**національний університет БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ україни**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ЛАБОРАТОРНА РОБОТА №5**

Виконав:

студент групи ІПЗ-2006б

Купчик В. Г.

Київ – 2023

**Завдання до лабораторної роботи**

**Мета роботи:** дослідити механізм серіалізації об’єктів .Net

**Завдання до лабораторної роботи:**

1. Створити об’єкт певного класу відповідно варіанту завдання. Ініціалізувати поля і

властивості об’єкта. Зберегти стан об’єкта, використовуючи механізм бінарної і XMLсеріалізації.

2. Після перезапуску програми встановити параметри об’єкта, використовуючи

механізм бінарної і XML-десеріалізації.

3. Отримати інформацію про власивості і методи класу, використовуючи механізм

рефлексії.

Варіант – 15

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

using System;

using System.Drawing;

using System.IO;

using System.Reflection;

using System.Windows.Forms;

using System.Xml.Serialization;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace WindowsFormsApp6

{

public partial class Form1 : Form

{

XmlSerializer xmlSerial;

FileStream fs;

SandClock sandclock;

Bitmap bmp;

public Form1()

{

InitializeComponent();

bmp = new Bitmap(pictureBox1.ClientSize.Width, pictureBox1.ClientSize.Height);

Type type = typeof(SandClock);

MethodInfo[] mi = type.GetMethods();

foreach (MethodInfo m in mi)

{

listBox1.Items.Add(m.ToString());

}

}

[Serializable]

public class SandClock

{

private double a;

private double b;

public double A { get { return a; } set { a = value; } }

public double B { get { return b; } set { b = value; } }

public SandClock()

{

a = 0;

b = 0;

}

public SandClock(double \_a, double \_b)

{

a = \_a;

b = \_b;

}

public void Draw(Graphics g, double x, double y)

{

g.Clear(Color.Empty);

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

PointF[] topTriangle = setTopTrianglePoints(x, y);

PointF[] bottomTriangle = setBottomTrianglePoints(x, y);

g.DrawPolygon(pn, topTriangle);

g.DrawPolygon(pn, bottomTriangle);

g.FillPolygon(new SolidBrush(Color.White), topTriangle);

g.FillPolygon(new SolidBrush(Color.White), bottomTriangle);

}

public PointF[] setTopTrianglePoints(double x, double y)

{

PointF[] points = new PointF[3];

points[0] = new PointF((float)(x - a / 2), (float)(y - b / 2));

points[1] = new PointF((float)(x + a / 2), (float)(y - b / 2));

points[2] = new PointF((float)x, (float)(y));

return points;

}

public PointF[] setBottomTrianglePoints(double x, double y)

{

PointF[] points = new PointF[3];

points[0] = new PointF((float)(x - a / 2), (float)(y + b / 2));

points[1] = new PointF((float)(x + a / 2), (float)(y + b / 2));

points[2] = new PointF((float)x, (float)(y));

return points;

}

}

private void button1\_Click(object sender, EventArgs e)

{

sandclock = new SandClock(double.Parse(textBoxA.Text), double.Parse(textBoxB.Text));

double x = pictureBox1.ClientSize.Width / 2.0;

double y = pictureBox1.ClientSize.Height / 2.0;

if (sandclock != null)

{

using (Graphics g = Graphics.FromImage(bmp))

{

sandclock.Draw(g, x, y);

}

pictureBox1.Image = bmp;

}

}

private void button2\_Click\_1(object sender, EventArgs e)

{

sandclock = new SandClock(double.Parse(textBoxA.Text), double.Parse(textBoxB.Text));

xmlSerial = new XmlSerializer(typeof(SandClock));

fs = new FileStream("sandclock.xml", FileMode.OpenOrCreate);

xmlSerial.Serialize(fs, sandclock);

fs.Close();

}

private void button3\_Click(object sender, EventArgs e)

{

sandclock = new SandClock();

if (File.Exists("sandclock.xml"))

{

fs = new FileStream("sandclock.xml", FileMode.Open);

xmlSerial = new XmlSerializer(typeof(SandClock));

sandclock = (SandClock)xmlSerial.Deserialize(fs);

fs.Close();

}

textBoxA.Text = sandclock.A.ToString();

textBoxB.Text = sandclock.B.ToString();

}

private void button4\_Click\_1(object sender, EventArgs e)

{

if (File.Exists("sandclock.dat"))

{

fs = new FileStream("sandclock.dat", FileMode.Open);

using (BinaryReader reader = new BinaryReader(fs))

{

double a = reader.ReadDouble();

double b = reader.ReadDouble();

sandclock = new SandClock(a, b);

}

fs.Close();

textBoxA.Text = sandclock.A.ToString();

textBoxB.Text = sandclock.B.ToString();

}

}

private void button5\_Click\_1(object sender, EventArgs e)

{

sandclock = new SandClock(double.Parse(textBoxA.Text), double.Parse(textBoxB.Text));

fs = new FileStream("sandclock.dat", FileMode.OpenOrCreate);

using (BinaryWriter writer = new BinaryWriter(fs))

{

writer.Write(sandclock.A);

writer.Write(sandclock.B);

}

fs.Close();

}

}

}

Изображение выглядит как текст, снимок экрана, дисплей, компьютер

Автоматически созданное описаниеРезультати виконання

Изображение выглядит как снимок экрана, текст, программное обеспечение, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, компьютер, программное обеспечение

Автоматически созданное описание

Висновок

В цій лабораторній роботі ми створили C# програму для обчислення інтегралів використовуючи метод трапецій для трьох різних функцій. Ми також додали можливість спостереження за натисканням клавіші, яка починається з нашого імені.

Контрольні питання

1. Що таке делегат?

Делегат - це тип даних в мові програмування C#, який представляє собою вказівник на метод, і використовується для створення відгуків на події.

1. Як оголошуються делегати?

Делегат оголошується за допомогою ключового слова "delegate", за яким слідує сигнатура методу, який може бути вказаний делегатом.

1. Як здійснюється виклик методів класу за допомогою делегата?

Для виклику методу класу за допомогою делегата необхідно присвоїти йому вказівник на цей метод, а потім викликати делегат як звичайну функцію.

1. Що таке багатоадресність делегатів?

Багатоадресність делегатів вказує на те, що один делегат може вказувати на багато методів. При виклику такого делегата всі вказані методи будуть викликані послідовно.

1. Що таке подія?

Подія - це спеціальний механізм в C#, який дозволяє сповіщати інші частини програми про те, що сталася певна подія, і надавати можливість підписуватися на цю подію.

1. Як оголошуються події?

Подія оголошується у класі за допомогою ключового слова "event" і визначає делегат, який буде відповідати за обробку цієї події.

1. Як використовуються обробники подій?

Обробники подій - це методи, які підписуються на події і виконуються, коли подія відбувається. Для підписування обробників на подію використовується ключове слово "+=" в контексті об'єкта події.