

C#プログラミング講習第二回

1. クラス

特にプログラミング言語における「クラス」は C#だけでなく、C++や Java なども含めて、**オブジェクト指向プログラミング(Object-Oriented Programing)**の中心となる概念です。オブジェクト指向はあるオブジェクト(「もの」や「目的」)とそのオブジェクト同士の振る舞いを 1 つのクラスにまとめるプログラミングのスタイルを指しています。

たとえば、あるゲームにキャラクタ(Character)やアイテム(Item)が登場することとします。それらを**オブジェクト**とみなして Character クラス、Item クラスにします。

Character クラスには、体力や荷物(Item リスト)といった情報(変数)を持たせます。

さらに、キャラクタ同士やキャラクタとアイテムがどのような振る舞いや相互作用を持つのか Character クラスや Item クラスのメソッド(関数)にするといった具合です。

メソッドにするものは、たとえば「キャラクタ同士がぶつかり合う」、「キャラクタがアイテムを拾ったり使ったり」といったことです

オブジェクトが持つデータと振る舞いをクラスとしてまとめることで、あるオブジェクトを表現したり操作したりする手段の見通しがよくなります

Character クラス

変数 :

```
int x;    // X 座標
int y;    // Y 座標
string Name; // 名前
int Life; // 体力
```

メソッド :

```
// 更新
void Update() { }
// 描画
void Draw() { }
// 当たり判定処理
void Hit(Character character) { }
// アイテムを取得したときの処理
void GetItem(Item item) { }
```

Item クラス

変数 :

```
int x;    // X 座標
int y;    // Y 座標
int No;   // アイテムの番号
```

メソッド :

```
// 更新
void Update() { }
// 描画
void Draw() { }
// アイテムを使用したときの処理
void Use() { }
// アイテムを捨てる
void Remove() { }
```

2. new 演算子

クラスのインスタンスを作るには new 演算子を用います。

```
class Character {  
    public string Name;  
    public int Life;  
}  
  
class Program {  
    public static void Main(string[] args) {  
        Character character1 = new Character();  
        Character character2 = new Character();  
  
        character1.Name = "Red";  
        character1.Life = 20;  
        character2.Name = "Blue";  
        character2.Life = 50;  
  
        Console.WriteLine(character1.Name);  
        Console.WriteLine(character1.Life);  
        Console.WriteLine(character2.Name);  
        Console.WriteLine(character2.Life);  
    }  
}
```

それぞれのインスタンスには、異なる数値が入っていることが確認できます。

3. 継承

既存のクラスの機能、構造を共有する新たなクラスを派生することを継承といいます。継承は単に機能を拡張するだけでなく、継承させた子クラスに同じ性質を持たせるという意味もあります。この考えはインターフェースを勉強するときに大切になります。

たとえば、Character クラスと Item クラスでは、共通する変数やメソッドがいくつか存在します。int x, int y, void Update(), void Draw() です。これらは他に作るであろうオブジェクトとも共通することが考えられるのでこれを GameObject とし、これから継承することとしましょう。

```
using System;

class GameObject {
    public int x;
    public int y;

    public void Update() {
        // 更新
    }

    public void Draw() {
        // 描画
    }
}

class Character : GameObject {
    // void Update()とvoid Drawの中身を実装する
}

class Item : GameObject {
    // void Update()とvoid Drawの中身を実装する
}

public class Program {
    public static void Main(string[] args) {
        GameObject[] gameObject = new GameObject[2];
        gameObject[0] = new Character();
        gameObject[1] = new Item();

        for(int i = 0; i < gameObject.Length; i++) {
            gameObject[i].Update();
            gameObject[i].Draw();
        }
    }
}
```

Main メソッドに注目してください。継承したクラス(Character, Item)を、**継承もとのクラス (GameObject)の配列で管理**し、Update()と Draw()を呼び出しています。**しかし、実際はそれぞれのクラスで実装した処理が実行されます。**これについては、次回以降詳しく説明していく予定です。

4. 課題

4.3 の結果をメール内に記載し、4.2 でつくったソースファイルを添付したものを
■■■■@gmail.com まで送信してください

- 4.1 上のプログラムのソースをそのまま書き写せ(コメントは不要)
- 4.2 4.1 で作った Character クラスと Item クラスに void Update() メソッドを実装し、それぞれクラスの名前を出力するようにせよ
- 4.3 4.2 のプログラムをコンパイルしてきたときにでてきたエラーをコピーせよ
- 4.4 (できる人は)4.3 のコンパイルエラーを解決せよ。解決できた場合、ソースファイルを添付せよ