
Design Patterns


Um guia para aplicar de forma correta

Motivação

- Código complexo e obscuro
- Classes e métodos gigantes
- Mudança em vários pontos do código
- Classes invejosas
- Todo software vai mudar (Novidades)
- Mudanças não planejadas



```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email cannot be empty';
                                    } else $msg = 'Username already exists';
                                } else $msg = 'Username must be only a-z, A-Z, 0-9';
                            } else $msg = 'Username must be between 2 and 64 characters';
                        } else $msg = 'Password must be at least 6 characters';
                    } else $msg = 'Passwords do not match';
                } else $msg = 'Empty Password';
            } else $msg = 'Empty Username';
            $_SESSION['msg'] = $msg;
        }
        return register_form();
    }
}
```



—

O que são Design Patterns (Padrões de projeto)?

—

Solução elegante para problemas que são recorrentes

(Eu explico!)



1. Introdução

—

Quanto padrões ou filosofias você precisa saber para **para escrever um bom código?**

—

Apenas dois princípios! A
alta coesão e o baixo
acoplamento.

(Eu explico!)

Alta coesão, a classe tem
poucos **OBJETIVOS**



baixo acoplamento, a
classe tem poucas
DEPENDÊNCIAS





2. Alguns Padrões



strategy pattern

Para uma mesma ação gera comportamentos diferentes.



TemplateMethod pattern

Define um esqueleto para o desejado.



State pattern

Define um estado para o pattern



Observer pattern

Conjunto de ações executadas após uma ação principal.



Decorator pattern

Encapsula a regra de outro objeto.



Builder pattern

Criar um objeto com uma interface fluente.



Adapter pattern

Transforma um objeto em outro



Bridge pattern

Criar uma ponte entre dois sistemas



Muito Obrigado!

Espero que vocês apliquem essas dicas no seu dia a dia :)