

Nome –

Assinatura –

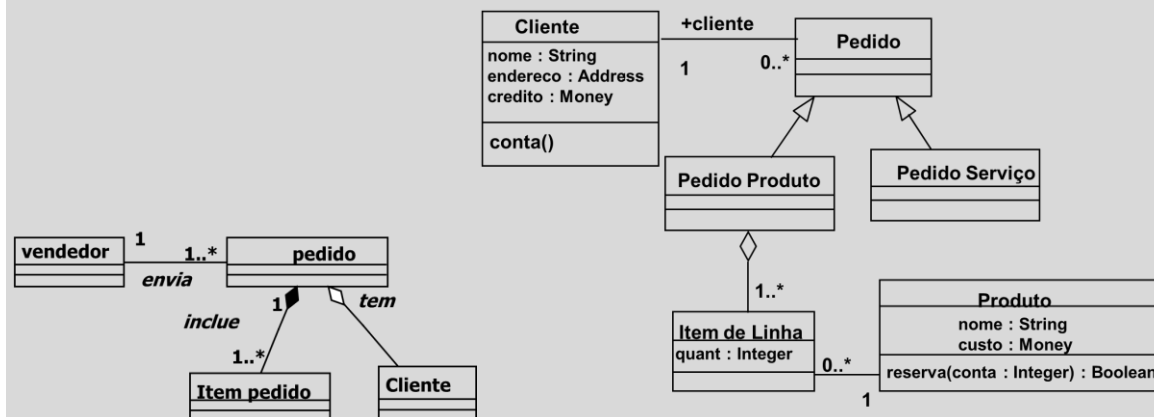
Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

1. Quais são os componentes de um diagrama de classes da UML? Explique como estes componentes se relacionam e o que cada um representa no diagrama. Apresente um exemplo contendo todos os componentes relacionados na sua resposta (Valor 3,0 pontos; máximo: 10 linhas)

Um diagrama de classes da UML representa as classes/objetos e seus relacionamentos. Compõem este diagrama as classes, relacionamentos de herança, associação, agregação e composição. A indicação de cardinalidade e navegabilidade dos relacionamentos trazem indicação importante sobre a organização estrutural do modelo, que representa a visão estática no mundo de objetos. Os relacionamentos de herança permitem organizar hierarquicamente as classes, classificando-as de acordo com seu nível de abstração da descrição dos objetos. Os relacionamentos de associação indicam as dependências entre classes/objetos enquanto os relacionamentos de agregação e composição permitem a estruturação de objetos complexos e compostos de diferentes partes.

Modelos apresentando os diferentes componentes e suas relações podem ser vistos a seguir:



2. Dizemos que um projeto de software deve ser gerenciado em cinco grandes etapas: inicialização, planejamento, execução, controle e fechamento. Explique o papel de cada uma destas etapas no gerenciamento de projetos de software. (valor: 0,0 pontos; máximo: 15 linhas)

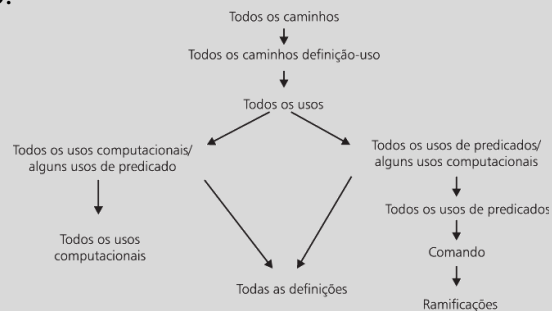
QUESTÃO ANULADA DEVIDO A INCONSISTENCIA DE CONTEUDO FACE A ANTECIPAÇÃO DA AP2 EM RELAÇÃO A PERIODOS ANTERIORES. PONTOS REDISTRIBUIDOS ENTRE AS OUTRAS QUESTOES.

3. Para as métricas a-e abaixo, indique seu nível de precisão (objetiva, subjetiva), escala (nominal, ordinal, intervalar, razão) e objeto de medição (processo, produto). Informe uma situação de projeto de software que a variação dos seus valores pode ajudar a observar. (Valor: 0,0 pontos; máximo 10 linhas)

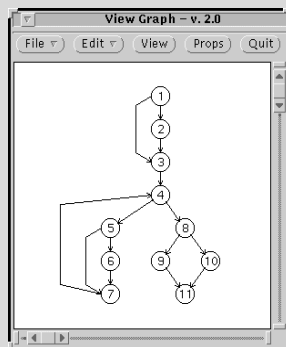
QUESTÃO ANULADA DEVIDO A INCONSISTENCIA DE CONTEUDO FACE A ANTECIPAÇÃO DA AP2 EM RELAÇÃO A PERIODOS ANTERIORES.. PONTOS REDISTRIBUIDOS ENTRE AS OUTRAS QUESTOES.

4. Explique o que é teste estrutural. Qual a diferença para teste funcional? Identifique três critérios de teste que podem ser utilizados para projetar casos de teste estrutural. Escolha um deles e explique como usá-lo para projetar os casos de teste (valor: 3,0 pontos)

Casos de teste estrutural, ou testes caixa-branca (ou caixa aberta), exploram a estrutura interna do software para apoiar o projeto dos casos de teste. A maior diferença para os testes funcionais está na forma como o software é tratado, pois não se consideram as funcionalidades do software mas sim sua estrutura de construção. Diferentes critérios podem ser usados para apoiar a geração de testes estruturais, conforme pode ser visto no modelo abaixo:



Usando o Critério Todos os caminhos para o software representado pelo grafo de programa abaixo, teríamos os seguintes casos de teste: <1,2,3,4,5,6,7,4,8,9,11>, <1,3,4,5,7,4,8,10,11>, <1,2,3,4,8,9,11>,<1,3,4,10,11>



5. Marque Verdadeiro (V) ou Falso (F) para as afirmações abaixo (valor até 4,0 pontos – 0,4 por marcação correta. Atenção: cada 2 erros/branco eliminam 1 acerto!):
- (F) A) Inspeções de Software aumentam significativamente a produtividade, porém não afetam a qualidade e a estabilidade dos projetos de software.
 - (F) B) Remoção de Defeitos é tão bom quanto prevenir defeitos.
 - (F) C) O custo para correção de um defeito não muda durante o ciclo de vida do software.
 - (V) D) Testes podem mostrar a presença, mas não a ausência de defeitos.
 - (F) E) O melhor testador do código é o próprio desenvolvedor.
 - (V) F) Os defeitos são usualmente gerados na comunicação e na transformação das informações ao longo do projeto do software
 - (F) G) Se combinarmos Inspeções de Software e Testes de Software podemos capturar todos os defeitos no software.
 - (F) H) Inspeções *ad-hoc* possuem custo e resultados semelhantes as inspeções com *checklists*.
 - (F) I) Não importa o perfil do inspetor, a eficiência da inspeção será sempre a mesma.
 - (V) J) As técnicas de leitura (inspeção) garantem a cobertura do documento como um todo.

Boa Prova!