Лабораторна робота № 21

Тема. Функціональні можливості класу Form. Створення векторних графічних зображень об’єктів.

Мета. Формування вмінь і навиків створення та корегування статичних і динамічних графічних зображень засобами C#. Закріплення вмінь і навиків використання класів, об'єктів, підпрограм, елементів керування. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

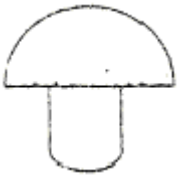
Контрольні запитання.

1. Які геометричні фігури можливо створити у формах засобами C#?
2. Де знаходиться початок координат при формуванні графічних зображень?
3. Які одиниці вимірювання використовуються при формуванні графічних зображень?
4. Які параметри процедур для створення ліній та еліпсів?
5. Як та коли слід встановлювати кольори ліній та фону?

Завдання.

1. Відкрийте розроблений раніше додаток-форму з описом базового та породжених класів згідно варіанту. Доповніть породжені класи методами *draw3D()* для побудови їх тривимірних зображень. Якщо зобразити дані породжених класів не вдається, то створіть для них одинарні (3 бали до рейтингу) або повторювані (5 балів до рейтингу) фігури по варіантах.

Варіанти одинарних фігур:

1).  Гриб

 Прапорець

2).  Світильник

 Цифра

3).  Келих

 Трикутник

4).  Парасолька

 Молоток

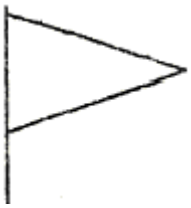
5).  Пісочний годинник

 Буква

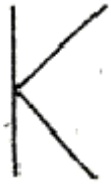
6).  Морозиво

 Цифра

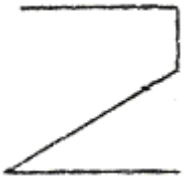
7).  Рогач

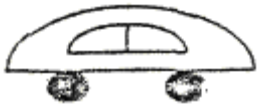
 Прапорець

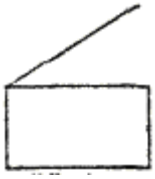
8).  Буква

 Буква

9).  Буква

 Цифра

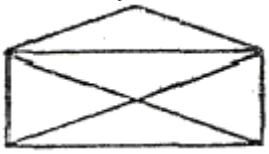
10).  Автомобіль

 Цифра

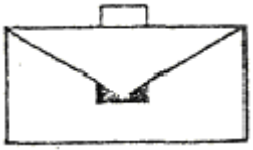
11).  Колобок

 Буква

12).  Буква

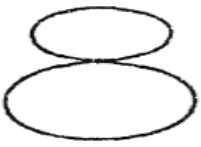
 Конверт

13).  Буква

 Портфель

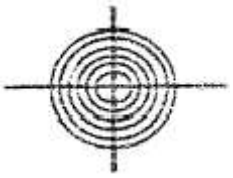
14).  Знак оклику

 Цеглина

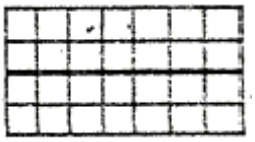
15). 
Цифра

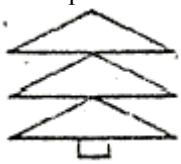

Дорожній знак

Варіанти повторюваних фігур (формуються з використанням оператора циклу):

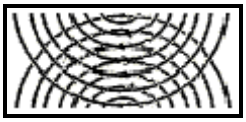
1). 
Мішень

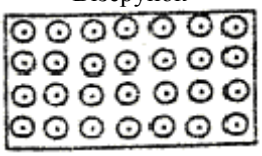
2). 
Доміно

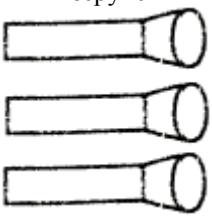
3). 
Гратка

4). 
Ялинка

5). 
Веселка

6). 
Візерунок

7). 
Візерунок

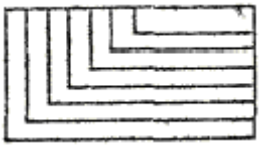
8). 
Ліхтарики

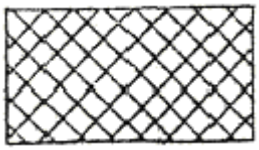
9). 
Будинок

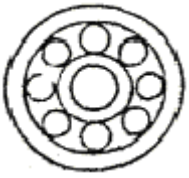
10). 
Свічки

11). 
Вкладені прямокутники

12). 
Світлофор

13). 
Вкладені прямокутники

14). 
Візерунок

15). 
Підшипник

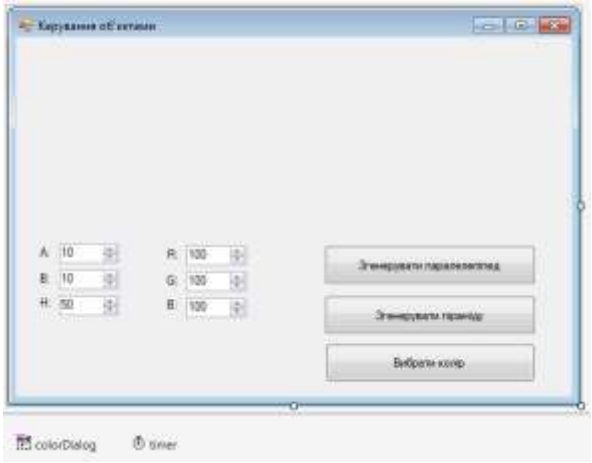
2. Забезпечте створення та переміщення чи інші візуальні ефекти зображень породжених об’єктів.

Наприклад, об’єкти прямокутного паралелепіпеда і піраміди з основою-прямокутником доцільно породити від класу прямокутника. Опис батьківського та породжених класів можуть виглядати так:

```
class Rectangle
{   private double a, b;

    public double A
    { get { return a; }
      set { if (value >= 0) a = value; }}

    public double B
    { get { return b; }
      set { if (value >= 0) b = value; }}

    public Rectangle(double a, double b)
    { this.a = a; this.b = b; }

    public double area()
    { return a * b; }

    public double perimeter()
    { return 2 * (a + b); }

    public virtual void Info()
    { MessageBox.Show("Дані прямокутника:\nдві сторони по " + a.ToString() + " та дві по " + b.ToString() +
                      " од.;\nплоща: " + area().ToString() + " кв. од.;\nпериметр: " + perimeter().ToString() + " од.;",
                      "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information); }
}

class Parallelepiped : Rectangle
{   private byte cr, cg, cb;
    private double h;

    public byte CR
    { get { return cr; }
      set { if (value >= 0 && value <= 255) cr = value; }}

    public byte CG
    { get { return cg; }
      set { if (value >= 0 && value <= 255) cg = value; }}

    public byte CB
    { get { return cb; }
      set { if (value >= 0 && value <= 255) cb = value; }}

    public double H
    {get { return h; }
     set { if (value >= 0) h = value; }}

    public Parallelepiped(double a, double b, double h, byte cr, byte cg, byte cb) :
                          base(a, b)
    { H = h; CR = cr; CG = cg; CB = cb; }

    public void draw3D(int startX, int startY, PaintEventArgs e)
    {   Pen pen = new Pen(Color.FromArgb(255, CR, CG, CB), 3);
        int x = (int)A, y = (int)B, z = (int)H;
        e.Graphics.DrawLine(pen, startX, startY, startX + x, startY);
        e.Graphics.DrawLine(pen, startX, startY, startX + y, startY + y);
        e.Graphics.DrawLine(pen, startX + y, startY + y, startX + y + x, startY + y);
        e.Graphics.DrawLine(pen, startX + x, startY, startX + y + x, startY + y);

        e.Graphics.DrawLine(pen, startX, startY, startX, startY + z);
        e.Graphics.DrawLine(pen, startX + x, startY, startX + x, startY + z);
        e.Graphics.DrawLine(pen, startX + y, startY + y, startX + y, startY + y + z);
        e.Graphics.DrawLine(pen, startX + x + y, startY + y, startX + x + y, startY + y + z);

        e.Graphics.DrawLine(pen, startX, startY + z, startX + x, startY + z);
        e.Graphics.DrawLine(pen, startX, startY + z, startX + y, startY + y + z);
        e.Graphics.DrawLine(pen, startX + y, startY + y + z, startX + y + x, startY + y + z);
        e.Graphics.DrawLine(pen, startX + x, startY + z, startX + y + x, startY + y + z);
    }

    new public double area()
    { return 2 * (A * B + A * H + B * H); }

    new public double perimeter()
    { return 4 * (A + B + H); }

    public double volume()
    { return A * B * H; }

    public override void Info()
    { MessageBox.Show("Дані паралелепіпеда:\nсторони по " + A.ToString() + ", " + B.ToString() + ", " + H.ToString() +
                      " од.;\nплоща: " + area().ToString() + " кв. од.\nоб'єм: " + volume().ToString() + " куб. од.",
                      "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

}
```

```
class Piramida : Rectangle
{private byte cr, cg, cb;
    private double h;

    public byte CR
    { get { return cr; }
      set { if (value >= 0 && value <= 255) cr = value; }}

    public byte CG
    { get { return cg; }
      set { if (value >= 0 && value <= 255) cg = value; }}

    public byte CB
    { get { return cb; }
      set { if (value >= 0 && value <= 255) cb = value; }}

    public double H
    {get { return h; }
     set { if (value >= 0) h = value; }}

    public Piramida(double a, double b, double h, byte cr, byte cg, byte cb) :
                    base(a, b)
    { H = h; CR = cr; CG = cg; CB = cb; }

    public void draw3D(int startX, int startY, PaintEventArgs e)
    {   Pen pen = new Pen(Color.FromArgb(255, CR, CG, CB), 3);
        int x = (int)A, y = (int)B, z = (int)H, vx = startX + (x + y) / 2, vy = startY + y / 2;
        e.Graphics.DrawLine(pen, vx, vy, startX, startY + z);
        e.Graphics.DrawLine(pen, vx, vy, startX + y, startY + y + z);
        e.Graphics.DrawLine(pen, vx, vy, startX + x, startY + z);
        e.Graphics.DrawLine(pen, vx, vy, startX + x + y, startY + y + z);

        e.Graphics.DrawLine(pen, startX, startY + z, startX + x, startY + z);
        e.Graphics.DrawLine(pen, startX, startY + z, startX + y, startY + y + z);
        e.Graphics.DrawLine(pen, startX + y, startY + y + z, startX + y + x, startY + y + z);
        e.Graphics.DrawLine(pen, startX + x, startY + z, startX + y + x, startY + y + z);
    }

    public double volume()
    {return 1.0 / 3 * A * B * H; }

    public override void Info()
    {MessageBox.Show("Дані піраміди:\nсторони основи по " + A.ToString() + " та " + B.ToString()+
      " од., висота "+H.ToString()+" од.;\nплоща основи: "+area().ToString()+" кв. од.\nоб'єм: "+
      volume().ToString()+" куб. од.","Інформація",MessageBoxButtons.OK,MessageBoxIcon.Information); }
}
```

Тоді головна кнопкова форма з додатковими елементами керування, нумераторами і кнопками для створення об’єктів буде така:

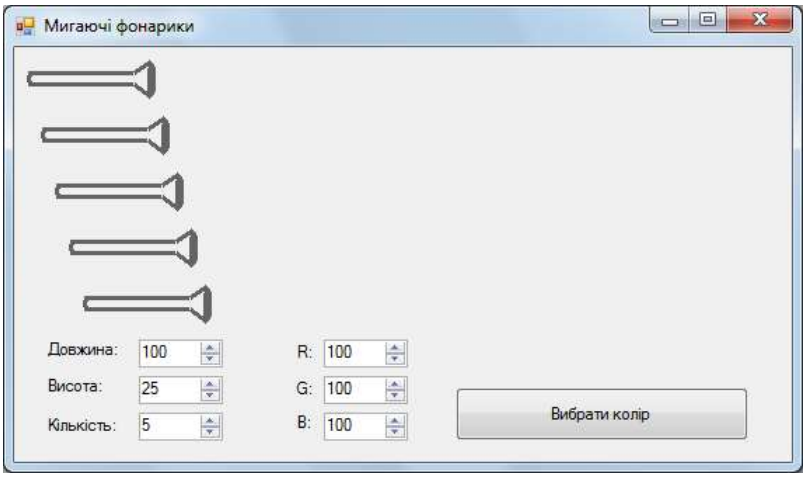


А її код буде, наприклад, таким:

```
public partial class Form1 : Form
    {   Parallelepiped pd;
        Piramida pa;

        public Form1()
        {InitializeComponent(); }

        private void button3_Click(object sender, EventArgs e)
        {   pd = new Parallelepiped((double)A.Value, (double)B.Value, (double)H.Value,
                (byte)CR.Value, (byte)CG.Value, (byte)CB.Value);
            pd.Info();
            Refresh();
        }
```

```
        private void Form1_Paint(object sender, PaintEventArgs e)
        {   if (pd is Parallelepiped)
                pd.draw3D(DateTime.Now.Second + 10, 10, e);
            if (pa is Piramida)
                pa.draw3D(100, 10, e);
        }

        private void button1_Click(object sender, EventArgs e)
        { colorDialog.Color=Color.FromArgb((int)CR.Value,(int)CG.Value,(int)CB.Value);
          if (colorDialog.ShowDialog()==DialogResult.OK)
          {CR.Value = colorDialog.Color.R;
           CG.Value = colorDialog.Color.G;
           CB.Value = colorDialog.Color.B;
          }
        }

        private void button2_Click(object sender, EventArgs e)
        {   pa = new Piramida((double)A.Value, (double)B.Value, (double)H.Value,
                              (byte)CR.Value, (byte)CG.Value, (byte)CB.Value);
            pa.Info();
            Refresh();
        }

        private void timer_Tick(object sender, EventArgs e)
        {Refresh();
        }
    }
```

Ще один приклад. Створимо зображення декількох мигаючих ліхтариків. В процесі виконання форма може бути, наприклад, така:



А її код може бути, наприклад, таким:

```
public partial class Form1 : Form
{   public Form1()
    {InitializeComponent();}

    private void Form1_Paint(object sender, PaintEventArgs e)
    {   Pen pen = new Pen(Color.FromArgb(255, (int)CR.Value, (int)CG.Value, (int)CB.Value), 3);
        SolidBrush br = new SolidBrush(Color.FromArgb(255, (int)CR.Value, (int)CG.Value, (int)CB.Value));
        int h=(int)H.Value, l=(int)L.Value, count=(int)Count.Value;
        for (int i = 0; i < count; i++)
        {   int startX = 10+i*10, startY = 10 + i * 40;
            e.Graphics.DrawLine(pen, startX, startY+h/3, startX+3*l/4, startY+h/3);
            e.Graphics.DrawLine(pen, startX, startY+2*h/3, startX+3*l/4, startY + 2*h/3);
            e.Graphics.DrawLine(pen, startX+3*l/4, startY+h/3, startX+7*l/8, startY);
            e.Graphics.DrawLine(pen, startX+3*l/4, startY+2*h/3, startX +7*l/8, startY+h);
            e.Graphics.DrawArc(pen, startX, startY+h/3, l/16+1, h/3, 90, 180);
            if (System.DateTime.Now.Millisecond % 100 < 50)
             e.Graphics.FillPie(br, startX+7*l/8 - (l/16 + 1)/2, startY, l/16 + 1, h, 0, 360);
            else
             e.Graphics.DrawArc(pen, startX+7*l/8 - (l/16 + 1)/2, startY, l/16 + 1, h, 0, 360); }}
```

```
    private void button1_Click(object sender, EventArgs e)
    { colorDialog.Color=Color.FromArgb((int)CR.Value, (int)CG.Value, (int)CB.Value);
      if (colorDialog.ShowDialog()==DialogResult.OK)
      {CR.Value = colorDialog.Color.R;
       CG.Value = colorDialog.Color.G;
       CB.Value = colorDialog.Color.B;}}

    private void timer_Tick(object sender, EventArgs e)
    {Refresh(); }

    private void L_ValueChanged(object sender, EventArgs e)
    {Refresh(); }

    private void H_ValueChanged(object sender, EventArgs e)
    {Refresh(); }

    private void Count_ValueChanged(object sender, EventArgs e)
    {Refresh(); }
}
```