|  |  |
| --- | --- |
| ***ケービーソフトウェア株式会社*** | |
|  | |
|  |  |
| 資料名 | PHP言語を用いたシステム開発時のコーディングルール |

**２０１９年　９月（初版）**

**２０２０年　７月（二版）**

**改訂履歴**

|  |  |  |  |
| --- | --- | --- | --- |
| 改訂 | 日付 | 改訂内容 | 担当 |
| 初版 | 2019/9/4 | 初版発行 | 本村 |
| 二版 | 2020/7/3 | 生産管理システム用にカスタマイズ（Laravel対応） | 西田 |
|  |  |  |  |
|  |  |  |  |

目 次

[1 本書類の位置付け 3](#_Toc220492369)

[2 コーディング規約 3](#_Toc220492370)

[**2.1** **はじめに** 3](#_Toc220492371)

[**2.2** **コメントの記述** 3](#_Toc220492372)

[**2.3** **変数について** 6](#_Toc220492373)

[**2.4** **ネーミングルール** 7](#_Toc220492374)

[**2.5** **プレフィックス** 9](#_Toc220492375)

[3 コーディング時の注意点 10](#_Toc220492376)

[**3.1** **例外（エラー）処理について** 10](#_Toc220492377)

[**3.2** **エラーログファイルの出力について** 11](#_Toc220492378)

# 本書類の位置付け

本書はPHPを使用したプログラムの開発規約を記述するものである。

原則「PSR-2」に準拠する。

# コーディング規約

## **はじめに**

（１）文字コード

文字コードは「UTF-8」を使用する

（２）PHPコード

PHPコードは「<?php? >」を使用する。

またPHPコードのみのファイルは末尾の「?>」を記述しない事。

（３）１行の長さ

半角120文字を超えないようにする。

（４）行の折り返し

式を1行で書ききれない場合，以下の原則に基づいて複数の行に分割する。

・コンマの後で区切る。

・演算子の前で区切る。

・2つ目以降のインデントでは，前の行と同じ位置に左端を揃える。

（５）予約語　True/False/Null

PHPの予約語は小文字を使用する。

PHP定数であるtrue, false, nullも小文字とする。

（６）フォント

日本語（マルチバイト）の表示・入力を可能とする事。

フォントは「MSゴシック」とする。

※CSSファイルにて定義すること。

（７）インデント

タブによるインデントとし、サイズは4とする。

（８）改行コード

CRLFとする。

（９）文字列

文字列を囲む場合は、シングルコーテーションを使用する。

（１０）開発時の実行環境

ローカル環境にIISを設定する。

Laravelの簡易サーバ「php artisan serve」での実行は行わない。

（１１）ルートURL

サーバーのルートURLをVIEWで参照する場合は、ブレードテンプレートの URL メソッドを使う。

（１２）トランザクション処理

トランザクションにはDB::transaction を使う。

## **コメントの記述**

（１）冒頭のコメント

/\*

\* @ファイル名

\* ファイル説明

\*

\* @create 日付　名前

\*

\* @update

\*

\*/

　例）

/\*

\* @config.php

\* 設定ファイル

\*

\* @create 2020/04/01 T.Yamada

\*

\* @update

\*

\*/

（２）クラスのコメント

/\*\*

\* クラスの説明

\*

\* @create 日付　名前

\* @update

\*/

　例）

/\*\*

\* データベース接続・切断処理をおこなうクラス

\*

\* @create 2020/04/01　T.Yamada

\* @update

\*/

（３）メソッドのコメント

/\*\*

\* メソッドの説明

\*

\* @param 引数名　説明

\* @return 返り値　説明

\*

\* @create 日付　名前

\* @update

\*/

例）

/\*\*

\* データベースへの接続をおこなう

\*

\* @param stmt　ステートメント

\* @return boolean true:成功、false:失敗

\*

\* @create 2020/04/01　T.Yamada

\* @update

\*/

（４）変数・定数のコメント

　例）

$intLevel = 1; // インデントのレベル

$intSize = 10; // テーブルのサイズ

1行につき1つの宣言だけを書くようにして，コメントを書きやすくする。

下記のようには記述しない。

×int intLevel, intSize;

（５）その他コメント

　・１行コメント

if (条件) {

// この条件に対応した処理

...

}

　・行末に書くコメント

非常に短いコメントであればコードと同じ行に書く。文と区別するために十分な間隔を空けること。

if (a == 2) {

return true; 　// 特別なケース

} else {

return isPrime(a); // a が奇数のときだけ働く

}

　・コード変更時のコメント

変更箇所をコメントで残さない。SVNの履歴追跡機能を使うことを前提として、コメントで古いコード

を残すことを禁止する。

ただしヘッダーコメントに、変更概要を記述させる。メインファイルにはコメントしない。

ファイルごとのコメントとする。「@update」が複数ある場合は、下が最新とする。

※HTML、Javascript上にはコメントを残さない

## **ネーミングルール**

（１）namespace

Laravelのフォルダ構成・ネーミングルールに従う。

（２）クラス名

クラス名はUpperCamelCase で定義する。

例)

class Raster;

class ImageSprite;

（３）メソッド名

メソッド名はcamelCaseで定義する。

小文字で始まる動詞とする。

複数の単語で構成される場合，2つ目以降の単語は大文字で始める。

必ずaccess修飾子を定義する事。

例)

run();

runFast();

getBackground();

（４）変数名

小文字で始める。複数の単語で構成される場合，2つ目以降の単語は大文字で始める。

ループ変数は$i, $j, $kとする。

例)

$i;

$value;

$unitText;

（５）定数名

すべて大文字のアンダーバー（\_）で分割した単語で構成する。

クラス定数についてはaccess修飾子を定義する事。

例)

define (“MIN\_WIDTH”, 4);

define (“MAX\_WIDTH”, 999);

define (“GET\_THE\_CPU”, 1);

## **プレフィックス**

変数・ポインタ・構造体等のプレフィックスは、使用しない。

# コーディング時の注意点

## **例外（エラー）処理について**

try-catchブロックは使用せず、例外の場合はLaravelに処理を任せ、処理を記述しない。

（Laravelが処理し、エラー画面が表示される）

意図的にtry-catchにて例外を拾う必要がある場合は、使用してもよい。

　注意）

　try-catchブロックはループ外に置く事！！

　try-catchブロックがループの内部にあると、コードの実行速度が落ちる可能性がある。