

C#プログラミング講習第 6 回

1. 反復処理(foreach 文)

foreach 文は配列やリストなどのコレクションにアクセスするための構文です。Java でいう拡張 for 文です。

```
int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
  
// 普通の for 文  
for (int i = 0; i < a.Length; i++)  
    Console.WriteLine(a[i])  
  
// foreach 文  
foreach (int array in a)  
    Console.WriteLine(array);
```

サンプルの for 文を使った場合と foreach 文を使った場合では同じ挙動を示します。

foreach(型 変数名 in 配列)

のように記述します。

例えば、一回目のループでは、a[0] の 値が array に入り、array を出力しているので、結果的に a[0]の値を出力していることになります。

```
GameObject gameObject = new GameObject[10];  
  
for (int i = 0; i < a.Length; i++)  
    GameObject.Update();  
  
foreach (GameObject go in gameObject)  
    go.Update();
```

2. 型推論(var 型)

C# には、右辺からクラスの型を予測してくれる var 型があります。

var 型にはタイプ数を減らしてくれる役割があります。また、ローカル変数の定義にしか使えないので、var 型で宣言している変数はローカル変数であることが推測され、ソースコードの可読性を高めます。

```
int a = 10;
var b = 10; // b は int 型

GameObject go1 = new GameObjcet();
var go2 = new GameObject(); // go2 は GameObject 型

List<GameObject> list1 = new List<GameObject>();
var list2 = new List<GameObject>(); // list2 は GameObject のリスト
```

では、foreach 文と var 文を組み合わせて使ってみましょう。

```
// foreach 文と var 型を使わない場合
GameObject gameObject = new GameObject[10];
for(int i = 0; i < gameObject.Length; i++)
    gameObject[i] = new GameObject();
for(int i = 0; i < gameObject.Length; i++)
    gameObject[i].Update();

// foreach 文と var 型を使う場合
// (foreach 文ではオブジェクトに代入することができなため、
// インスタンスの生成は普通の for 文で行う)
var gameObject = new GameObject[10];
for(int i = 0; i < gameObject.Length; i++)
    gameObject[i] = new GameObject();
foreach(var go in gameObject)
    go.Update();
```

```
// Java で foreach(拡張 for 文)
GameObject gameObject = new GameObject[10];
for (GameObject go : gameObject)
    go.Update();
```

3. 課題

以下のプログラムを作成し、████████@gmail.com にソースコードを添付して送信してください。できるだけ var 型と foreach 文を使うこと。

- 3.1 10 個の要素を持った int 型の配列を用意する。それぞれ適当な値を入れておく。
foreach 文を用いて、その合計と平均を求めるプログラム
- 3.2 Student クラスを作り、String 型の Name プロパティを持たせる(コンストラクタも定義すること)。
Main メソッドで 10 個の要素を持った Student クラスの配列を用意し、foreach 文を用いて Name を出力する。

ヒント : 数値(または文字列)の出力

プログラムの先頭に using System;

Console.WriteLine("{0}", sum);

Console.WriteLine("{0} {1}", sum, ave);