

Lập trình Windows

Chương 2. Ngôn ngữ lập trình C# Phần 3

Nội dung

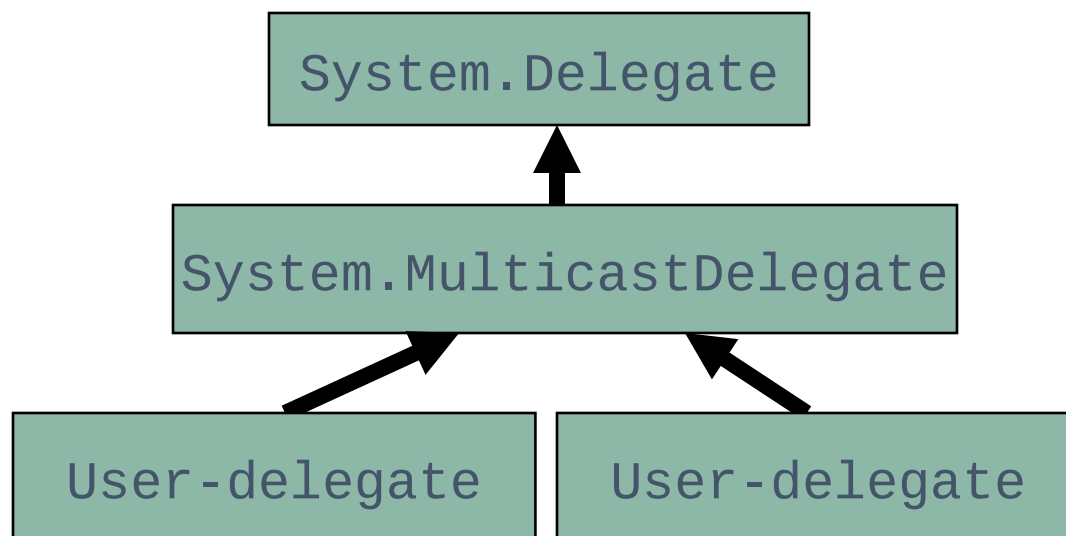


- Delegate
- Event

Delegate

Khái niệm

- Kiểu delegate là cho phép khai báo các biến (đối tượng) tham chiếu đến các phương thức (có chữ ký do delegate quy định)



Các bước sử dụng delegate

Bước 1: Khai báo kiểu delegate

Bước 2: Tạo đối tượng delegate

Bước 3: Gắn phương thức vào đối tượng delegate

Bước 4: Gọi đối tượng delegate

Khai báo kiểu delegate

```
[attributes][modifiers]  
delegate <type> <DelegateName>(parameterlist);
```

■ Ví dụ

```
void Method1();  
int Method2();  
double Method3(double x, double y);  
void Method4(EventArgs e);
```

```
delegate void DelegateType1();  
delegate int DelegateType2();  
delegate double DelegateType3(double x, double y);  
delegate void EventHandler(EventArgs e);
```

Tạo đối tượng delegate

```
DelegateName objD;  
  
objD = new DelegateName(obj.methodName);  
objD = new DelegateName(methodName);  
objD = obj.methodName;  
objD = obj.methodName;
```

Đối tượng delegate có thể được khai báo như là **field**, **tham số** của phương thức, **biến cục bộ** của phương thức

Tạo đối tượng delegate

- Bất kỳ phương thức nào phù hợp với chữ ký của delegate đều có thể gán cho đối tượng delegate
- Các phương thức này được lưu trong invocation list của delegate
- Mỗi khi một delegate được gán bằng một phương thức thì biến đó hoạt động như một phương thức

Gọi đối tượng delegate

```
objD(parameterList);
```

- Một lời gọi đến delegate sẽ gọi tất cả các phương thức trong invocation list theo đúng thứ tự
- Nếu invocation list rỗng thì sẽ ném ra một ngoại lệ `NullReferenceException`
- Nếu kiểu trả về của các phương thức khác void thì kết quả trả về của delegate là kết quả của phương thức cuối cùng

Gọi đối tượng delegate

```
namespace Delegate.Example1
{
    delegate int DelegateType(int x, int y);

    class Program
    {
        static void Main(string []args)
        {
            Program obj = new Program();
            DelegateType a = new DelegateType(obj.TinhTong);
            int kq = a(3, 4);
            Console.WriteLine("Ket qua = {0}", kq);
        }
        int TinhTong(int a, int b)
        {
            return a + b;
        }
    }
}
```

Phép toán trên delegate

```
objD1 = objD2 + objD3  
objD1 += objD4
```

```
objD1 = objD2 - objD3  
objD1 -= objD4
```

Ứng dụng

- Có 2 cách dùng quan trọng của delegate
 - Cho phép hàm được truyền như là tham số của hàm khác (tạo hàm callback)
 - Tạo các sự kiện
- Bài tập
 - Viết phương thức sắp xếp bằng interface
 - Viết phương thức sắp xếp bằng delegate

Event

Khái niệm

Event là một field thuộc kiểu delegate mà chỉ được gọi từ bên trong của lớp mà nó được khai báo

```
class ClassName
{
    public event DelegateName objE;
    ...
}
```

So sánh delegate và event

```
namespace Delegate.Example2
{
    delegate void DelegateType();

    class ClassA
    {
        public DelegateType objD;

        public void Method()
        {
            if (objD != null)
            {
                objD();
            }
        }
    }
}
```

delegate

So sánh delegate và event

```
class ClassB
{
    public void Xuat1()
    {
        Console.WriteLine("Xuat 1");
    }
    public void Xuat2()
    {
        Console.WriteLine("Xuat 2");
    }
    public void Ham()
    {
        ClassA p = new ClassA();
        p.objD = Xuat1;
        p.objD += Xuat2;
        p.objD();
        p.Method();
    }
}
```


So sánh delegate và event

```
namespace Delegate.Event.Example
{
    delegate void DelegateType();

    class ClassA
    {
        public event DelegateType
objE;

        public void Method()
        {
            if (objE != null)
objE();
        }
    }
}
```



Sự kiện

So sánh delegate và event

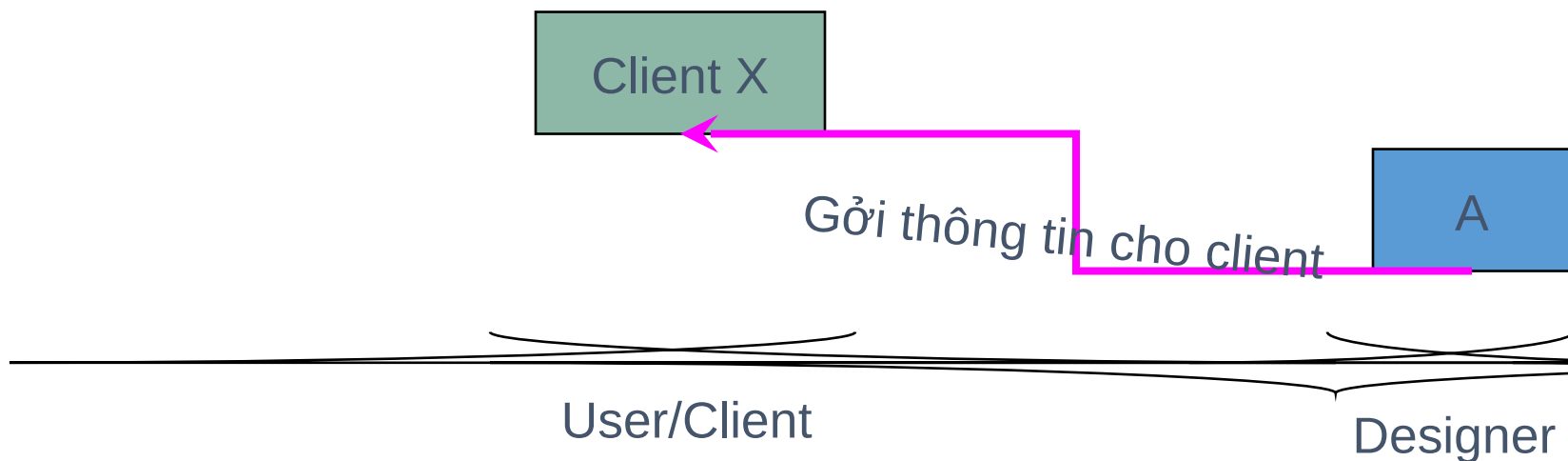
```
class ClassB
{
    public void Xuat1()
    {
        Console.WriteLine("Xuat 1");
    }
    public void Xuat2()
    {
        Console.WriteLine("Xuat 2");
    }
    public void Ham()
    {
        ClassA p = new ClassA();
        p.objE = Xuat1; //Error
        p.objE += Xuat1;
        p.objE += Xuat2;
        p.objE(); //Error
        p.Method();
    }
}
```

So sánh delegate và event

- Nhận xét
 - Event là trường hợp đặc biệt của delegate
 - Không được sử dụng phép gán = trên event
 - Chỉ được dùng phép toán += hay -= trên event
 - Event chỉ được gọi bên trong lớp nó đang khai báo

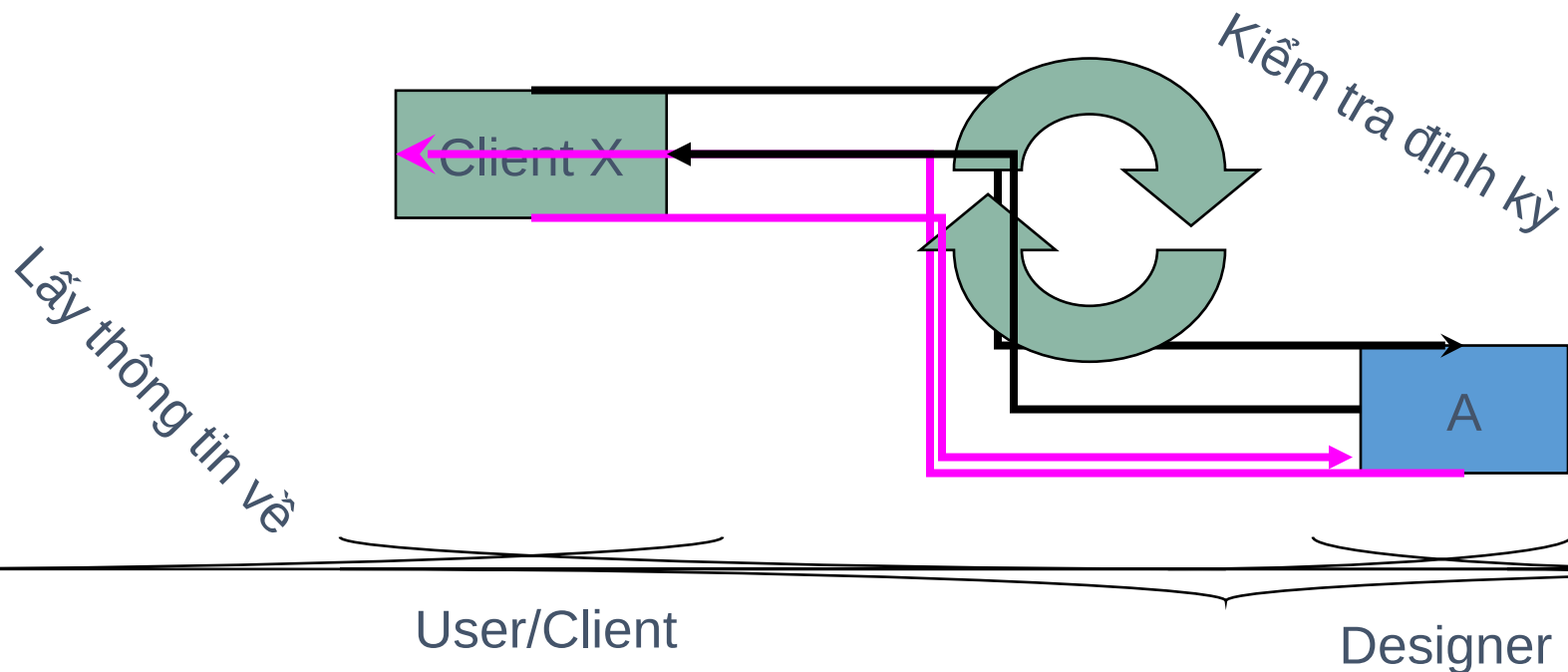
Ứng dụng

- Ngữ cảnh: Giả sử có 2 lớp A, lớp X: Lớp A là lớp tiện ích, lớp X là lớp client
 - Lớp A có nhiệm vụ tập hợp thông tin
 - Sau đó phân phối thông tin đó đến lớp X
- Làm thế nào?



Ứng dụng

- Tiếp cận thứ 1 – tiếp cận từ phía client: Kỹ thuật polling
 - X định kỳ kiểm tra bool property trong lớp A
 - Nếu giá trị này bằng true thì gọi phương thức trong lớp A để lấy dữ liệu



Ứng dụng

- Tiếp cận thứ 2 – tiếp cận từ phía designer: Lớp A gọi phương thức trong lớp X

Cung cấp tên một phương thức M cho lớp A



Gọi phương thức M khi cần thông báo cho client

User/Client

Designer

- Event của một lớp A có thể được dùng như là một cách để thông báo cho client đang dùng lớp A biết điều gì đó đã xảy ra trên đối tượng của lớp A

- Các bước tổng quát

1. Lớp A định nghĩa một sự kiện
2. Lớp X định nghĩa một phương thức để truyền cho sự kiện trong A (gọi phương thức này là event handler)
3. Lớp X đăng ký event handler này cho sự kiện của A

Ứng dụng

- Có nhiều cách tạo/sử dụng event, nhưng thông thường theo cách chuẩn như sau
 - Class designer
 - Khai báo kiểu delegate có hai tham số
 - Tham số thứ nhất thuộc lớp object, chứa đối tượng phát sinh sự kiện (VD: Đối tượng lớp A)
 - Tham số thứ hai thuộc lớp dẫn xuất từ EventArgs, chứa thông tin cần truyền cho client
 - Định nghĩa public event thuộc kiểu delegate trên
 - Gọi event khi có tác động nào đó xảy ra

Ứng dụng

- Client
 - Định nghĩa phương thức event handler theo chữ ký của kiểu delegate
 - Đăng ký event handler này với sự kiện

Ứng dụng

```
class MyClass
{
    public event EventHandler InformationAlert;

    public void Method()
    {
        ...
        if (InformationAlert != null)
            InformationAlert(this, new EventArgs());
    }
}
```

- Sự kiện:
 - Tên sự kiện là InformationAlert
 - Thuộc kiểu delegate **EventHandler** được định nghĩa trước trong .NET

Ứng dụng

- `EventHandler` được định nghĩa trong Namespace System

```
public delegate void EventHandler(object sender, EventArgs e);
```

- EventArgs là lớp được định nghĩa trong namespace System, là lớp được dùng làm cơ sở cho những lớp được dùng liên quan đến event

Ứng dụng

```
class Client
{
    void AlertMethod(object sender, EventArgs e)
    {
        // Xử lý event
    }

    public void MyMethod()
    {
        MyClass a = new MyClass();
        a.InformationAlert += AlertMethod;
    }
}
```

Q&A

