

Лабораторна робота № 6

- Тема.** Перевантаження унарних та бінарних операцій.
Мета. Формування вмінь і навиків перевантаження операцій над об'єктами. 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. Навіщо виконують перевантаження операцій?
2. Чому перевантаження операцій завжди статичне?
3. Які операції можливо перевантажити в C#?
4. Як при порівнянні об'єктів забезпечити перевірку рівності їх полів, а не вказівок?
5. Для яких операцій застосовується перевантаження ++?

Завдання.

1. Відкрийте розроблений раніше додаток-форму з описом базового та породжених класів згідно варіанту. Перевантажте в одному з ваших класів хоча б один унарний оператор.
2. Перевантажте в одному з ваших класів оператори перевірки об'єктів на рівність та нерівність.
3. Розробіть у формі кнопки для демонстрації дій перевантажених операцій та використайте метод *Info()* для відображення стану об'єктів до і після використання цих операцій.

Наприклад, перевантаження операцій рівності, нерівності та інкрементації для класу прямокутного паралелепіпеда *Parallelepiped* може виглядати так:

```
class Parallelepiped : Parallelogram
{
    public double minSide()
    {return Math.Min(Math.Min(A, B), H);
    }

    public double maxSide()
    {double max = A;
     if (B > max) max = B;
     if (H > max) max = H;
     return max;
    }

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

    public static bool operator ==(Parallelepiped p1, Parallelepiped p2)
    {return p1.minSide() == p2.minSide() &&
           p1.maxSide() == p2.maxSide() &&
           p1.perimeter() == p2.perimeter();
    }

    public static bool operator !=(Parallelepiped p1, Parallelepiped p2)
    {return !(p1 == p2);
    }

    public static Parallelepiped operator ++(Parallelepiped p)
    {return new Parallelepiped(p.A + 1, p.B + 1, p.H + 1);
    }
}
```

Тоді процедура обробки події натиснення кнопки *Інкрементації паралелепіпеда* буде така:

```
private void button8_Click(object sender, EventArgs e)
{var p = new Parallelepiped(5, 4, 8); p.Info();
  p++; p.Info();
  ++p; p.Info();
}
```

А процедура обробки події натиснення кнопки *Порівняння паралелепіпедів* буде, наприклад, такою:

```
private void button7_Click(object sender, EventArgs e)
{ var p1 = new Parallelepiped(4, 5, 8);
  p1.Info();
  var p2 = new Parallelepiped(5, 4, 10);
  p2.Info();
  if (p1 == p2) MessageBox.Show("Паралелепіпеди рівні");
  else MessageBox.Show("Паралелепіпеди не рівні");
}
```

```

class Parallelepiped : Parallelogram
{...
    public double minSide()
    {return Math.Min(Math.Min(A, B), H);
    }

    public double maxSide()
    {double max = A;
     if (B > max) max = B;
     if (H > max) max = H;
     return max;
    }

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

    public static bool operator ==(Parallelepiped p1, Parallelepiped p2)
    {return p1.minSide() == p2.minSide() &&
           p1.maxSide() == p2.maxSide() &&
           p1.perimeter() == p2.perimeter();
    }

    public static bool operator !=(Parallelepiped p1, Parallelepiped p2)
    {return !(p1 == p2);
    }

    public static Parallelepiped operator ++(Parallelepiped p)
    {return new Parallelepiped(p.A + 1, p.B + 1, p.H + 1);
    }
}

```