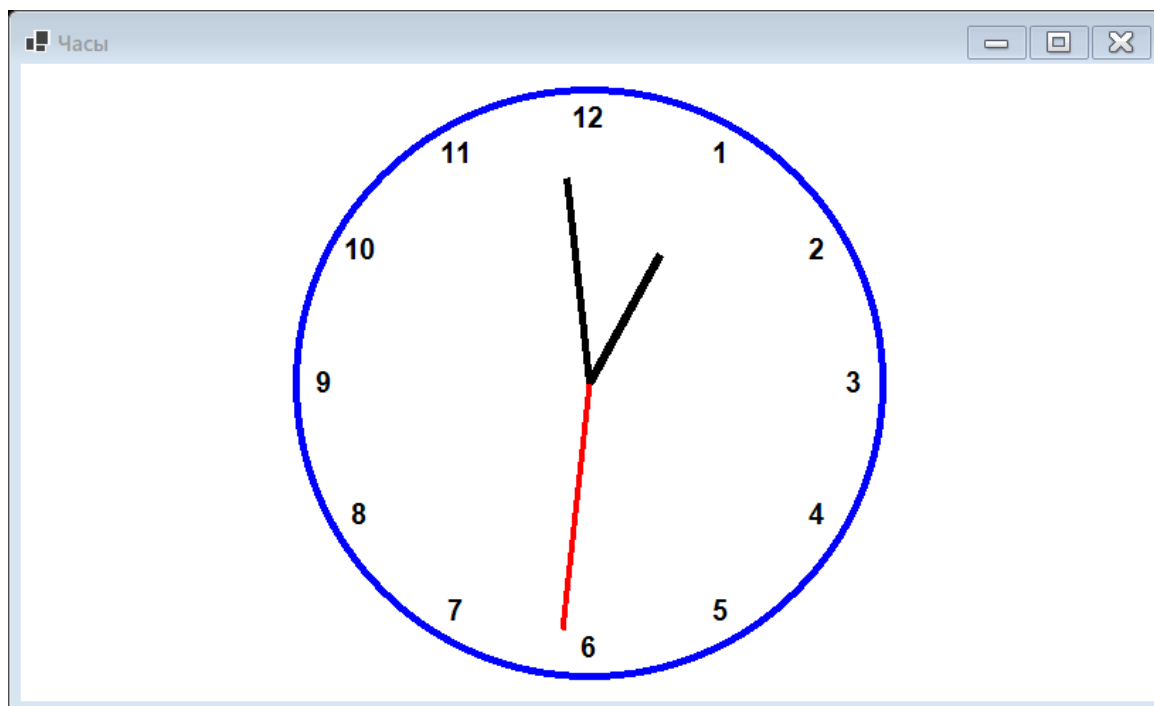


Рисунок 29.1 – Результат работы программы

Задание 2. Движение по траектории.

Листинг программы:

```
namespace Task3
{
    public partial class Form1 : Form
    {
        private List<Point> points = new List<Point>(); // Список точек для
отображения пути
        private Random random = new Random(); // Генератор случайных чисел
для цветов
        private int x1, y1, x2, y2;
        private double a, t, fi;
        private Pen pen = new Pen(Color.DarkRed, 2);

        public Form1()
        {
            InitializeComponent();
            this.Load += Form1_Load; // Подключаем обработчик события Load
            this.Paint += Form1_Paint; // Подключаем обработчик события Paint

            timer1.Interval = 10; // Устанавливаем интервал времени (в
миллисекундах)
            timer1.Tick += timer1_Tick; // Подключаем обработчик события Tick
            timer1.Start(); // Запускаем таймер
        }
    }
}
```

Изм.	Лист	№ док	Подпись	Дата

```

        this.DoubleBuffered = true;
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        x1 = ClientSize.Width / 2;
        y1 = ClientSize.Height / 2;
        a = 150;
        fi = -0.5;
        t = Math.Sin(fi);
        x2 = x1 + (int)((3 * a * t) / (1 + t * t * t));
        y2 = y1 - (int)((3 * a * t * t) / (1 + t * t * t));

        this.BackColor = Color.White; // Установка цвета фона формы
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = e.Graphics;

        // Рисуем линии, соединяющие точки в списке points
        for (int i = 1; i < points.Count; i++)
        {
            Color lineColor = Color.FromArgb(random.Next(256),
random.Next(256), random.Next(256));
            Pen linePen = new Pen(lineColor, 2);
            g.DrawLine(linePen, points[i - 1], points[i]);
        }

        // Рисуем текущую позицию в виде круга
        g.DrawEllipse(pen, x2, y2, 20, 20);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        fi += 0.05; // Увеличение шага изменения угла
        t = Math.Tan(fi);
        x2 = x1 + (int)((3 * a * t) / (1 + t * t * t));
        y2 = y1 - (int)((3 * a * t * t) / (1 + t * t * t));

        // Добавляем текущую позицию в список points
        points.Add(new Point(x2 + 10, y2 + 10));
    }

```

```

        // Ограничиваем количество точек, чтобы не рисовать слишком
        // длинный путь
        if (points.Count > 100)
        {
            points.RemoveAt(0);
        }

        Invalidate();
    }
}

```

Анализ результатов:

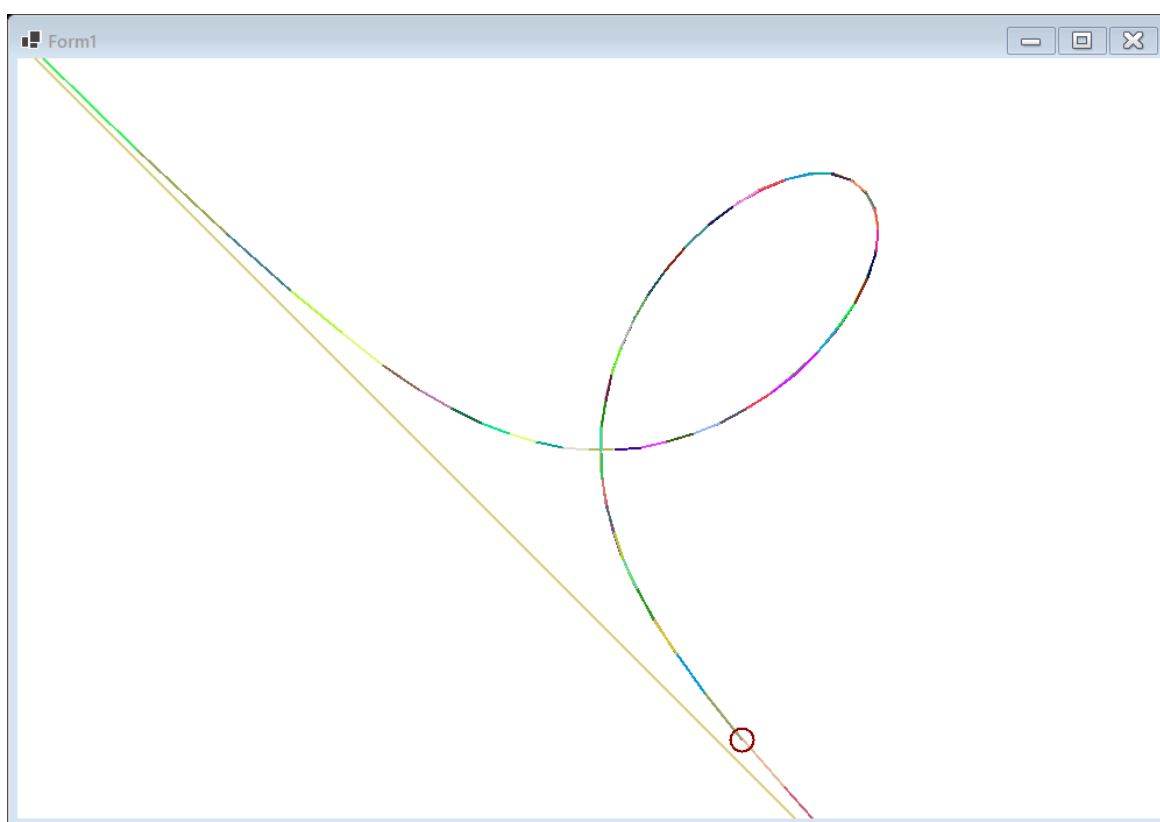


Рисунок 29.2 – Результат работы программы

Изм.	Лист	№ док	Подпись	Дата

УП 2-40 01 01.37ТП.227.23.29

Лист
131