Кам’янець-Подільський національний університет

 імені Івана Огієнка

КАФЕДРА КОМП’ЮТЕРНИХ НАУК

Навчальна дисципліна «Об’єктно-орієнтоване програмування»

ЛАБОРАТОРНА РОБОТА #0110

Тема: Перевантаження операцій

Варіант №3

Виконав:

студент 1-го курсу

групи KNms1-B21

Волошин В.Ю.

Прийняла:

старший викладач,

Мястковська М.О.

Кам’янець-Подільський – 2023

1. **Хід роботи.**

Перевантаження операцій є важливою концепцією в мові програмування C#, яка дозволяє створювати більш зручний та зрозумілий код за допомогою визначення різних варіантів операцій для різних типів даних чи об'єктів.

Операції в C# можуть бути перевантажені для користувачевих типів, що дозволяє визначати, як об'єкти цих типів поводяться під час різних операцій, таких як додавання, віднімання, порівняння тощо.

Важливі аспекти перевантаження операцій включають:

1. Можливість Перевизначення Операцій:

* Розробники можуть перевизначати вбудовані оператори для своїх типів, надаючи їм бажаний функціонал.

1. Оператори та Методи:

* Перевантаження може включати як перевизначення операторів, так і визначення методів, які викликаються для виконання операцій.

1. Визначення Користувацьких Класів:

* Розробники можуть визначати свої класи та перевантажувати операції для цих класів, щоб забезпечити зручну та зрозумілу роботу з об'єктами цих класів.

1. Типизовані та Не типизовані Перевантаження:

* Операції можуть бути перевантажені як для типизованих, так і для не типизованих аргументів, що розширює можливості визначення функціоналу.

1. Інкремент та Декремент:

* Перевантаження можливе для операцій інкремента (++) та декремента (--), що спрощує роботу зі збільшенням чи зменшенням значень.

1. Оператори Порівняння:

* Розробники можуть визначати власні критерії порівняння для об'єктів своїх класів.

Перевантаження операцій сприяє зручності та читабельності коду, роблячи програми більш інтуїтивно зрозумілими. Воно дозволяє створювати короткий та елегантний код, що використовується для роботи з різними типами даних.

1. **Завдання до виконання.**

Абстрактний тип даних (АТД ) – множина з елементами типу **char.**Додатково перенавантажувати наступні операції:

“-” - видалити елемент з множини(типу set-char);

“>”- перевірка на підмножину;

“!=” - перевірка множин на нерівність.

1. **Лістинг програми.**

class CharSet

{

    private HashSet<char> set;

    public CharSet()

    {

        set = new HashSet<char>();

    }

    // Додавання елементу до множини

    public void Add(char element)

    {

        set.Add(element);

    }

    // Видалення елементу з множини

    public void Remove(char element)

    {

        set.Remove(element);

    }

    // Перевірка належності до множини

    public bool Contains(char element)

    {

        return set.Contains(element);

    }

    // Перевірка на підмножину

    public bool IsSubsetOf(CharSet other)

    {

        return set.IsSubsetOf(other.set);

    }

    // Перевантаження оператору "-"

    public static CharSet operator -(CharSet set1, char element)

    {

        set1.Remove(element);

        return set1;

    }

    // Перевантаження оператору ">"

    public static bool operator >(CharSet set1, CharSet set2)

    {

        return set1.set.IsProperSupersetOf(set2.set);

    }

    // Перевантаження оператору "<"

    public static bool operator <(CharSet set1, CharSet set2)

    {

        return set1.set.IsProperSubsetOf(set2.set);

    }

    // Перевантаження оператору "=="

    public static bool operator ==(CharSet set1, CharSet set2)

    {

        return set1.set.SetEquals(set2.set);

    }

    // Перевантаження оператору "!="

    public static bool operator !=(CharSet set1, CharSet set2)

    {

        return !set1.set.SetEquals(set2.set);

    }

    // Виведення множини на консоль

    public void Print()

    {

        Console.Write("{ ");

        Console.Write(string.Join(", ", set));

        Console.WriteLine(" }");

    }

}

        CharSet set1 = new CharSet();

        CharSet set2 = new CharSet();

        set1.Add('a');

        set1.Add('b');

        set1.Add('c');

        set2.Add('b');

        set2.Add('c');

        Console.WriteLine("Set 1:");

        set1.Print();

        Console.WriteLine("Set 2:");

        set2.Print();

        Console.WriteLine("Removing 'a' from Set 1:");

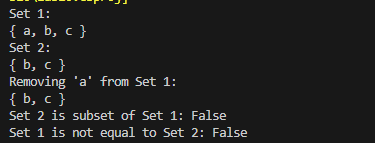
        set1 = set1 - 'a';

        set1.Print();

        Console.WriteLine("Set 2 is subset of Set 1: " + (set2 > set1));

        Console.WriteLine("Set 1 is not equal to Set 2: " + (set1 != set2));

1. **Результат роботи.**



1. **Посилання на проект.**