

Curs06

Curs06

Tipul delegate

Tipul delegate

Tipul `delegate` permite încapsularea metodelor (ca referință) în obiecte.

Este similar pointerilor la funcții din C++, dar de data aceasta varianta `object oriented`, prin intermediul unei referințe. Dacă scriem o aplicație în care se dorește executarea unor anumite acțiuni, pentru care cunoaștem `semnătura` acestora (tip returnat, listă parametrii), dar nu ne interesează neapărat de implementarea lor (aceasta va putea fi stabilită ulterior), putem utiliza tipul `delegate`.

În acest fel aplicația capătă un caracter general, în sensul că acțiunea poate fi oricând modificată | optimizată | schimbată.

Sintaxă

```
public delegate tip_returnat nume_delegate([lista_param]);
```

Exemplu

```
public delegate int Operation(int a, int b);
```

Acest `delegate` poate încapsula orice funcție (metodă) care returnează `double` și are doi parametri de tip `int`.

Cum se atașează o metodă la un `delegat` (instanțierea delegatului)

```
namespace DelegateApp {
    class Methods {
        public static int Add(int a, int b) {
            return a + b;
        }
        public static int Mult(int a, int b) {
            return a * b;
        }
    }

    public delegate int Operation(int a, int b);
    public delegate int Operation1();

    class InstanceMethods {
        public int A { get; set; }
        public int B { get; set; }

        public int Add() {
            return A + B;
        }
    }
}
```

```

    public int Mult() {
        return A * B;
    }
}

class Program {
    static void Main(string[] args) {
        // op este un obiect de tip delegat
        Operation op;
        op = new Operation(Methods.Mult);

        // Apelarea delegatului se face ca si cum am apela metoda incapsulata
        Console.WriteLine(op(3, 4));
        InstanceMethods obj = new InstanceMethods() { A = 10, B = 20 };
        Operation1 op1 = new Operation1(obj.Add);
        Console.WriteLine(op1());

        Console.ReadKey();
    }
}

```

Observație: Un `delegate` poate încapsula de fapt mai multe metode (`multicast` cu `+` sau `+=`)

```

static void Main(string[] args) {
    // op este un obiect de tip delegat
    Operation op;
    op = new Operation(Methods.Mult);
    op += Methods.Add;

    // Apelarea delegatului se face ca si cum am apela metoda incapsulata
    Console.WriteLine(op(3, 4));

    //InstanceMethods obj = new InstanceMethods() { A = 10, B = 20 };
    //Operation1 op1 = new Operation1(obj.Add);
    //Console.WriteLine(op1());

    Console.ReadKey();
}

```