

Condensação do Princípio Aberto-Fechado

Diego Vieira

Outubro 2021

Este documento será um resumo do artigo "OCP: The Open-Closed Principle" de Robert C. Martin que é uma versão condensada do livro "The Principles and Patterns of OOD" que seria publicado posteriormente pela "Prentice Hall".

Robert C. Martin, também conhecido como "Uncle Bob" estudava bastante sobre formas de se escrever bem um código de forma que pudesse ser reutilizado e que fosse simples de fazer manutenções. Com isso ele desenvolveu o "SOLID" que é um acrônimo para vários princípios criados por ele, porém o foco deste documento será o princípio vinculado a letra "O" do acrônimo que é o "Open-Closed Principle" ou "Princípio do Aberto-Fechado". A principal ideia deste princípio era que as entidades de um programa deveriam ser abertas para a extensão, porém fechadas para modificação e a forma de fazer isso será o foco deste texto.

A ideia principal do SOLID e também do OCP é criar sistemas que durem mais do que sua primeira versão pois todos os sistemas mudam durante sua vida útil e esse fato é o que torna o OCP mais presente, já que é um princípio muito utilizado quando se precisa adicionar algo ao sistema, pois o foco é deixar o sistema o mais abstrato possível para que seja possível fazer futuras adições sem que prejudique quem já está utilizando esse sistema, por exemplo: existe um sistema de cadastro de usuários que recebe um login e uma senha apenas e está funcionando, agora se no futuro for pedido para que esse sistema tenha que utilizar alguma métrica nova para criação, dependendo de como foi escrito esse sistema será difícil adicionar isso sem prejudicar os usuários que já foram cadastrados no modelo antigo e por isso deve-se fazer com que seja possível estender as funções do sistema sem que seja necessário alterar, assim será possível o usuário novo ser cadastrado e o antigo usar a nova métrica.

Para que esse princípio seja aplicado de forma adequada é necessário um alto nível de abstração em todo o sistema e com isso será mais fácil estender os métodos sem fazer nenhuma alteração no que já está funcionando, por exemplo: se existir uma classe de um cachorro que anda e senta, caso queira ser adicionado essa classe o método "deitar" deve ser necessário que esses métodos não estejam dentro da classe "cachorro" e sim fora, assim ela chama eles para fazer essas ações, se eles estiverem dentro da classe e fosse necessário implementar o método "deitar", provavelmente geraria muitas alterações

em vários lugares do sistema e quebraria os cachorros que apenas andam e sentam, se esses métodos estivesse fora da classe e fossem chamados por ela seria simples adicionar uma nova função, simplesmente é só fazer um método chamado “deitar” e fazer essa função chamar o novo método, isso não quebraria quem utiliza os métodos antigos e tudo funcionaria de forma correta.

Este foi um breve resumo do OCP que é um princípio muito importante na orientação a objetos já que ele pode ser usado em sistemas simples e ate em sistemas bastante complexos, o principal ponto deste principio assim como todos os outros que fazem parte do SOLID é fazer com que o sistema seja de fácil reutilização e também de fácil manutenção, ele é bastante relevante para a programação orientada a objetos já que ela é feita para ser usada por vários usuários e como esse principio além de deixar o sistema mais limpo ele também faz com que fique mais fácil de entender e com isso será possível ter manutenções ate por pessoas que não estavam trabalhando na projeção inicial de um projeto que teve esse principio aplicado corretamente.