



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP21º semestre de 2015.

1. Explique porque é importante evitar a duplicação de código no desenvolvimento de software. (valor: 2,0 pontos)

Criar código duplicado em duas rotinas indica um erro de decomposição do programa, pois o código duplicado deveria ter sido isolado e movido para uma rotina a parte. Esta separação reduz a memória ocupada pelo programa, facilita a correção de erros, a otimização do código e a manutenção do sistema, pois um erro encontrado nesta parte do código não precisaria ser resolvido em duas cópias.

2. Explique a relação entre as fases de planejamento, execução e controle no gerenciamento de projetos de software. (valor: 2,0 pontos)

O planejamento implica na definição, revisão e manutenção de uma organização de trabalho para a realização do projeto. Esta etapa define que trabalho será feito no projeto, como será distribuído ao longo do tempo, quem fará cada trabalho e a que custo, documentando todas estas informações em um plano de projeto. A execução implica na coordenação das pessoas e recursos necessários para a execução do plano de projeto. Por fim, o controle implica no monitoramento e medição do progresso para garantir que os objetivos do projeto serão atingidos.

3. Na notação UML, uma classe é representada por um retângulo dividido em 3 regiões. Que informações colocamos em cada região da classe? Como estas informações aparecem no Diagrama de Sequência? Apresente um exemplo relacionando os dois tipos de diagrama. (valor: 2,0 pontos).

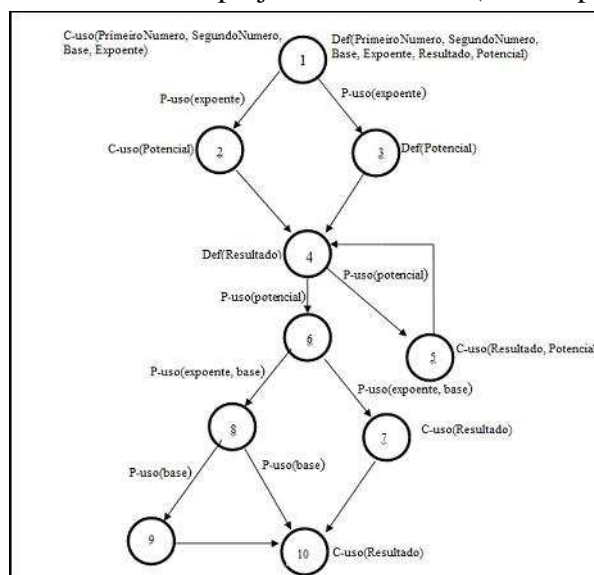
No topo é apresentado o título da classe. Na região central, são apresentados os seus atributos. Na região inferior, são apresentados os seus métodos. O nome da classe tipicamente é apresentado no topo dos diagramas de sequência, usualmente acompanhado do nome de um objeto. Os métodos da classe são apresentados como setas partindo de sua linha do tempo.

4. Explique as diferenças entre teste e depuração, indicando os principais objetivos de cada um. Em complemento, liste no mínimo 3 princípios associados aos testes de software. (Valor: 2,0 pontos)

Teste de Software é o processo de executar um programa ou sistema de software com o objetivo de revelar a presença de defeitos; ou, falhando nesse objetivo, aumentar a confiança sobre o programa. Teste de software revela falhas. Portanto, seu objetivo é refutar a assertiva de que o produto está correto. Princípios: devem ser rastreáveis aos requisitos do usuário; devem ser completamente planejados antes de sua execução; o princípio de Pareto se aplica (80% dos defeitos vem de 20% dos módulos); devem ser aplicados inicialmente em pequena escala e depois expandidos; não é possível sua aplicação de maneira exaustiva; para aumentar sua eficácia deve ser planejado e executado por equipes independentes.

Depuração é consequência do teste. Consiste em encontrar o defeito (falha) que provocou a falha. Portanto, depuração não é teste.

5. Calcule a métrica *Complexidade Ciclomática* para o grafo de fluxo de controle abaixo e indique como seu valor pode ser usado para apoiar a tomada de decisão no gerenciamento e desenvolvimento do projeto de software?(valor 2 pontos) .



Seu calculo se dá através da identificação do número de regiões no grafo de fluxo do programa, através da formula

$$V(G)=E-N+2$$

E: número de arcos

N: número de nós

ou, $V(G) = P + 1$

P: número de nós predicados (decisões)

Neste caso, o valor da Métrica é 5 (13 arcos, 10 Nós) ou (4 Nós Predicados: 1, 4, 6 e 8)

Esta métrica que fornece uma medida quantitativa da complexidade lógica de um programa. No contexto do teste estrutural, seu valor define o número de caminhos independentes e nos fornece o número máximo de casos de teste que garantem que todos os comandos tenham sido executados pelo menos uma vez.