

Лабораторна робота № 3

- Тема.** Реалізація наслідування елементарних класів.
Мета. Формування вмінь і навиків проектування елементарних класів з використанням наслідування. Закріплення вмінь і навиків використання функцій вводу-виводу. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Для чого використовується наслідування класів?
2. Як реалізується наслідування класів? Від скількох класів можливе наслідування в C#?
3. Як з методу класу-нащадка викликати метод батька?
4. Як з методу класу-батька викликати метод нащадка?
5. У чому різниця між статичними і нестатичними класами?

Завдання.

1. Створіть додаток-форму Windows. Опишіть чи скопіюйте з попередньої лабораторної в модуль форми *під класом форми* базовий клас згідно варіанту, передбачивши в ньому не менше трьох полів, двох конструкторів різних видів, двох методів для виконання обчислень. Передбачте в цьому класі змінну *count* для підрахунку кількості відповідних створених об'єктів та *power* – для зберігання номера поточного об'єкта. Створіть також у цьому класі метод *Info* для виводу інформації про об'єкт. *Забезпечте виклик цього методу з усіх конструкторів.* Реалізуйте деструктор класу для виведення номера знищеного об'єкта та зменшення їх загальної кількості.
2. Опишіть *під базовим класом* два породжених класи від базового класу згідно варіанту і створіть у кожному з них не менше двох конструкторів різних видів. Доповніть їх додатковими полями та методами. Створіть також у цих класах методи *Info* для виводу інформації, але *не викликайте його з конструкторів.*

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 базового класу *Figure2D* та породжених від нього класів *Parallelogram* та *Square* може виглядати так:

```

class Figure2D
{public static int count=0;
  int nomer;

  public Figure2D()
  {Info();
   count++;
   nomer = count;
  }

  ~Figure2D()
  {MessageBox.Show("Знищується фігура № " + nomer.ToString(), "Увага!",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
   count--;
  }

  public double area()
  {return 0; }
  public double perimeter()
  {return 0; }
  public void Info()
  {MessageBox.Show("Дані абстрактної двовимірної фігури", "Інформація",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
  }
}

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

  public Parallelogram()
  { }

  public Parallelogram(double sa, double sb, double salfa): base()
  {a = sa; b = sb; alfa = salfa; }

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

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

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

class Square : Figure2D
{double a;

  public Square()
  { }

  public Square(double sa): base()
  {a = sa; }

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

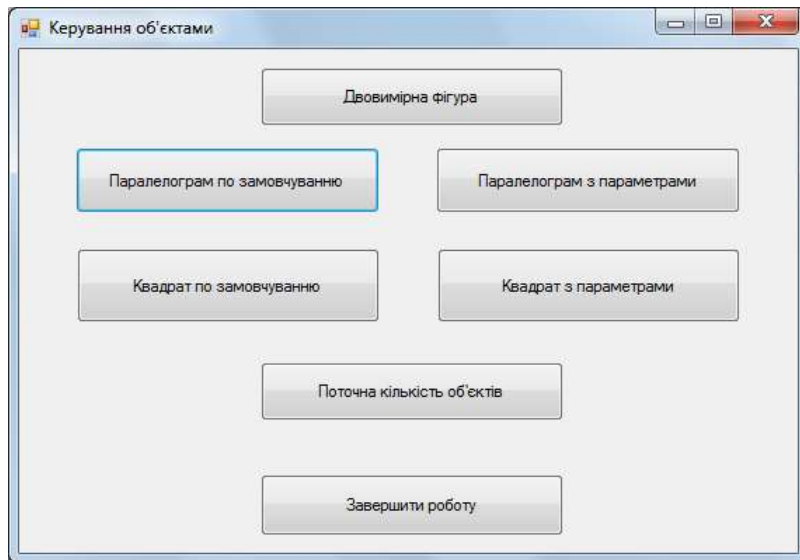
  public double perimeter()
  {return 4 * a; }

  public void Info()
  {MessageBox.Show("Дані квадрата:\nчотири сторони по " + a.ToString() + " од.;\nплоща: " + Math.Sqrt(area()).ToString() +
    "\n од.;\nплоща " + area().ToString() + " кв. од.;\nпериметр " + perimeter().ToString() +
    "\n од.;\nчотири кути по 90 градусів", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
  }
}

```

3. Розробіть у формі кнопку для генерації об'єкта базового класу, по дві кнопки для створення об'єктів кожного з породжених описаних класів (викликаючи, відповідно, конструктор по замовчуванню і конструктор з параметрами та метод *Info* створених об'єктів). В нижній частині форми створіть кнопку для виведення поточної кількості створених об'єктів та кнопку для завершення роботи і забезпечте її функціонування. Завантажте додаток на виконання та переконайтеся у дієздатності кнопок. Чому спочатку виводиться інформація про базовий об'єкт по замовчуванню, а вже потім – інформація про створений об'єкт? Як забезпечити вивід інформації лише про створюваний об'єкт?

Наприклад, для керування об'єктами класів двовимірної фігури, паралелограма та квадрата доцільно створити таку головну кнопку форму:



Тоді процедура обробки події натиснення, наприклад, кнопки *Паралелограм з параметрами* буде такою:

```
private void button7_Click(object sender, EventArgs e)
{
    double a, b, alfa;
    a=Convert.ToDouble(Interaction.InputBox("Введіть першу сторону паралелограма",
        "Введення", "4"));
    b=Convert.ToDouble(Interaction.InputBox("Введіть другу сторону паралелограма",
        "Введення", "5"));
    alfa=Convert.ToDouble(Interaction.InputBox("Введіть гострий кут між сторонами "+
        "паралелограма в градусах", "Введення", "30"));
    Parallelogram p2 = new Parallelogram(a, b, alfa);
    p2.Info();
}
}
```

Процедура ж обробки події натиснення кнопки *Поточна кількість об'єктів* буде така:

```
private void button7_Click(object sender, EventArgs e)
{
    MessageBox.Show("Поточна кількість об'єктів: " + Figure2D.count.ToString(),
        "Увага!", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
}
```

4. Перед описом методу *Info* в базовому класі встановіть модифікатор *virtual*, а в породжених – *override*. Що змінилося при створенні об'єктів за допомогою кнопок? Чому? Звідки взялися значення полів перед викликом кожного методу *Info*?