

Princípios em Projeto de Software

Atividades de Aprendizagem e Avaliação

Aluno: André de Carli Dias RA: 2160552

Use esta cor em seu texto

1. Considerando o conteúdo do texto do link “Princípios em Projeto de Software”

a) Comente a afirmação de John Ousterhout.

R: A afirmação vai de encontro indiretamente com problemáticas reais de projeto de software, assumindo que existe um projeto com partes e que precisamos dividi-las de forma a permitir com que não fique tão extensa que atrapalhe o restante do projeto, mas que não fique tão pequena a ponto de não ficar dependente de diversas outras tarefas.

b) Comente o conceito de Integridade Conceitual.

R: Integridade conceitual é uma peça importante para o desenvolvimento de um produto coeso, que possui “personalidade” por assim dizer, e não um amontoado de funções.

c) Cite e comente a(s) estratégia(s) para tornar sistemas de software mais flexíveis e fáceis de entender.

R: O autor menciona a estratégia de “esconder informação” que define basicamente a possibilidade de expor métodos genéricos e esconder a implementação dos mesmos, pois não é necessário revelar como o mesmo funciona.

d) Comente a relação entre Coesão e Separação de Interesses.

R: Ambas possuem uma característica em comum, e defende que uma classe deve implementar apenas um requisito, funcionalidade etc

e) Conceitue Acoplamento no contexto do código fonte de um software.

R: Acoplamento se refere a quanto um componente construído com código fonte possui de dependência com demais componentes.

f) Considere a tabela dada no início da seção 5.6 relacionando os **Princípios SOLID** com as **Propriedades de Projeto**. Estude o conteúdo que vem em sequência. Reproduza a tabela aqui, adicionando uma breve explicação (síntese) de cada relacionamento. (máximo de 5 linhas por item)

Dica: Enriqueça seu aprendizado com os exercícios de fixação que estão ao final do capítulo.

Princípio de Projeto	Propriedade de Projeto	Síntese
Responsabilidade Única	Coesão	Indica que um método ou função precisa ter somente uma única função.
Segregação de Interfaces	Coesão	As interfaces devem ser pequenas e específicas para cada tipo de cliente.
Inversão de dependências	Acoplamento	Ao invés de depender de classes concretas, depender de interfaces.
Prefira Composição a Herança	Acoplamento	A composição é mais vantajosa pois a relação é expressa por meio de atributos e permite a mudança em tempo de execução.
Demeter	Ocultamento de Informação	Proteger suas classes e não permitir acesso a informação via objetos intermediários.
Aberto/Fechado	Extensibilidade	Aberto para extensão fechado para modificação.
Substituição de Liskov	Extensibilidade	Se está esperando um objeto da classe base, você pode usar uma subclasse sem problemas.