

## 47 親から子へ機能を引き継ぐ「継承」

クラスには「継承 (inheritance)」という機能があります。リスト40は、その簡単な例です。(1)でClass2クラスのインスタンスcを作り、(2)でそのOutputメソッド、(3)でOutput2メソッドを呼びます。でも、(5)のClass2クラスを見る

と、Outputメソッドがありません。また、(6)のOutput2メソッドの中で参照しているSもありません。それらは、Class1から継承されているのです。図53のようなイメージです。

継承を行う場合には、(5)の1行目

ように、「:」(コロン)を使って継承元を指定します。

図53 ● Class1クラスを継承してClass2クラスを作り、Output2メソッドを追加した

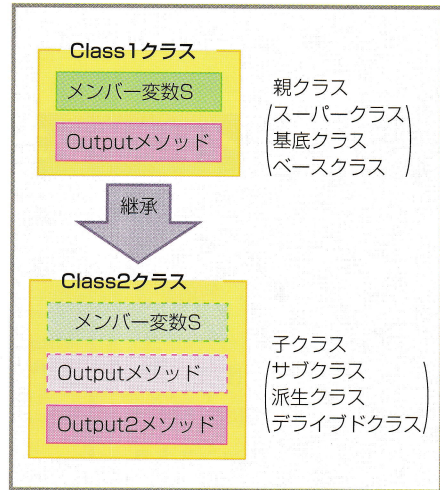
リスト40 ● 継承の例。Class1の子としてClass2を作った (Program.csの一部)

```
class Program {
    static void Main(string[] args) {
        Class2 c = new Class2(); — (1)
        c.Output(); — (2)
        c.Output2(); — (3)
    }
}

class Class1 {
    public string S = "こんにちは!"; — (4)

    public void Output() {
        Console.WriteLine(S);
    }
}

class Class2 : Class1 {
    public void Output2() {
        Console.WriteLine(S); — (6)
        Console.WriteLine(S); — (5)
    }
}
```



## 48 メソッドのオーバーライド

46番のオーバーロードと混同しそうな、メソッドのオーバーライド (override) という機能もあります。簡単に言えば子クラスで親クラスのメソッドを「上書き」する機能で、リスト41がその例です\*1。(1)でClass2のインスタンスcを作り、(2)でそのOutputメソッドを呼び出します。(5)のClass2ク

ラスの、(6)にあるOutputメソッドが呼び出されます。ここでは、「こんにちは! こんにちは!」と表示するようにしています。

Class2の親である(3)のClass1にも(4)のOutputメソッドがありますが、virtualと書いて、オーバーライドを許可しています。(6)では、overrideと書

いて、オーバーライドすることを示しています。

Class2の外部からClass1のOutputを呼び出すことはできません。でも、Class2の内部からであれば、

**base.Output()**

と記述することで、親クラスのOutputメソッドを呼び出せます。(8)をそう書き換えて、実験してみてください。

リスト41 ● メソッドのオーバーライドの例 (Program.csの一部)

```
class Program {
    static void Main(string[] args) {
        Class2 c = new Class2(); — (1)
        c.Output(); — (2)
    }
}

class Class1 {
    public string S = "こんにちは!"; — (3)

    public virtual void Output() { — (4)
        Console.WriteLine(S);
    }
}

class Class2 : Class1 {
    public override void Output() { — (5)
        Console.WriteLine(S); — (7)
        Console.WriteLine(S); — (8)
    }
}
```

\*1 new修飾子を使うと、親クラスのメソッドを宣言し直すことができます。オーバーライドとは少し違います。詳しくは「new修飾子 (C#リファレンス)」(<http://msdn.microsoft.com/ja-jp/library/435f1dw2.aspx>)を参照してください。