# 課題6

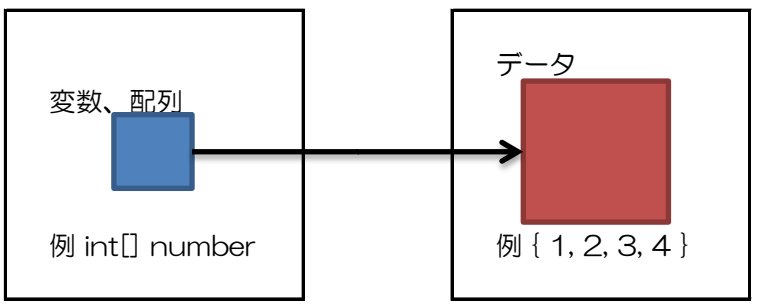
○ 課題6-1

参照型データの保護

C# の型には大きく分けて二通りのものがあります。 ひとつは「値型」と呼ばれ、もうひとつは「参照型」と呼ばれます。

int などの数値を扱う型は値型で、文字列やクラスや配列は参照型になります。

参照型の場合、下記図のように場所（アドレス）の情報を所持しています。

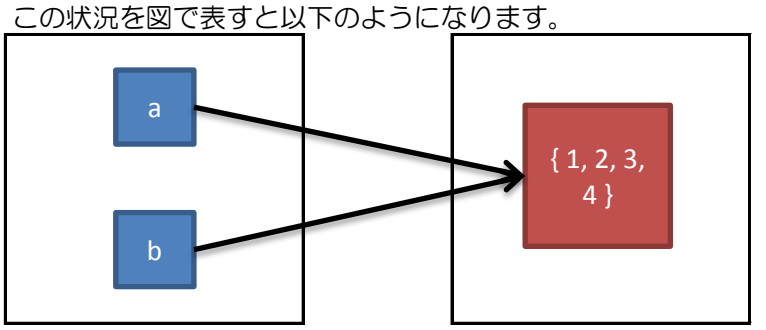



例えば

int[] a = { 1, 2, 3, 4 }:

int[] b = a:

この状況を図で表すと以下のようになります。




このように違う変数で同じデータを参照している状態ができます。
もしこのとき配列bを使用して、配列の要素を変更すると同じデータを参照している配列aでも変更されます。
これはクラスが異なっても同じことが起こりえます。
例えばList<>を返すメソッドがあったとしましょう。
もしそのメソッドで返すList<>がクラス内のメンバー変数にあるList<>を返していたとすると、
これは上述した配列a,bと同じく、クラスは異なるが同じデータを参照することになります。
このような構成は好ましくないとされています。

もしメンバー変数に格納されている参照型のデータを戻り値として返すような
メソッドを作る必要がある場合は次のように記述します。

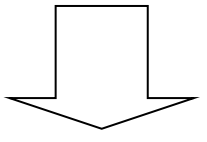
つづく

# 課題6

○ 課題6-1（つづき）
プログラム例

```
private List<string> dataList = new List<string>();

public List<string> GetData()
{
    return this.dataList;
}
```



```
private List<string> dataList = new List<string>();

public List<string> GetData()
{
    return new List<string>(this.dataList);
}
```

注：参照型のコピーは2種類あり、浅いコピー（ShallowCopy）と深いコピー（DeepCopy）と
呼ばれています。上記例はstringのListなので、浅いコピーを使っています。
もしstring以外の参照型のListの場合は深いコピーにする必要があります。
深いコピーはListの各要素もコピーすると言う処理が必要になります。

例を参考にプログラムRefactoring6をリファクタリングしてください。