



ĐẠI HỌC XÂY DỰNG HÀ NỘI

BM CNPM – KHOA CNTT

Bài 9

Ủy quyền(delegate)

Sự kiện (event)

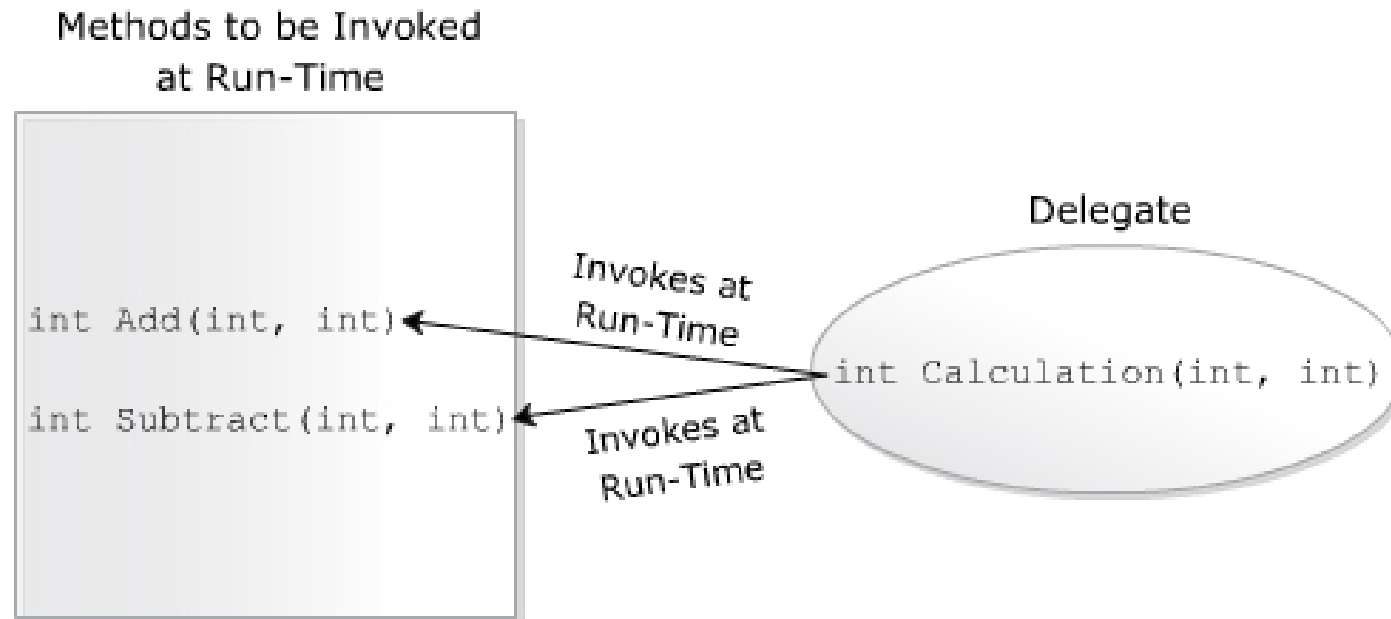


- Giới thiệu về ủy quyền (delegate)
- Tạo và sử dụng delegate
- Giới thiệu về sự kiện
- Các bước tạo sự kiện

- Delegate đơn giản chỉ là một kiểu dữ liệu tham chiếu (reference type) dùng để bao bọc bất kỳ phương thức nào có mẫu giống delegate.
- Delegate dùng trong mô hình điều khiển sự kiện của C#
- Đặc tính
 - An toàn kiểu (type safe)
 - Thực thi trực tiếp các lệnh
 - Gọi nhiều phương thức
 - Bao bọc cả phương thức tĩnh và phương thức không tĩnh

- Cú pháp

phạm_vi delegate kiểu_dl tên_delegate(ds_tham_số);



Tạo delegate 2-2



SE - FIT

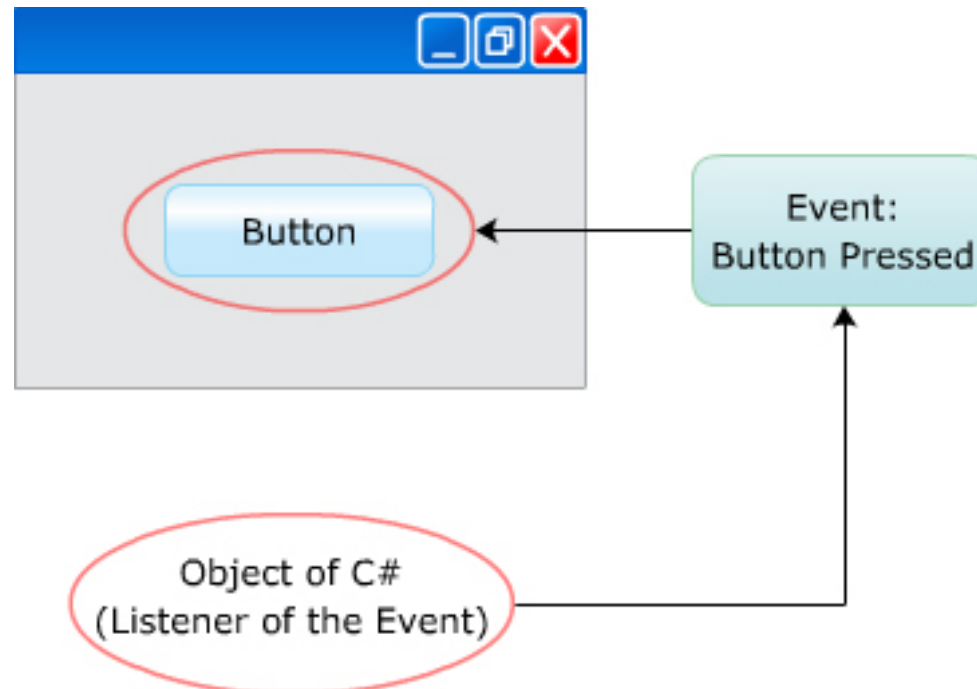
Ví dụ

```
class Program
{
    public delegate int Calculator(int a, int b);
    static void Main(string[] args)
    {
        //tạo đối tượng calculator và trỏ tới phương thức cùng mẫu (signature method)
        Calculator cal = new Calculator(Mathematic.Add);
        //thực thi phương thức Add thông qua đối tượng delegate "cal"
        int tong = cal(5, 6);
        //trỏ delegate tới phương thức có cùng mẫu khác
        cal = Mathematic.Sub;
        //thực thi phương thức Sub thông qua đối tượng delegate "cal"
        int hieu = cal(10, 3);
        Console.WriteLine("Tong:" + tong);
        Console.WriteLine("Hieu:" + hieu);
    }
}

class Mathematic
{
    public static int Add(int a, int b)
    {
        return (a + b);
    }
    public static int Sub(int a, int b)
    {
        return (a - b);
    }
}
```

Sự kiện (Event) 1-2

- Một sự kiện là một hành động được sinh ra bởi người dùng hoặc hệ thống cho phép các đối tượng thông báo cho nhau.
- Một trong những ứng dụng thường thấy nhất của delegate là event. Mỗi sự kiện thực chất là một delegate.



- **Một số đặc điểm của sự kiện**

- Sự kiện có thể được khai báo trong lớp hoặc giao diện.
- Sự kiện có thể khai báo là abstract hoặc sealed.
- Sự kiện có thể khai báo là virtual.
- Sự kiện thực thi sử dụng delegate.

- **Khai báo sự kiện**

`phạm_vi event tên_delegate tên_event;`

- **Ví dụ**

```
public delegate void MyEventHandler(string msg);  
public event MyEventHandler Click;
```

Các bước tạo sự kiện 1-3



SE - FIT

Khai báo delegate

Khai báo sự kiện

Đăng ký và điều khiển sự kiện

Phát sự kiện

Các bước tạo sự kiện 2-3



SE - FIT

Ví dụ

```
//khai báo delegate
delegate void Display();
class EventTest
{
    //khai báo sự kiện
    public event Display Print;
    //định nghĩa phương thức được gọi khi sự kiện xảy ra
    void Show()
    {
        Console.WriteLine("Đây là chương trình điều khiển sự kiện");
    }
    static void Main(string[] args)
    {
        EventTest obj=new EventTest();
        //đăng ký sự kiện và liên kết với phương thức Show qua delegate
        obj.Print+=new Display(obj.Show);
        //phát sự kiện
        obj.Print();
    }
}
```

Các bước tạo sự kiện 3-3



SE - FIT

Ví dụ

```
// ListBox dẫn xuất từ Window
public class ListBox : Window
{
    public ListBox(int top, int left): base(top, left)
    {
        Console.WriteLine("Constructor's ListBox have 2 parameter");
    }
    // Khởi dựng có tham số
    public ListBox(int top, int left, string theContents)
        : base(top, left) // gọi khởi dựng của lớp cơ sở
    {
        mListBoxContents = theContents;
    }
    public override void DrawWindow()
    {
        Console.WriteLine("DrawWindow's ListBox");
    }
    // biến thành viên private
    private string mListBoxContents;
}
public class Tester
{
    public static void Main()
    {
        Window w = new Window(100, 100);
        w.Click += w_Click;
        w.FireEvent(); // phát sinh sự kiện
        Console.ReadLine();
    }
    static void w_Click(string msg)
    {
        Console.WriteLine(msg);
    }
}
```

Question & Answer

