

Princípios em Projeto de Software

Atividades de Aprendizagem e Avaliação

Aluno: LUCAS GABRIEL SCHUMANN GARCIA RA: 2206293

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: John quis dizer, em sua afirmação, que a parte mais complexa na ciência da computação é decompor um problema e vários pequenos problemas para que sejam resolvidos de forma independente. O texto traz como um grande inimigo também a complexidade, que deve ser brevemente combatida em grande maioria com a sua decomposição. O texto também cita algumas ferramentas frequentemente usadas para a utilização da abstração de uma entidade, por exemplo, como a utilização de bibliotecas, funções, classes, interfaces e pacotes.

b) Comente o conceito de Integridade Conceitual.

R: No texto, é citado como Integridade Conceitual, uma proposta de Frederick Brooks que diz que, um sistema, deve oferecer, de maneira uniforme as suas soluções. Tanto estando ligado à nível de código quanto de design. Cita, como exemplo de um sistema que fere a Integridade Conceitual, um sistema que em partes, oferece resultados na moeda brasileira Real e em outras partes em dólar. Fere também ao design Íntegro telas com fontes e tamanhos de componentes diferentes, dando uma leve desconfiança ao usuário sobre a procedência. À nível de código, cita como exemplo a utilização de Camel Case em determinados trechos e Snake Case em outros. Utilização de estruturas e frameworks diferentes em determinados pontos do sistema.

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

R: O autor cita como estratégias para tornar um software mais entendível: Desenvolvimento em Paralelo, Flexibilidade a mudanças, Facilidade de entendimento. O desenvolvimento em paralelo é, basicamente, ocultar suas principais informações principais para que seja possível implementá-los em paralelo. A flexibilidade a mudanças diz que uma responsabilidade (classe) do sistema deve ser de fácil manutenção sem que interfira no resultado final de outras classes. A facilidade do entendimento está ligado alocar responsabilidades fixas para o desenvolvedor, para que não exija eu entendimento completo do sistema.

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

R: Coesão é o termo que está ligado ao “S” do SOLID, que significa Single Responsibility (Responsabilidade Única), que diz que um método deve ser responsável apenas por uma solução, sem estender suas responsabilidades ao que poderia estar atribuído para outras classes ou métodos. Separação de interesses está mais ligado que a classe ou método está INTERESSADO em fazer, solucionar apenas o que está descrito em seu nome, por exemplo: calcular a área de um retângulo.

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

R: Resumidamente, o acoplamento descrito no texto divide em dois tipos: acoplamento aceitável e o acoplamento ruim. O acoplamento aceitável é aquele que, a classe cliente utiliza apenas métodos públicos da classe provedora e estável do ponto de vista sintático e semântico. O acoplamento ruim, em poucas palavras, é o que acessa diretamente propriedades de outra classe, compartilham variáveis globais, e os métodos são frequentemente renomeados para que atendam necessidades de outra classe.

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