|  |  |
| --- | --- |
| ***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)

# Mục đích tài liệu

Tài liệu này mô tả quy tắc code của program sử dụng PHP

# Quy tắc coding

## **Mở đầu**

（１）Endcode

Endcode sử dụng「UTF-8」

（２）Code PHP

Code PHP sử dụng「<?php? >」

Và file chỉ có code PHP thì không viết「?>」ở cuối

（３）chiều dài 1 dòng

Không vượt quá 120 ký tự haflsize

（４）lặp lại dòng

Trường hợp không thể viết hết biểu thức ở 1 dòng thì tách thành nhiều dòng theo quy tắc sau đây

・ngắt sau command

・ngắt trước operator

・ở indent thứ 2 trở đi thì căn chỉnh mép trái ở vị trí giống với dòng trước

（５）từ dành riêng　True/False/Null

Từ dành riêng của PHP sử dụng chữ in thường

true, false, null là hằng số PHP cũng để chữ in thường

（６）Font

Có thể input, hiển thị tiếng nhật (multibyte)

Font là「MSゴシック」

※định nghĩa ở file CSS

（７）Indent

Indent sẽ tùy thuộc vào tab, size là 4

（８）code cách dòng

CRLF

（９）chuỗi ký tự

Trường hợp bao quanh chuỗi ký tự, thì sử dụng single-quotation – nháy đơn

（１０）môi trường chạy khi code

ở môi trường Local thiết lập IIS

không thực hiện chạy ở server đơn giản 「php artisan serve」 của Laravel

（１１）Root URL

Trường hợp tham chiếu Root url của server bằng VIEW thì sử dụng method url của Blade template

（１２）xử lý transaction

ở transaction sử dụng DB::transaction

## **Viết comment**

（１）comment mở đầu

/\*

\* @ファイル名 (file name)

\* ファイル説明 (giải thích file)

\*

\* @create 日付(ngày)　名前 (tên)

\*

\* @update

\*

\*/

　Ví dụ）

/\*

\* @config.php

\* 設定ファイル (file setup)

\*

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

\*

\* @update

\*

\*/

（２）comment class

/\*\*

\* クラスの説明 (giải thích class)

\*

\* @create 日付 (ngày)　名前 (tên)

\* @update

\*/

　Ví dụ）

/\*\*

\* 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）comment method

/\*\*

\* giải thích method

\*

\* @param tên param　giải thích

\* @return giá trị trả về　giải thích

\*

\* @create ngày　tên

\* @update

\*/

Ví dụ）

/\*\*

\* 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）comment biến, hằng số

　Ví dụ）

$intLevel = 1; // インデントのレベル

$intSize = 10; // テーブルのサイズ

Cố gắng mỗi dòng chỉ viết 1 khai báo，viết comment dễ

Không viết như sau

×int intLevel, intSize;

（５）comment khacs

　・comment 1 dòng

if (điều kiện) {

//この条件に対応した処理

...

}

　・comment viết ở cuối dòng

Nếu comment ngắn thì có thể viết chung 1 dòng với code

Để phân biệt được với câu lệnh thì để khoảng trống đủ冒頭

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

}

　・comment khi thay đổi code

Không để lại những đoạn code thay đổi bằng comment

Nghiêm cấm việc sử dụng chức năng theo dõi lịch sử của SVN rồi để lại code cũ dạng comment

Tuy nhiên ở header comment、viết khái quát thay đổi, không comment ở file chính

Comment theo từng file. Trường hợp có nhiều「@update」thì lấy cái bên dưới là mới nhất

※không để lại comment trên HTML、Javascript

## **Quy tắc đặt tên**

（１）namespace

Đặt tên theo cấu trúc folder , quy tắc đặt tên của Laravel

（２）class name

Class name định nghĩa bằng UpperCamelCase

Ví dụ)

class Raster;

class ImageSprite;

（３）method name

Method name ddịnh nghĩa bằng camelCase

Là động từ bắt đầu bằng chữ in thường

Trường hợp cấu thành bằng nhiều từ đơn, thì từ thứ 2 trở đi viết in hoa chữ cái đầu

Cần định nghĩa bổ ngữ access

Ví dụ)

run();

runFast();

getBackground();

（４）tên biến

Bắt đầu bằng chữ in thường, trường hợp cấu thành từ nhiều từ đơn

Thì từ từ đơn thứ 2 trở đi viết hoa chữ cái đầu。

Biến loop là $i, $j, $k

Ví dụ)

$i;

$value;

$unitText;

（５）tên hằng

Cấu tạo bằng những từ đơn in hoa được phân tách bằng dấu shift -（\_）

Về hằng class thì cần định nghĩa bổ ngữ access

Ví dụ)

define (“MIN\_WIDTH”, 4);

define (“MAX\_WIDTH”, 999);

define (“GET\_THE\_CPU”, 1);

## **Prefix**

Không sử dụng prefix của biến, pointer, cấu trúc

# Điểm chú ý khi coding

## **Về xử lý ngoại lệ**

Không sử dụng block try-catch、trường hợp ngoại lệ, thì giao xư lý cho Laravel, ko viết xử lý

（Laravel sẽ xử lý、và cho hiển thị màn hình error）

Trường hợp cần lấy ngoại lệ bằng try-catch có ý đồm thì sử dụng cũng được

Chú ý）

đặt block try-catch bên ngoài loop！！

　Block try-catch có trong loop thì có khả năng giảm tốc độ chạy của code