

Podcast

Disciplina: Arquitetura de Software

Título do tema: Análise e Projeto de Arquiteturas de Software

Autoria: Anderson da Silva Marcolino

Leitura crítica: Marco Ikuro Hisatomi

Abertura:

Olá, aluno! No podcast de hoje vamos falar sobre o princípio de abstrações estáveis (SAP)!

Este princípio estabelece a relação entre a estabilidade e abstração e são fundamentais no contexto das Arquiteturas de Software, pois garantirão a criação de arquiteturas mais flexíveis e menos impeditivas nos processos de manutenção, evolução e refatoração.

A abstração, no conceito da programação, refere-se ao modo de reduzir a complexidade, para tornar-se um projeto e o seu desenvolvimento mais eficiente. No contexto de programação orientada a objetos (OO), trata-se da especificação de algo que existe no mundo real, em algo que possa ser manipulado e utilizado no contexto de um software levando em consideração três itens: a identidade, que irá referenciar de modo único o objeto; as características, que são chamadas de propriedades em OO e as ações, que são chamadas de métodos, em OO. Enquanto as propriedades refletem o que definem um objeto, os métodos correspondem às ações que o objeto realiza.

Já o princípio da estabilidade, corresponde ao quão estável é um componente no contexto de sofrer ou não alterações no seu código e, conseqüentemente, seu comportamento.

Voltando ao SAP, temos a indicação que um componente estável deve ser abstrato para que essa estabilidade não impeça a sua extensão. Do outro lado, um componente instável deve ser concreto, visto que sua instabilidade permite que o código concreto dentro dele seja facilmente modificado.

Diz-se que para um componente ser estável, ele deve consistir de interfaces e classes abstratas de modo que possa ser estendido. Com componentes estáveis e extensíveis, obtêm-se arquiteturas menos restritas.

Nem todos os componentes devem ser estáveis, uma vez que se todos assim o fossem, o sistema seria imutável. Um sistema deve ser projetado de modo que tenhamos componentes estáveis e instáveis, para poder modificar a solução.

O SAP, junto com o princípio das dependências estáveis (SDP), que indica que um componente que seja difícil de mudar não deve ser dependente de

qualquer componente que esperamos que seja volátil, visto que o componente volátil também se tornará difícil de mudar; forma o DIP aplicável aos componentes.

O princípio da inversão de dependência (DIP) indica que os sistemas mais flexíveis são os que suas dependências de código-fonte se referem apenas a abstrações e não a itens concretos. No caso de Java, que é uma linguagem fortemente tipada, por exemplo, as declarações *use*, *import* e *include* devem se referir apenas a módulos-fonte que contenham interfaces, classes abstratas ou outro tipo de declaração abstrata. Não deve haver dependência de nada que seja considerado concreto.

Assim, ainda que o DIP é um princípio que trabalha com classes e suas definições. Uma classe é abstrata ou não. Porém, com o SAP e SDP combinados, pode se aplicar aos componentes o conceito de parcialmente abstrato e parcialmente estável. Essa junção dá à luz a métrica que permite medir o nível de abstração de um componente, muito útil para a criação de arquiteturas de software!

Gostou de saber mais sobre o princípio de abstrações estáveis (SAP)? Considerando sua importância para o contexto das arquiteturas, recomenda-se o estudo do tema em fontes bibliográficas e de artigos na Internet!

Fechamento:

Bons estudos e até uma próxima oportunidade!