

共通基底クラス

System.Objectクラス

C# 言語は言語仕様のみが制定されていて、その機能の実体は .NET に依存します
つまり、文字を出力したりウィンドウを生成したりというようなアプリケーションとしての機能は
C# の機能ではなく、.NET クラスライブラリを使用して行うことになります

C# 言語のあらゆる型は、基底クラスとして **System.Object** クラスを継承します
Object クラスの継承は暗黙的に行われ、全ての型は Object に変換できるとも言えます
これは、例えば参照型の実体が何かは興味がなく、単純に参照型を受け取りたい場合
C 言語で例えるならば、汎用ポインタにあたる処理を行いたい場合にこれを用います

Object クラスはいくつかの公開メソッドを持ちます
詳しくはリファレンスを参照してもらいたいのですが
クラス設計者は **ToString()** メソッドの存在を知る必要があります

```
public virtual string ToString();
```

このメソッドは、このインスタンスに関する文字列表現を返すメソッドです
一般的に、全てのクラスはこのメソッドをオーバーライドすることが推奨されます

```
class Kitty {  
    public override string ToString() {  
        return "Kitty on your lap";  
    }  
}  
  
class Test {  
    static void Main() {  
        Kitty temp = new Kitty();  
        System.Object obj = temp;  
        System.Console.WriteLine(obj);  
    }  
}
```

このプログラムを実行すると Kitty on your lap という文字が出力されます
WriteLine() メソッドは、Object 型を受け取ることも可能であり
Object 型を受け取った場合はそのオブジェクトの ToString() メソッドを出力するからです

このプログラムを見れば、暗黙的に Object クラスが継承されていることがわかります
基底クラスである Object 型へのキャストは暗黙的に行うことができますが
Object 型から派生クラス型にキャストするには、必ず明示的に行う必要があります

C# において、全ての型は Object クラスの機能を保証しているということです
場合によっては、このクラスのメソッドをオーバーライドして独自の処理を追加することができます

[前のページへ](#)

[戻る](#)

[次のページへ](#)