C# Essential

Перевантаження операторів

Після уроку обов'язково

Повторіть цей урок у відеоформаті на ITVDN.com

Перевірте як Ви засвоїли даний матеріал на TestProvider.com

Тема

Перевантаження операторів

Object

В C# всі типи, зумовлені і користувацькі, посилальні типи і типи значень, слідують безпосередньо або побічно від `Object`.

Оскільки всі класи в платформі **.NET Framework** є похідними класу `Object`, всі методи, визначені у класі `Object`, доступні для всіх об’єктів в системі.

Змінним типу `Object` можна призначати значення будь-яких типів.

Перевизначення

Методи класу Object

- `Equals` – даний метод підтримує порівняння об'єктів.
- `Finalize` – даний метод виконує операції очистки перед автоматичною утилізацією об'єкта.
- `GetHashCode` – даний метод створює число, що відповідає значенню об'єкта, який забезпечує можливість використання хеш-таблиці. Щоб перевизначити даний метод необхідно перевизначити і метод `Equals`
- `ToString` – створює зрозумілий для користувача рядок тексту, в якому описується екземпляр класу.
- `MemberwiseClone` – створює "неповну" копію об'єкта. При цьому копіюються члени, але не об'єкти, на які посилаються ці члени.

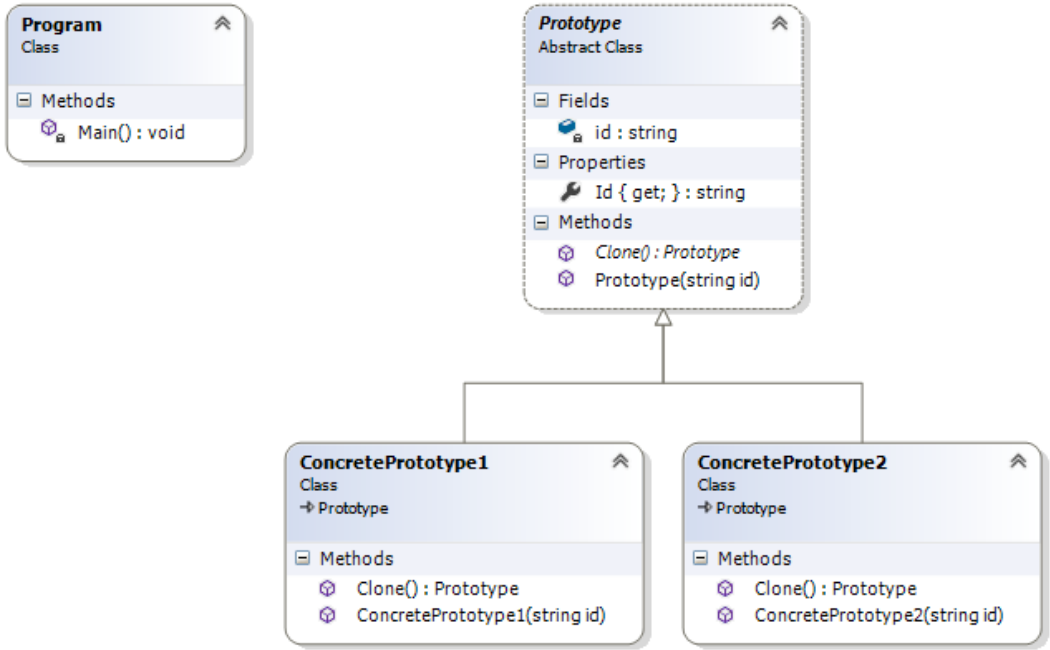
Клонування асоціації відбувається поверхово.

Граф наслідування клонується глибоко.

Прототип

Prototype

Прототип – патерн, що породжує об’єкти. Він визначає, задає види створюваних об’єктів за допомогою інтерфейсу деякого екземпляра-прототипу, і створює нові об’єкти шляхом копіювання (клонування) цього екземпляру.



Прототип – це єдиний патерн із серії «породжуючих патернів», який для створення нових об'єктів використовує не явну інстанціацію(мнстанциирование), а клонування.

Інтерфейс ICloneable

Інтерфейс `ICloneable` підтримує копіювання, при якому створюється новий екземпляр класу з тим же значенням, що і у існуючого екземпляру.

```
public interface ICloneable
{
    object Clone();
}
```

Реалізувавши інтерфейс `ICloneable`, можна створити всі умови для копіювання об’єкту. Інтерфейс `ICloneable` містить один член, `Clone`, призначений для підтримки копіювання окрім виконуваного за допомогою методу `MemberwiseClone`.

Оператори

Operators

Оператор – це елемент програми, який застосовується до одного або декількох операндів у виразі або операторі.

Оператори, які отримують на вхід один операнд, наприклад, оператор інкременту **(++)** або new, називаються унарними операторами.

Оператори, які отримують на вхід два операнда, наприклад, арифметичні оператори **(+, -, *, /)** називаються бінарними.

Перевантаження операторів

operator

В C# призначені для користувача типи можуть перевантажувати оператори шляхом визначення функцій статичних членів за допомогою ключового слова operator.

```
public static Point operator +(Point p1, Point p2)
{
    return new Point(p1.x + p2.x, p1.y + p2.y);
}
```

```
static void Main()
{
    Point a = new Point(1, 1);
    Point b = new Point(2, 2);

    Point c = a + b;
}
```

Використовувати ключове слово operator можна тільки разом із ключовим словом static.

Правила перевантажень

Оператори порівняння можна перевантажувати, але тільки парами: якщо перевантажений оператор **==**, то **!=** також має бути перевантаженим.

Зворотній принцип теж є дійсним і діє для операторів **<** та **>**, а також для **<=** та **>=**.

Для перевантаження оператора у користувацькому класі потрібно створити метод у класі з правильною сигнатурою.

Метод треба назвати "operator **X**", де **X** – ім'я або символ перевантажуваного оператора.

Унарні оператори мають один параметр, а бінарні – два. В кожному випадку один параметр має бути такого ж типу, як клас або структура, який оголосив оператор.

Оператор явного перетворення типу

explicit

```
public static explicit operator Digit(byte argument)
{
    Digit digit = new Digit(argument);
    return digit;
}
```

```
class MainClass
{
    static void Main()
    {
        byte variable = 1;

        // Явное преобразование byte-to-Digit.
        Digit digit = (Digit)variable;

        }
}
```

Ключове слово `explicit` служить для створення оператора явного перетворення типу.

Оператор неявного перетворення типу

implicit

```
public static implicit operator Digit(byte argument)
{
    Digit digit = new Digit(argument);
    return digit;
}
```

```
class MainClass
{
    static void Main()
    {
        byte variable = 1;

        // Неявное преобразование byte-to-Digit.
        Digit digit = variable;

        }
}
```

Ключове слово `implicit` служить для створення оператора неявного перетворення типу.

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association