برنامه نویسی پیشرفته
C#

۱۹ آذر ۹۸

ملکی مجد

Delegate

- A delegate is an object which **refers to a method**
- similar to the function pointer in C/C++

Declaration of Delegate

[modifier] delegate [return_type] [delegate_name] ([parameter_list]);

Example:

```
public delegate int nameOfDelegate(int G, int F, int G);
```

Create instance of a sample delegate

[delegate_name] [instance_name] = new [delegate_name](calling_method_name);

Create instance of a sample delegate (Del)

```
// Declare a delegate.
public delegate void Del(string str);

// Declare a method with the same signature as the delegate.
static void Notify(string name)
{
    Console.WriteLine($"Notification received for: {name}");
}
```

```
// Create an instance of the delegate.
Del del1 = new Del(Notify);
```

Create instance of a sample delegate (Del)

```
// Declare a delegate.
public delegate void Del(string str);

// Declare a method with the same signature as the delegate.
static void Notify(string name)
{
    Console.WriteLine($"Notification received for: {name}");
}
```

```
// Create an instance of the delegate.
Del del2 = Notify;
```

Create instance of a sample delegate (Del)

```
// Declare a delegate.
public delegate void Del(string str);

// Declare a method with the same signature as the delegate.
static void Notify(string name)
{
   Console.WriteLine($"Notification received for: {name}");
}
```

```
Del del3 = delegate(string name)
   { Console.WriteLine($"Notification received for: {name}");
}; //using an anonymous method.
```

Create instance of a sample delegate (Del)

```
// Declare a delegate.
public delegate void Del(string str);

// Declare a method with the same signature as the delegate.
static void Notify(string name)
{
    Console.WriteLine($"Notification received for: {name}");
}
 Del del4 =
 name =>  { Console.WriteLine($"Notification received for: {name}"); };
 //using a lambda expression.
```

Use operator +=
to add methods to the delegate

```
public delegate void Alaki();

public void method1(){...}
public void method2(){...}
public void method3(){...}

Alaki aDelegate= new Alaki(obj.method1);
aDelegate += obj.method2;
aDelegate += obj.method3;
```

- Similarly -=

```
aDelegate -= obj.method2;
```

Simple Example

Test2.cs

Test3.cs