

Questão Extra

[EXTRA] Como os conceitos de coesão, acoplamento e encapsulamento se relacionam e definem códigos orientados a objetos?

Resposta:

Na programação orientada a objetos, implementa-se um conjunto de classes que definem os objetos presentes no sistema de software. Cada classe determina o comportamento, definido nos métodos, e estados possíveis, atributos, de seus objetos, assim como o relacionamento com outros objetos. Entre as características que fazem parte de uma classe estão o encapsulamento, a alta coesão e o baixo acoplamento.

A ideia de encapsular, isto é, esconder todos os membros de uma classe, além de esconder como funcionam os métodos do sistema. Encapsular é fundamental para que o sistema seja suscetível a mudanças, visto que não precisaremos mudar uma regra de negócio em vários lugares, mas sim em apenas um único lugar. Isso ocorre porque a regra em questão encontra-se encapsulada em um único ponto.

A coesão está fortemente relacionada à responsabilidade única da unidade funcional. Demonstra coerência e unidade conceitual no relacionamento com os outros componentes da unidade funcional. Ou seja, um método coeso realiza uma única função conceitual, servindo a apenas um propósito específico.

O acoplamento refere-se ao quanto uma unidade funcional depende de outra para funcionar. Uma unidade funcional pode ser um método, função ou mesmo uma classe. Quanto maior a dependência entre as unidades funcionais, mais fortemente acopladas elas estão.

Vê-se, portanto, que essas três características definem com precisão um código orientado a objetos.

- Já que há uma grande quebra do código em partes menores, métodos, que fazem, em geral, apenas uma única função, logo, servindo exclusivamente a um propósito específico e bem definido, podemos entender que o conceito de coesão é altamente presente em POO.
- O baixo acoplamento, por sua vez, se demonstra ao atestar que o grau de dependência existente entre as unidades funcionais, que podem ser métodos, funções ou mesmo classes, é muito baixo.
- Por fim, o encapsulamento que nada mais é do que proteger membros de outra classe de acesso externo, permitindo somente sua manipulação de forma indireta é uma característica que perpassa todo o código, já que as visibilidades modificam os meios de acessos e, conseqüentemente, promovem maior segurança a partes do código que devem permanecer protegidas ou privadas.