

前章「3. 自然言語処理の初歩」での、自然言語の文を形態素に分解するプログラムの実装方法は、毎回ことあるごとに形態素解析処理のコードをあちこちで繰り返し実装する方法でした。したがって、この方法では大変非効率といえます。PHPクラスとして一度実装し、そのクラスを繰り返し利用する方が理にかなっています。

そこでこの節では、後者のPHPクラスを繰り返し利用する方法、すなわちPHPにおけるオブジェクト指向の基本、形態素解析クラスの設計・実装、及びその使い方について解説します。

4.1.1 PHPにおけるオブジェクト指向の基本

前章の「3.3.4 php_mecab を使ってみよう」で、オブジェクト指向プログラミングについて少しだけ触れましたが、この項では、自分でクラスを実装し、そのクラスを利用する方法について解説します。

[1] クラスの実装

クラスは、オブジェクトを作成するためのひな型となるコードといえます。オブジェクトが持つプロパティやメソッドを定義します。最初に、サンプルコード「クラスのコードの書き方」を以下に示しておきましょう。

●例：クラスのコードの書き方

```
/* クラスの名前は Human とする */
class Human
{
    /* プロパティは var 文で定義する */

    /* 住んでいる地域 */
    var $habitat = '東京都';

    /* 年齢 */
    var $age = 20;

    /* 職業 */
    var $job = '大学生';
}
```



```

/* 持ち物：配列型のプロパティとして定義する */
var $item = array( 'ノートPC', 'モバイルルーター', 'リュックサック', '財布' );

/* 体力 */
var $hp = 100;

/* 集中力 */
var $mp = 50;

/** メソッド：歩く
 * 体力を消費する。気分転換になるので、集中力が回復する。
 */
function walk()
{
    if( $this->hp > 10 ){
        printf( "30分ほど歩いた。血の巡りが良くなったようだ。¥n" );
        printf( "気分転換になった。¥n" );
        $this->hp -= 10;
        $this->mp += 5;
    }else{
        printf( "へとへとで歩けない。¥n" );
    }
}

/** メソッド：勉強する
 * 集中力を消費する。
 */
function study()
{
    if( $this->mp > 5 ){
        printf( "本を開いて勉強をした。充実感が得られた。¥n" );
        $this->mp -= 5;
    }else{
        printf( "頭が痛い。目がチカチカする。勉強にうちこめそうにない。¥n" );
    }
}

/** メソッド：仮眠する
 * 体力と集中力を回復する。
 */
function nap()
{
    $this->hp = 100;
    $this->mp = 50;
    printf( "仮眠を取った。心身の疲れが取れたようだ。¥n" );
}

```

```
/** メソッド：持ち物一覧を表示する。
 *
 */
function showInventory()
{
    printf( " ●持ち物一覧¥n" );
    foreach( $this->item as $eachItem )
    {
        printf( " ・%s¥n" , $eachItem );
    }
}

/** メソッド：ステータスを表示する。
 *
 */
function showStatus()
{
    printf( "¥n" );
    printf( " ●ステータス¥n" );
    printf( "HP    %d¥n" , $this->hp );
    printf( "MP    %d¥n" , $this->mp );
    printf( "年齢  %d¥n" , $this->age );
    printf( "職業  %s¥n" , $this->job );
    printf( "¥n¥n" );
}
}
```

クラスを実装する際は、まずクラス名を指定します。"class"という言葉の後に、半角空白を1文字あけてからクラス名を記述します。本節の例では"Human"というクラス名としました。

また、クラスが持つプロパティ、メソッドの定義は以下のように、"class" に続く中括弧で囲まれた中に記述します。

●クラスのコードの基本構造

```
class (クラス名)
{
    (プロパティやメソッドの定義を記述する)
}
```

[2] プロパティの宣言

では、プロパティの定義をサンプルコード「クラスのコードの書き方」で見てください。プロパティを定義するには、「var」と記述した後に、変数を定義するコードを以下のように記述します。変数には初期値を代入するコードを記述します。

●プロパティの宣言

```
/* 住んでいる地域 */
var $habitat = '東京都';

/* 年齢 */
var $age = 20;

/* 職業 */
var $job = '大学生';

/* 持ち物：配列型のプロパティとして定義する */
var $item = array('ノートPC', 'モバイルルーター', 'リュックサック', '財布');

/* 体力 */
var $hp = 100;

/* 集中力 */
var $mp = 50;
```

「var」を用いて定義したプロパティは、外部から読み書きすることが可能となります。「public型」のプロパティとして定義されるわけです。プロパティの型には、外部から読み書きができる、このpublic型のほかに「private型」や「protected型」というプロパティの宣言方法もありますが、本書では使わないので、それらの解説は他の書籍に委ねます。

[3] メソッドの定義

続いてメソッドの定義方法を解説します。端的に言うと、以下のように、クラスの定義ブロックの内部(class{...}の内部)で関数を定義すると、それがメソッドになります。

●メソッドの定義

```
class (クラス名)
{
    function (メソッド名)
    {
```