

Covariance/Contravariance in Delegates



Covariance/Contravariance in Delegates

Er is een flexibiliteit toegelaten voor het type van parameters en return values bij delegates.

Dit wordt covariance & contravariance genoemd

- **Contravariance:**

Flexibiliteit bij het type van parameter dat wordt geaccepteerd door een delegate:

Ook afgeleide classes van het type van invoerparameters worden toegelaten

ContraVariance - Voorbeeld

```
delegate void ContraVarianceDel(Child c);  
class Parent{}  
class Child: Parent{}  
class Program {  
    static void ContraVarianceTestMethod(Parent p)  
    {  
        Child ch = p as Child;  
  
        Console.WriteLine("Contravariance demo");  
    }  
    static void Main()  
    {  
        ContraVarianceDel del = ContraVarianceTestMethod;  
        Child child = new Child();  
        del(child);  
    }  
}
```

Covariance/Contravariance in Delegates

- **Covariance:**

Flexibiliteit voor return value dat wordt geaccepteerd door een delegate:

Ook afgeleide classes van het gespecificeerde type van de return value worden toegelaten

Covariance - Voorbeeld

```
delegate Parent CovarianceDel(Child c);  
class Parent{}  
class Child: Parent{}  
class Program {  
    static Child CovarianceTestMethod(Parent p)  
    {  
        Console.WriteLine("Covariance demo");  
        return new Child();  
    }  
    static void Main()  
    {  
        ConVarianceDel del = CoVarianceTestMethod;  
        del();  
    }  
}
```

Oefening Covariance/Contravariance

- Herneem de code van class Animal en de afgeleide class Dog (uit demos) in een nieuw Console project
- Maak in de class Program een methode
static Dog AdopteerHond(string naam),
deze geeft een nieuwe hond terug met naam
nog een methode void LaatDierUit(Animal animal) die
afprint naar de console wat hij zegt als hij wordt uitgelaten
- Declareer 2 custom delegates:
 - - een delegate die een return waarde Animal vraagt en als
invoer parameter een naam van het dier
 - - een delegate die een void waarde teruggeeft en een
invoer parameter Dog vraagt.

Anonymous Methods

- **Anonymous methods** zijn methods zonder naam
 - Kunnen parameters aanvaarden en return values geven
 - Kunnen gedeclareerd worden via **delegate** keyword
- Voorbeeld:

```
class SomeClass
{
    delegate void SomeDelegate(string str);

    static void Main()
    {
        SomeDelegate d = delegate(string str)
        {
            MessageBox.Show(str);
        };
        d("Hello");
    }
}
```

Referenties

- C# Programming @ Telerik Academy
 - ♦ csharpfundamentals.telerik.com
- ♦ Telerik Software Academy
 - ♦ academy.telerik.com

