

ボクシング

パンチパンチドランカー

C# 言語はネットワークで動作することを想定されているため

C 言語のポインタのようなリスクのある低水準処理を行うわけにはいきません

しかし .NET 設計者は値型と参照型の自由度を Java より汎用的に設計しました
それも、安全であり実体の型変換は暗黙的に行うという簡易さも兼ね備えてです

値型の参照型への変換機構は、構造体とクラス型の特殊な性質にあります

.NET クラスタイプライブラリの全ての型は Object クラスを継承することは以前話しました

これは、**構造体でも同じ**で、構造体は Object を継承しています

ところが、構造体は値型であり Object クラスは参照型であるという矛盾があります

ここに、C# 言語の高度な変換技術の秘密があります

構造体型を Object 型にキャストすると、**ボクシング**と呼ばれる処理が行われます

当然スポーツのボクシングではなく同じ単語の「箱に詰める」という意味の Boxing です

構造体がどのように操作されているかを本質的に理解するには

十分なハードウェアと機械語レベルの知識と経験がなければ難しいと思いますが

ボクシングは「値型を参照型に変換する処理」という簡単な説明ができます

```
struct Kitty {
    public string str;
    public Kitty(string str) {
        this.str = str;
    }
    public override string ToString() {
        return str;
    }
}

class Test {
    static void Main() {
        Kitty obj = new Kitty("Kitty on your lap");
        System.Object box = obj;
        GetObject(box);
    }
    static void GetObject(System.Object obj) {
        System.Console.WriteLine(obj);
    }
}
```

Kitty クラスが Object 型にボクシングされ、参照型として扱われています

ただしその実体である Kitty 構造体は値型であることを忘れないでください

クラス型を Object にアップキャストするのは単純なキャストにすぎませんが

構造体を参照型にキャストする行為は、内部動作が異なります

ボクシングされた参照型を、元の構造体に変換することを**アンボクシング**と呼びます

これはボクシングと異なり、明示的にキャストしなければいけません

では、実際にはボクシング操作がどのように行われているかを考えてみましょう

値型を参照型にボクシングする時、まずは参照型のインスタンスが生成されます

このインスタンスに**値型をコピー**することでボクシングが実現されます

インスタンスは値型のコピーを含んでいるので、データが失われることはありません

アンボクシングはこの逆で、値型に**インスタンスの値をコピー**します

もちろん、値型はボクシングされた参照型の元となる型と互換性がなければいけません

このように、ボクシングによる値型と参照型の相互変換はコピーによって行われます

すなわち、ボクシング処理前とアンボクシング後の値型はまったく別の存在です

```
struct Kitty {
    public string str;
    public Kitty(string str) {
        this.str = str;
    }
    public override string ToString() {
        return str;
    }
}

class Test {
    static void Main() {
        System.Object box = new Kitty("Kitty on your lap");
        Kitty unbox = (Kitty)box;
        unbox.str = "Silver Gene";

        System.Console.WriteLine(box);
        System.Console.WriteLine(unbox);
    }
}
```

このプログラムを実行すれば、box と unbox は異なることがわかります

Kitty がクラスであれば、box と unbox のインスタンスは同じであることが常識でしたが

構造体はこれをコピーで実現するので、その実体はコピーされているのです

結果 box は ToString() で "Kitty on your lap" を、unbox は "Silver Gene" を返します

