

PHPによるデザインパターン入門 □ □

秀和システムから発売となった「**PHPによるデザインパターン入門**」(ISBN4-7980-1516-4・2006/11/23発売)を執筆しました(共著です)。



「PHPを使ってGoFパターンを見ていこう」的な書籍になっています。GoFパターンについては、それぞれパターンの説明とサンプルコードという構成です。サンプルコードは、CentOS4.4/Windows XP (SP2) + PHP5.1.x/5.2.0で動作確認しています。

目次は以下の通りです。

- 1章 デザインパターンの世界へようこそ
 - デザインパターンって何？
 - デザインパターンとは？
 - オブジェクト指向
 - GoFパターン
 - デザインパターンのメリット・デメリット
 - デザインパターンを使うメリット
 - デザインパターンを使うデメリット
 - PHPとオブジェクト指向
 - PHPとは？
 - PHP5でのオブジェクト指向開発
- 2章 環境設定
 - 本書で対象とする環境
 - Apache HTTP Serverのインストール
 - PHP5のインストール
- 3章 予備知識
 - UMLについて
 - クラス図
 - シーケンス図
- 4章 GoFデザインパターン ～STEP1: まずはここから
 - TemplateMethod ～処理を穴埋めする
 - Singleton ～いくつ作るかを制限する
 - Adapter ～APIを変更する
- 5章 GoFデザインパターン～STEP2: 少し慣れたら
 - Factory Method ～生成処理と使用処理を分離する
 - Facade ～シンプルな唯一の窓口
 - Iterator ～順々にアクセスする
- 6章 GoFデザインパターン～STEP3: どんどん使ってみよう
 - Abstract Factory ～関連する部品をまとめて作る工場
 - Bridge ～実装と機能の架け橋
 - Builder ～生成の手順と手段を分離する
 - Chain of Responsibility ～処理のたらい回し
 - Command ～要求をクラスで表す
 - Composite ～木構造を表す
 - Decorator ～かぶせて機能UP
 - Flyweight ～同じものは一度しか作らない
 - Interpreter ～言語の文法表現を通訳する
 - Mediator ～すべては相談役が知っている
 - Memento ～スナップショットを取る

Observer ~状態変化を通知する
Prototype ~コピーして作る
Proxy ~具体的な実装を隠す身代わり
State ~状態を表す
Strategy ~戦略を切り替える
Visitor ~要素と要素に対する操作を分離する

正誤表

- p.158 クライアント側コード(chain_of_responsibility_client.php)[2007/01/18追加]
プルダウンで「任意」以外を選択した場合、Validatorオブジェクトのチェーンがうまく作られません。修正版([chain_of_responsibility_client.php.zip](#))をダウンロードし、展開後差し替えてください。なお、[Nullオブジェクトパターン](#)を併せて適用したサンプルも用意しました([ChainOfResponsibility_NullObject.zip](#))。こちらのchain_of_responsibility_client.phpを見ていただくと、チェーンの作成部分が非常にすっきりしていることが分かります。ここでも、NullValidationHandlerクラスが他のValidatorと同じValidationHandlerクラスを継承しているところがポイントです。