

## 第45章 デリゲート



メソッドのアドレスをわりあて、これからメソッドを呼び出す方法があります。

デリゲートは、メソッドのアドレスを割り当てることができます。デリゲートを通じてメソッドを呼び出すことができます。デリゲートには、複数のメソッドを割り当てることにより、連続してメソッドを呼び出すことも可能です。また、デリゲートは後の章で解説する「イベント」とも深い関係があります。

デリゲートは次のように宣言します。

```
delegate 戻り値の型 デリゲート名 (パラメータリスト);
```

戻り値の型とパラメータリストの一致するメソッドがデリゲートに登録することができます。具体的には、次のようにします。

```
delegate int MyDelegate(int x);
....
MyDelegate md = new MyDelegate(メソッド名);
```

デリゲートをインスタンス化する際、引数にメソッド名を指定すればよいのです。

デリゲートはみんなから見える位置に宣言する必要があります。

では、簡単な例を見てみましょう。

```
// delegate01.cs

using System;

delegate void MyDelegate(string str);

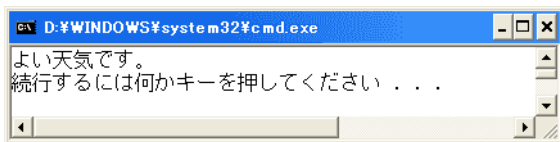
class MyClass
{
    public void show(string s)
    {
        Console.WriteLine(s + "です。");
    }
}

class delegate01
{
    public static void Main()
    {
        MyClass mc = new MyClass();

        MyDelegate md = new MyDelegate(mc.show);

        md("よい天気");
    }
}
```

実行結果は次のようになります。



この例では、Mainメソッドのあるクラス外のインスタンスメソッドのアドレスをデリゲートに格納しましたが、staticなメソッドでも大丈夫です。

次は、Mainメソッドのあるクラスのstaticなメソッドをデリゲートで呼び出します。

```
// delegate02.cs

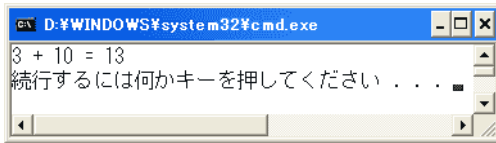
using System;

delegate int Add(int x, int y);

class delegate02
{
    static int Calc(int a, int b)
    {
        return a + b;
    }
}
```

```
public static void Main()
{
    Add add = new Add(Calc);
    Console.WriteLine("3 + 10 = {0}", add(3, 10));
}
}
```

実行結果は次のようになります。



---

[\[C# Index\]](#) [\[総合Index\]](#) [\[Previous Chapter\]](#) [\[Next Chapter\]](#)

Update 20/Sep/2006 By Y.Kumei

当ホーム・ページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。