



PHPオブジェクト指向再入門 クラスの継承と委譲

1回3,000円のWeb技術教育・アシアル塾



<http://www.asial.co.jp/school/juku.php>



本講座の目的

■ 本講座の目的

- 書籍だけでは理解することが難しいオブジェクト指向の技術や考え方を、ひとつずつ理解していきます。

■ 受講対象者

- プログラミングの基本的な手法である構造化プログラミング(順次・反復・分岐)や変数・関数の仕組みを理解している方

■ 第二回目（クラスの継承と委譲）の目標

1. クラスの継承について理解する
2. クラス図について理解する
3. 継承ではなく委譲という選択肢があることを理解する

継承とは



継承とは

- クラスのプロパティ・メソッドを受け継いで新しいクラスを作成する仕組みです
 - 元になるクラスを親クラス・基底クラス・スーパークラスなどと呼びます
 - 新しいクラスを子クラス・サブクラス・派生クラスなどと呼びます
- 継承のメリット
 - 親クラスの機能を再利用することができます
- 記述方法
 - クラスを定義する際に「extends」で親クラスを指定します
- プロパティ・メソッドのオーバーライド（上書き）
 - protectedやpublicなものは子クラスでオーバーライドが可能です
 - privateなものはオーバーライドすることができません
- 注意点
 - 単純に機能を再利用したいだけの場合、継承ではなく委譲を検討して下さい
 - ✓ 詳しくは後半の委譲で解説します



Personを継承してEmployeeを定義する

- Personクラスを親クラスとして「Employee」という子クラスを作成します
 - Employeeの仕様
 - ✓ 給与を受け取ることができます
 - ✓ 給与は勤続年数に応じて昇給します
 - ✓ プロパティ名やメソッド名の仕様
 - － 次に紹介する「クラス図」にて確認します



クラス図とは

■ クラス図とは？

- クラスを分かりやすく表現するための設計図
 - ✓ クラスのメソッド名やプロパティ名を表現できます
 - ✓ 継承や委譲の関係を表現できます
- クラス図はUML（統一モデリング言語）で規定されています

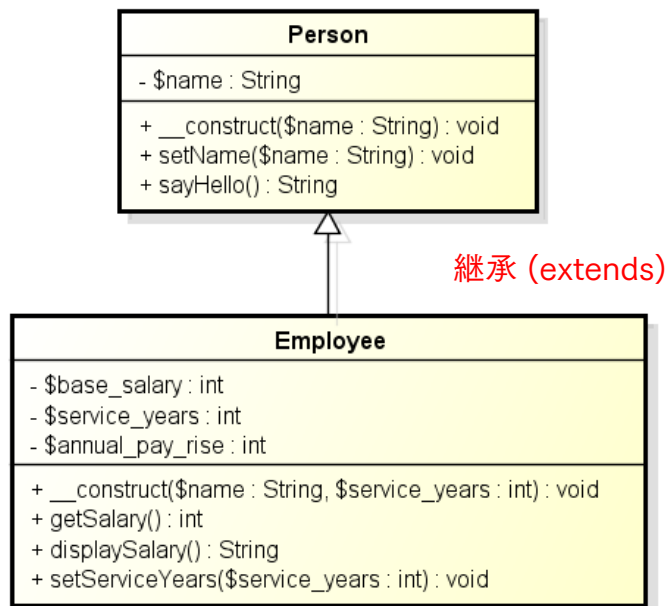
■ モデリングとは？

- 対象を分かりやすく表現するための技術
 - ✓ オブジェクト指向の世界ではUMLが利用されています
 - クラス図はUML2.0で規定された13種類の図の一つです
 - ✓ データベースのテーブル構造を表現するためのモデルとしてはER図というものが利用されています

PersonとEmployeeのクラス図

■ 記号の意味

- アクセス修飾子
 - ✓ 「-」はprivate 「+」はpublicを表します。protectedは「#」になります。
- 継承
 - ✓ 継承の関係は矢印で表すことができます



クラス名
- 属性名 : 属性の型
+ メソッド名() : 返り値の型
+ メソッド名(引数名 : 引数の型) : 返り値の型



【実習】 Personクラスの作成

- Person.phpを作成します

Person.php

```
<?php
class Person
{
    private $name = '';

    public function __construct($name)
    {
        $this->setName($name);
    }

    public function setName($name)
    {
        $this->name = trim($name);
    }

    public function getName()
    {
        return $this->name;
    }

    public function sayHello() {
        echo 'こんにちは、 ' . $this->name . 'です。';
    }
}
```




【実習】Employeeクラスの作成

- Employeeを作成します

Employee.php

```
<?php
class Employee extends Person
{
    private $base_salary = 200000;
    private $annual_pay_rise = 10000;
    private $service_years = 0;

    public function __construct($name, $service_years)
    {
        $this->setName($name);
        $this->setServiceYears($service_years);
    }

    public function setServiceYears($service_years)
    {
        $this->service_years = $service_years;
    }

    public function getSalary()
    {
        $salary = $this->base_salary + ($this->annual_pay_rise * $this->service_years);
        return $salary;
    }

    public function displaySalary()
    {
        echo $this->getName(). "さんの給与は" . number_format($this->getSalary()) . "円です";
    }
}
```



【実習】Employeeオブジェクトの作成

- Employeeを呼び出します

employee_client.php

```
<?php
require_once('Person.php');
require_once('Employee.php');

$taro = new Employee('アシアル太郎', 10);
$taro->displaySalary();
```



継承のポイント

■ 継承のポイント

- EmployeeはPersonの機能を利用することができます
 - ✓ setName()メソッドを再利用して名前をセットすることができます
 - \$this->nameはprivateのため、Employeeから直接変更することはできません。
- オーバーライド
 - ✓ EmployeeはPersonの機能をオーバーライドすることができます
 - ✓ 今回は__constructメソッドをオーバーライドしています
 - 親クラスの__constructは呼ばれなくなります
 - ✓ 親クラスのprivateなメソッドはオーバーライドすることができません
 - ✓ オーバーライドしていても子クラスから親クラスのメソッドを呼び出すことは可能です
 - 例えば、子クラスの__construct()中で親クラスの__construct()も呼び出せます
 - » 親クラスのメソッドを呼びたい場合は parent::__construct(); のように記述します。

■ 注意事項

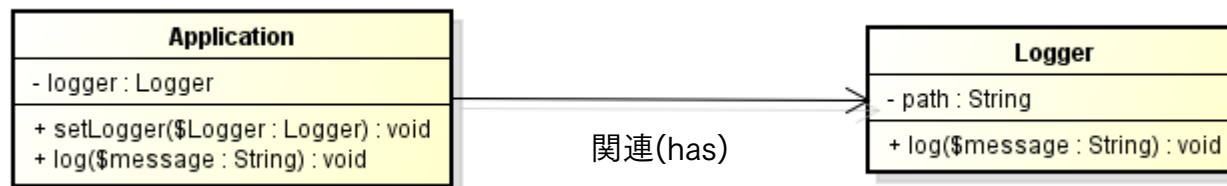
- メソッドから直接echoなどで出力するとエスケープ処理やテストが大変になるので避けましょう。
 - ✓ getXXXのように値をreturnで返すようにします。

委譲とは

委譲とは

■ 委譲 (delegation)

- あるクラスの処理の一部を「別のクラスに任せる」こと
- 例えばアプリケーション全体を管理するApplicationクラスがあったと仮定
 - ✓ 「ログを出力する」という処理が必要な場合
 - 具体的な処理はApplicationクラス内には直接記述しない
 - Loggerクラスを実装し、具体的な処理はそのクラスに任せる（委譲する）
- ApplicationとLoggerの関連を表したクラス図



■ 委譲のメリット

- クラスの役割分担を行うことができる
 - ✓ 巨大なクラスができてしまうことを防ぐことができる
 - ✓ Loggerを別のApplicationで再利用することも可能



【実習】 ApplicationとLoggerを実装しよう

■ Application

Application.php

```
<?php
class Application
{
    private $logger;

    public function setLogger(FileLogger $logger)
    {
        $this->logger = $logger;
    }

    public function log($message)
    {
        $this->logger->log($message);
    }
}
```



【実習】 ApplicationとLoggerを実装しよう

■ FileLogger(Logger)

FileLogger.php

```
<?php
class FileLogger
{
    private $path = '';

    public function __construct($path)
    {
        $this->setPath($path);
    }

    public function setPath($path)
    {
        $this->path = $path;
    }

    public function log($message)
    {
        file_put_contents($this->path, $message . PHP_EOL, FILE_APPEND);
    }
}
```



【実習】 ApplicationとLoggerを実装しよう

application_client.php

```
<?php
require_once('FileLogger.php');
require_once('Application.php');

$logger = new FileLogger('/tmp/log.txt');

$app = new Application();
$app->setLogger($logger);

$app->log('アプリケーション起動');
$app->log('処理を受け付けました');
$app->log('アプリケーション終了');
```




継承と委譲の使い分け

■ 継承を使う場合

- 子クラスが親クラスに対して is-a関係になる場合
 - ✓ Employee is a Person
- logメソッドを再利用したいという理由だけで継承を利用してはいけない
 - ✓ Application is a logger (NG!)

■ 委譲を使う場合

- 再利用したいクラスに対して has-a関係になる場合
 - ✓ Application has a Logger

【発展的な内容】



オブジェクトの型判定

■ オブジェクトの判定が必要な理由

- 特定のオブジェクトを必要とする処理を行う場合、オブジェクトの型を判定する必要があります。

■ 判定方法

- instanceof演算子
 - ✓ オブジェクトが特定のクラスに属しているかどうかを判定します
- タイプヒントング
 - ✓ 関数やメソッドの引数で受け取ることのできる値を、特定のクラスに属しているオブジェクトだけに制限します



オブジェクトの型判定方法

■ instanceof

- instanceof 演算子でクラスを指定する

```
<?php
require_once('Person.php');
require_once('Employee.php');

$taro = new Person('太郎');
$hanako = new Employee('花子');

var_dump($taro instanceof Person);
var_dump($hanako instanceof Person); // 花子は従業員かつ人間でもあるのでtrueになる
var_dump($taro instanceof Employee); // 太郎は従業員ではないのでfalseになる
var_dump($hanako instanceof Employee);
```

■ タイプヒント

- 関数やメソッドを定義する際、引数名とセットでクラス名を指定する

```
public function setLogger(FileLogger $logger)
{
    $this->logger = $logger;
}
```



【発展課題】 継承

- EmployeeでsayHello()メソッドをオーバーライドしてみよう
- Employeeを更に継承して新しいクラスを定義してみましょう
 - 仕様例：Programmerクラス
 - ✓ ソースコードを記述することができる
 - ✓ 年功序列ではなく違う指標で給与が決まる
- クラス図を書いてみよう
 - 今日利用したクラス図を書いてみる
 - Employeeを継承した新しいクラスのクラス図を書いてみよう

【CM】 アシアルスクール



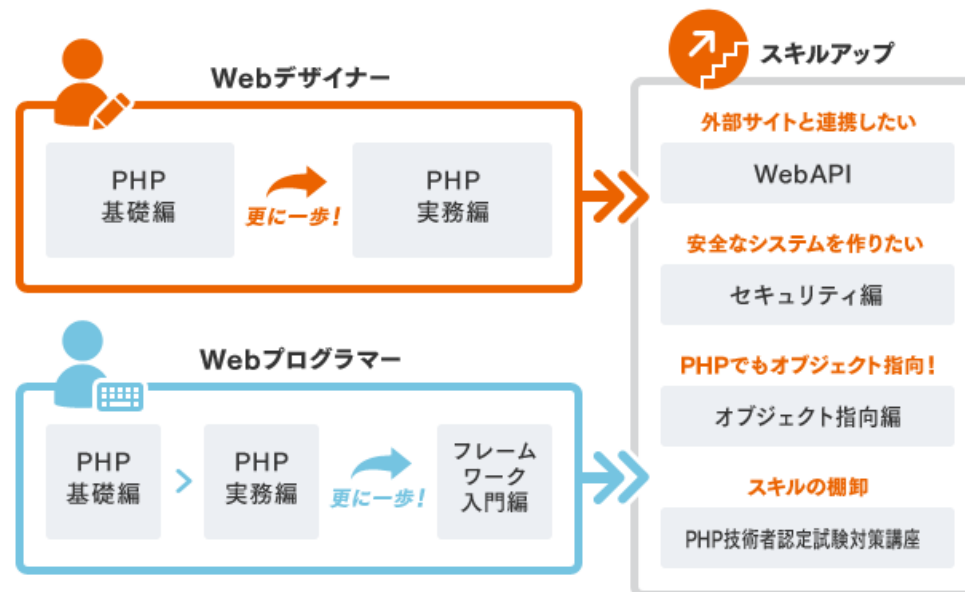
アシアル塾のご案内

- 1回3,000円でWeb技術を約90分学習できる格安の教育サービスです
 - 60分の講義+30分の復習&質問タイム
- 主な対象
 - IT業界やWeb業界で働いているプロの方の学習を応援します！
 - ✓ Webデザイナーの方大歓迎
 - 技術を分かりやすく教えます
- 絶対理解保証キャンペーン実施中
 - もし講義内容にご満足いただけなかった場合は返金いたします
 - ✓ (返金手数料のみ頂戴いたします)
- URL
 - <http://www.asial.co.jp/school/juku.php>



アシアルスクールのご案内

- Web関連技術を集中的かつ体系的に学べる研修サービスです
 - 半日～二日間の集中講義を中心に実施
 - PHPやJavaScript、LinuxといったWeb関連技術が学べます
- URL
 - <http://www.asial.co.jp/school/guide.php>
- 例：PHP講座の学習ロードマップ





企業研修のご案内

■ アシアルではWeb技術に関する企業研修を行っています

- 技術職採用の社員に向けた新人研修・中途採用者向け研修
- 総合職向けのWeb技術入門研修

■ 主な実績

- グリー株式会社様
- 株式会社ドワンゴ様
- 株式会社朝日新聞社様

■ URL

- <http://www.asial.co.jp/business/training/>