

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк ____ / 61

Лабораторна робота №11. Перевантаження операторів

Мета роботи: Навчитися перевантажувати оператори для реалізації зручної роботи з користувацькими типами даних у C#.

Теоретичні відомості

Перевантаження операторів у C# — це механізм, який дозволяє змінювати поведінку стандартних операторів (+, -, *, /, ==, <, тощо) для роботи з користувацькими типами (класами та структурами). Це робить код зручнішим для читання та більш інтуїтивним під час роботи з об'єктами. Ключова ідея перевантаження операторів — надати об'єктам можливість взаємодіяти один з одним так само, як це роблять вбудовані типи. Наприклад, якщо ми хочемо додавати два об'єкти класу `Vector` як математичні вектори, то можна перевантажити оператор +.

Для перевантаження оператора використовується ключове слово `operator` у поєднанні зі статичним методом, який визначає нову поведінку оператора.

```
public static тип operator символ(параметри)
{
    // Реалізація оператора
}
```

Припустимо, у нас є клас `Vector`, який представляє математичний вектор із двома координатами (X та Y). Ми хочемо додати можливість додавання двох векторів за допомогою оператора +.

```
class Vector
{
    public double X { get; set; }
    public double Y { get; set; }

    public Vector(double x, double y)
    {
        X = x;
        Y = y;
    }

    // Перевантаження оператора +
    public static Vector operator +(Vector v1, Vector v2)
    {
        return new Vector(v1.X + v2.X, v1.Y + v2.Y);
    }

    public override string ToString()
    {
        return $"({X}, {Y})";
    }
}
```

Тепер ми можемо додавати два вектори, як це робиться для чисел:

```
Vector v1 = new Vector(3, 4);
Vector v2 = new Vector(1, 2);
Vector result = v1 + v2;
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк __ / 62

```
Console.WriteLine(result); // Виведе: (4, 6)
```

Оператори порівняння (`==`, `!=`, `<`, `>`) часто використовуються для перевірки рівності або порівняння об'єктів. Для їх перевантаження важливо також перевизначити методи `Equals` і `GetHashCode`, щоб зберегти узгодженість. Розглянемо клас `Rectangle`, у якому ми хочемо порівнювати два прямокутники за їх площею.

```
class Rectangle
{
    public double Width { get; set; }
    public double Height { get; set; }

    public Rectangle(double width, double height)
    {
        Width = width;
        Height = height;
    }

    public double Area => Width * Height;

    // Перевантаження оператора ==
    public static bool operator ==(Rectangle r1, Rectangle r2)
    {
        return r1.Area == r2.Area;
    }

    // Перевантаження оператора !=
    public static bool operator !=(Rectangle r1, Rectangle r2)
    {
        return !(r1 == r2);
    }

    // Перевизначення Equals
    public override bool Equals(object obj)
    {
        if (obj is Rectangle other)
        {
            return this == other;
        }
        return false;
    }

    public override int GetHashCode()
    {
        return Area.GetHashCode();
    }
}
```

Тепер можна порівнювати прямокутники за площею:

```
Rectangle r1 = new Rectangle(3, 4);
Rectangle r2 = new Rectangle(6, 2);

Console.WriteLine(r1 == r2); // Виведе: True
Console.WriteLine(r1 != r2); // Виведе: False
```

Особливості перевантаження

1. Оператори повинні бути статичними: Методи для перевантаження завжди оголошуються як `static`, оскільки вони працюють із кількома об'єктами, а не з конкретним екземпляром.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк ____ / 63

2. Не всі оператори можна перевантажувати: Наприклад, оператори = та . не підтримують перевантаження.

3. Перевантаження має бути інтуїтивним: Поведінка перевантажених операторів має бути зрозумілою та очікуваною для користувачів класу.

Зміст роботи

Згідно з обраним у лабораторній роботі 5 варіантом теми, необхідно виконати наступну дії. Всі дії відбуваються в репозиторії OOPWPFProject.

Завдання 1: Злити гілку feature/add- abstraction в основну гілку (master або main). Створити нові гілку

```
git checkout -b feature/add-override
```

Завдання 2: У базовому класі перевантажити оператори (оператори для перевантаження та дії, що повинні виконуватись, описані в таблиці).

2.1 Арифметичні оператори: наприклад, +, -, які виконуватимуть дії з числовими властивостями (наприклад, кількість, рейтинг, ціну тощо).

2.2 Оператори порівняння: наприклад, ==, <, >, щоб перевіряти рівність або порівнювати об'єкти за ключовими властивостями (наприклад, дата, рейтинг).

2.3 Комбінуючі оператори: наприклад, +=, щоб об'єднувати властивості об'єктів.

	Поля для перевантаження	Оператори для перевантаження	Опис дій операторів
1	BirthDate, FirstName, LastName	+, ==, <	+: Комбінування імен та прізвищ ==: Перевірка однакової дати народження <: Порівняння дати народження
2	Year, Title	+, ==, <	+: Об'єднання назв ==: Перевірка, чи видання мають однаковий рік <: Порівняння по року
3	Deadline, Title	+, >, ==	+: Об'єднання завдань >: Порівняння дедлайнів ==: Однакові назви
4	Duration, Type	+, -, ==	+: Сума тривалості -: Різниця в тривалості ==: Порівняння типів
5	Quantity, Price	+, -, <	+: Об'єднання замовлень (сума кількостей і цін) -: Зменшення кількості <: Порівняння цін
6	Date, Time	==, <, ++	==: Перевірка однакових дат і часу <: Порівняння подій за датою ++: Додавання +1 дня до дати
7	ShowTime, SeatNumber	+, ==, <	+: Комбінування резервів (наприклад, для групових бронювань)

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк ____ / 64

			==: Той самий час і місце <: Час
8	PhoneNumber, Name	+, ==, >	+ : Комбінування імен ==: Ті ж номери телефонів >: Порівняння по імені
9	Date, Location	+, ==, <	+ : Об'єднання подій (наприклад, конференції) ==: Ті самі дата і місце <: Дата події
10	Year, Brand	+, ==, >	+ : Комбінування назв моделей ==: Той самий рік випуску >: Новіший рік
11	Rating, Name	+, >, ==	+ : Комбінування назв >: Місце з кращим рейтингом ==: Той самий рейтинг
12	Category, ItemName	+, ==	+ : Об'єднання назв інвентарю ==: Та сама категорія
13	Amount, Date	+, -, ==	+ : Додавання суми транзакцій - : Віднімання суми ==: Порівняння по даті
14	Price, Quantity	+, -, >	+ : Додавання кількості товарів - : Зменшення кількості >: Замовлення з вищою ціною
15	Date, PatientName	+, ==, --	+ : Об'єднання записів (наприклад, для кількох призначень) ==: Та сама дата і пацієнт --: Віднімання 1 дня від дати
16	Role, RegistrationDate	>, ==, +	>: Порівняння по даті реєстрації ==: Однакова дата реєстрації + : Комбінування ролей
17	DispatchDate, Address	>, ++, +	>: Порівняння по даті ++: Додавання +1 дня до дати + : Комбінування адрес
18	VisitDate, TicketType	+, ==, <	+ : Комбінування типу квитка ==: Перевірка однакової дати <: Порівняння дати
19	BikeCode, RentDate	+, ==, >	+ : Комбінування коду ==: Перевірка однакової дати та коду >: Порівняння дати
20	PublishDate, Title	+, ==, <	+ : Об'єднання назв публікацій у спільне видання з меншою датою (повертається новий запис) ==: Порівняння по даті та назві <: Перевірка по даті
21	JoinDate, FirstName, LastName	+, ==, <	+ : Комбінування імен та прізвищ ==: Перевірка однакової дати <: Порівняння дати
22	Amount, InterestRate, BorrowerName	+, ==, >	+ : Об'єднання кредитів (по сумі) ==: Порівняння по рейтингу та сумі

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк. ___ / 65

			>: Порівняння по рейтингу
23	Year, Name, Manufacturer	+, <, ==	+ : Об'єднання назви та виробника <: Перевірка по року ==: Перевірка чи рік, виробник та назва співпадають
24	Location, CO2, Humidity	+, ==, ++	+ : Об'єднання двох записів (в новий запис заносяться середні значення по кожному з трьох полів) ==: Повне співпадіння по трьом полям ++: +1 до CO2 та Humidity
25	PrizeFund, Game	+, ==, >	+ : Об'єднання турнірів в новий (призовий фонд додається, ігри об'єднуються) ==: Перевірка по назві гри та призовому фонду >: Порівняння по призовому фонду
26	OrderDate, OrderNumber	-, ==, <	-: Різниця в днях між датами ==: Повна відповідність дати <: Порівняння номерів
27	UsageDate, Tariff	+, ==, -	+ : Об'єднання двох записів у новий з новою датою (дата операції + 1 день) та середнім тарифом ==: Ідентичні дата та тариф -: Різниця тарифів
28	Temperature, Wind	+, ==, >=	+ : Знаходження середньої температури ==: Повна відповідність по двом параметрам >=: Порівняння по двом параметрам
29	StartDate, Budget	+, ==, >	+ : Об'єднання двох кампаній в нову (бюджет додається, дата – береться дата операції + 30 днів) ==: Відповідність по даті та бюджету >: Порівняння бюджету
30	PurchaseDate, InventoryNumber	-, ==, --	-: Різниця в днях ==: Повна відповідність по двом полям --: -1 день до дати

Завдання 3: Інтеграція з інтерфейсом користувача

3.1 Додати елементи керування (кнопки, текстові поля) для виконання операцій над об'єктами через перевантажені оператори.

3.2 Реалізувати додавання/об'єднання об'єктів через +.

3.3 Реалізувати порівняння двох об'єктів через ==, < або >.

3.4 Виводити результат операцій у DataGridView або текстове поле.

Завдання 4: Створити коміт та завантажити зміни в віддалений репозиторій.

Контрольні запитання

1. Що таке перевантаження операторів у C#?

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.05- 05.02/2/125.00.1.Б/ОК14- 2024
	Екземпляр № 1	Арк ____ / 66

2. Які оператори можна перевантажувати в C#?
3. Які оператори не можна перевантажувати в C#? Наведіть приклади.
4. Як записується синтаксис перевантаження оператора?
5. Як перевантаження операторів пов'язане з поліморфізмом?
6. Які правила слід враховувати при перевантаженні операторів у C#?
7. Що таке унарні та бінарні оператори, і як їх перевантажувати?
8. Яка роль методу Equals у перевантаженні операторів?
9. Чи можна перевантажити оператори для роботи з різними типами даних? Як це реалізувати?
10. Які обмеження існують для перевантаження операторів у структурі?