

Лабораторна робота № 5

Тема. Розробка та використання властивостей і індексаторів.
Мета. Формування вмінь і навиків створення та використання властивостей і індексаторів об'єктів. Закріплення вмінь і навиків використання класів, об'єктів, підпрограм, функцій вводу-виводу та перетворення типів. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання.

1. Чим поля класу відрізняються від його методів?
2. Що таке властивості об'єкта? Чим властивості об'єкта відрізняються від полів? Чому для властивості зазначають тип даних?
3. Чому доступ до даних об'єкта рекомендується виконувати через його властивості?
4. Як заборонити зчитування/запис властивості об'єкта?
5. Що таке індексатори? Що спільного між властивостями та індексаторами?

Завдання.

1. Відкрийте розроблений раніше додаток з описом базового та породжених класів згідно варіанту. Відмініть виклик методу інформування з конструктора базового класу. Доповніть один з розроблених породжених класів властивостями для коректного зчитування/запису значень його полів.
2. Створіть ще один породжений клас від класу з властивостями. Використайте ці властивості в породжених класах для виводу інформації про об'єкти та виконання обчислень.
3. Модифікуйте форму, доповнивши її кнопками для супроводження життєвого циклу (виконання дій) над двома об'єктами нового класу.

Варіанти:

№ варіанту	Базовий клас	Породжений клас № 1	Породжений клас № 2
1.	Літак	Військовий літак	Цивільний літак
2.	Риба	Океанічна риба	Прісноводна риба
3.	Банківська картка	Картка для виплат	Картка універсальна
4.	Телефон	Стаціонарний телефон	Мобільний телефон
5.	Лист	Поштовий лист	Електронний лист
6.	Автомобіль	Вантажний автомобіль	Легковий автомобіль
7.	Верхній одяг	Куртка	Пальто
8.	Продукт	Овоч	Фрукт
9.	Взуття	Чоботи	Туфлі
10.	Цукерки	Шоколадні цукерки	Карамельні цукерки
11.	Тварина	Свійська тварина	Хижа тварина
12.	Книга	Паперова книга	Електронна книга
13.	Періодичне видання	Газета	Журнал
14.	Миючий засіб	Пральний порошок	Мило
15.	Годинник	Механічний годинник	Електронний годинник

Наприклад, опис класу з властивостями модифікованого паралелограма *Parallelogram* та породженого від нього класу **прямокутного** паралелепіпеда *Parallelepiped* може виглядати так:

```

class Parallelogram:Figure2D
{
    private double a, b, alfa;

    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 double Alfa
    {get { return alfa; }
    set { if (value >= 0)
        alfa = value;}}

    public Parallelogram(double a, double b, double alfa)
    {
        A = a; B = b; Alfa = alfa;
    }

    public override double area()
    {
        return a * b * Math.Sin(alfa / 180 * Math.PI);
    }

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

    public override void Info()
    {
        MessageBox.Show("Дані паралелограма:\nдві сторони по "+a.ToString()+
            " та дві по "+b.ToString()+" од.;\nплоща: " + area().ToString() +
            " кв. од.;\nпериметр: " + perimeter().ToString() +
            " од.;\nдва кути по " + alfa.ToString() + " і два кути по " +
            (180 - alfa).ToString() + " градусів", "Інформація",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

class Parallelepiped : Parallelogram
{
    private double h;

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

    public Parallelepiped(double a, double b, double h)
        : base(a, b, 90)
    { H = h;
    }

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

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

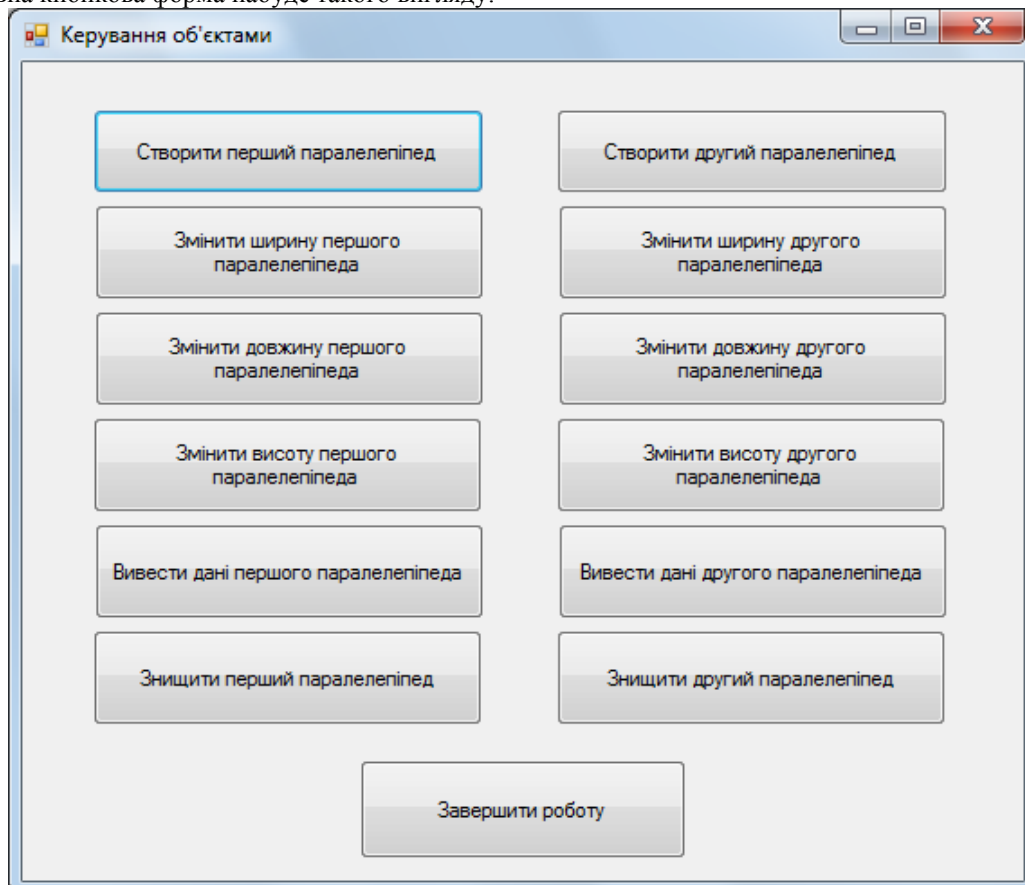
```

```

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

```

А головна кнопкова форма набуде такого вигляду:



Для керування двома паралелепіпедами створимо дві вказівки в формі і допоміжну процедуру:

```

public partial class Form1 : Form
{
    Parallelepiped p1 = null, p2 = null;

    public Form1()
    {
        InitializeComponent();
    }

    static bool inputDouble(ref double x, string Povidom)
    {
        string S;
        S = x.ToString();
    Повтор:
        S = Interaction.InputBox(Povidom, "Введення", S);
        try
        {
            x = Convert.ToDouble(S);
        }
        catch (System.FormatException)
        {
            if (MessageBox.Show("Ви ввели не число\n\nБажаєте повторити?",
                "Увага", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
                goto Повтор;
            else
                return false;
        }
        return true;
    }
}

```

```
...
}
```

Нагадаємо, що, наприклад, процедура обробки події натиснення кнопки *Створити перший паралелепіпед* буде приблизно такою:

```
private void button1_Click(object sender, EventArgs e)
{
    double a=4, b=5, h=8;
    if (!inputDouble(ref a, "Введіть довжину першого паралелепіпеда"))
        return;
    if (!inputDouble(ref b, "Введіть ширину першого паралелепіпеда"))
        return;
    if (!inputDouble(ref h, "Введіть висоту першого паралелепіпеда"))
        return;
    p1 = new Parallelepiped(a, b, h);
}
.
```

А процедура обробки події натиснення кнопки *Змінити висоту першого паралелепіпеда* може бути такою:

```
private void button9_Click(object sender, EventArgs e)
{
    if (p1==null)
    {
        MessageBox.Show("Перший паралелепіпед ще не створено!", "Увага!",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    double h=p1.H;
    if (inputDouble(ref h, "Змініть висоту першого паралелепіпеда"))
        p1.H = h;
}
}
```

4. Створіть новий клас чи використайте існуючий і опишіть у ньому індексатор. Продемонструйте його використання.

Наприклад, для розрахунку номера вкладеного прямокутного паралелепіпеда, об'єм якого не перевищує задане число, доцільно створити такий клас і його індексатор:

```
class EmbeddedParallelepiped : Parallelepiped
{
    public EmbeddedParallelepiped(double a, double b, double h) : base(a, b, h)
    { }

    public Parallelepiped this[int i]
    {
        get { if (i <= 0) return new Parallelepiped(A, B, H);
            else return new Parallelepiped(A / Math.Pow(2, i), B / Math.Pow(2, i),
                H / Math.Pow(2, i)); }
    }
}
}
```

Тоді використання об'єкту цього класу при натисненні відповідної кнопки (справа знизу на зразку) буде таким:

```
private void button4_Click(object sender, EventArgs e)
{
    var vp=new EmbeddedParallelepiped(5,7,6);
    int i=1;
    double VMax = 17;
    if (!inputDouble(ref VMax, "Введіть обмеження об'єму")) return;
    while (vp[i].volume() > VMax) i++;
    MessageBox.Show("Вкладений паралелепіпед № " + i.ToString() + " має об'єм " +
        vp[i].volume() + " куб. од.", "Інформація", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
}
}
```