

## 一：委托

### 1.委托的定义

```
public T delegate MyDelegate (T a,K b);
```

### 2.委托的本质是一个类，可以把方法作为参数传递的类

委托的使用如下：MyDelegate mD = new MyDelegate(MethodName)

```
mD.invoke();
```

### 3.委托的第一作用是可以减少重复代码，逻辑判断其实也是一种方法，把逻辑判断放到委托中。

### 4.委托的第二作用是可以实现异步多线程：mD.BeginInvoke(null,null);

### 5.多播委托，使用方法如下：

(1) 是对应于同一委托类型的不同方法，可以同事一起执行

//+=为委托增加方法，形成方法链，让方法安装顺序执行

```
mD += new MyDelegate(MethodName1)
```

```
mD += new MyDelegate(MethodName2)
```

```
mD += new MyDelegate(MethodName3)
```

//-=为委托减少方法，从方法链尾部开始匹配，遇到第一个完全吻合的，移除且只移除一个，没有也不会报错。

(2) 多播委托返回值以最后要给为准。

(3) 多播委托可以一次性调用各种方法。

## 二：事件

**1.事件的定义：事件是带有event关键字的委托实例，event关键字可以保存变量不被外部调用/直接赋值。既是事件是一种特殊的委托实例。**

**2.加上了event后，即使是子类都不能调用该事件，只有声明者内部才能调用。**

### 三:序列化

序列化的概念：本质是把对象转换成一种可以持久化保存的形式，以便于保存或传输的技术。类似降维和升维。

常用的序列化的格式有：json,xml, soap,byte,二进制 binary

下面用json来举例序列化和反序列化

#### **二进制序列化的优点：**

1. 所有的类成员（包括只读的）都可以被序列化；
2. 性能非常好。

#### **XML序列化的优点：**

1. 互操作性好；
2. 不需要严格的二进制依赖；
3. 可读性强