Object Oriented Programming Languages Pattern

# Mandatory

* Always use Camel notation.
* Do NOT use acronyms and reduced words.
* All code should be written in english.

## Files, Packages and Namespaces

* A fully-qualified namespace and class must have the following structure `\<Vendor Name>\<File Type>\(<Namespace>\)\*<Class Name>`
* Each namespace must have a top-level namespace ("Vendor Name").
* Each namespace can have as many sub-namespaces as it wishes.
* Should start as upper case.

### Examples

* `\Doctrine\Common\IsolatedClassLoader` => `/path/to/project/library/4CarrUs/php/Doctrine/Common/IsolatedClassLoader.php`
* `\Symfony\Core\Request` => `/path/to/project/library/vendor/php/Symfony/Core/Request.php`
* `\Zend\Acl` => `/path/to/project/library/vendor/stylesheet/Zend/Acl.css`
* `\Zend\Mail\Message` => `/path/to/project/library/vendor/java/Zend/Mail/Message.java`

## Classes, Objects, Variables, Functions, Methods

* If possible always make it private, if not make them protected, if still not possible then you make them public.

### Structure

### Classes

* Should start as upper case.
* If possible always use one file for a single class.

#### Structure

### Objects and Variables

* Should start as lower case.
* Always make gets and sets.
* Variables should specify the type on name.

#### Examples

* `int idade` => `int integerAge`
* `new Database main` => `new Database databaseMain`

#### Constants

* Constants should specify on name.

##### Examples

* `const double precoFixo` => `const double constantDoubleFixedPrice`

### Functions and Methods

* Should start as lower case.
* Return functions should specify the type on name.

#### Structure

##### Examples

### Switch

#### Structure

##### Examples

### Loops

#### Structure

##### Examples

### Conditionals

#### Structure

##### Examples

\* `int idade` => `int integerAge`

\* `const double precoFixo` => `const double constantDoubleFixedPrice`

\* `new Database main` => `new Database databaseMain`

Underscores in Namespaces and Class Names

-----------------------------------------

\* `\namespace\package\Class\_Name` => `/path/to/project/lib/vendor/namespace/package/Class/Name.php`

\* `\namespace\package\_name\Class\_Name` => `/path/to/project/lib/vendor/namespace/packageName/Class/Name.php`