



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina de Arquitetura e Projeto de Sistemas II**  
**Gabarito - AP1 1º semestre de 2009.**

**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.
- 

Questão 1 (2 pontos)

Em relação ao diagrama de classes UML,

- (a) Explique com suas palavras a diferença entre os tipos de relacionamento entre classes (associação, agregação, composição e generalização)?
- (b) Cite três dos modificadores de visibilidade de métodos existentes e explique com suas palavras para que servem?

**Resposta:**

- (a) Associação simples indica que duas classes colaboram, ambas com o mesmo nível de importância. Agregação é uma associação onde uma das partes exerce o papel de “todo” e a outra parte exerce o papel de “parte”. Composição é uma variação de agregação que introduz mais força na relação todo-parte, onde a “parte” pertence a somente um “todo” e tem o seu ciclo de vida atrelado a ele. Generalização é um relacionamento entre tipos que pertencem a uma mesma hierarquia de herança, onde um é mais genérico e outro é mais específico.
- (b) Privado, onde somente métodos da própria classe podem acessar o método em questão; Protegido, onde métodos de outras classes que herdaram da classe em questão podem acessar o método em questão; e Público, onde qualquer método de qualquer classe pode acessar o método em questão.

### Questão 2 (3 pontos)

Em relação ao tópico de “domínios” em princípios de projeto,

- (a) Explique com suas palavras por que é desejável que uma classe pertença a somente um domínio (base, arquitetura, negócio ou aplicação).
- (b) Cite um exemplo concreto de uma classe que mistura domínio de negócio com domínio de arquitetura e liste possíveis desvantagens.
- (c) Seria interessante reutilizar classes do domínio de aplicação? Justifique sua resposta.

Resposta:

- (a) Caso uma classe pertença a mais de um domínio, ela possivelmente terá baixa coesão e será de difícil reutilização em aplicações futuras. Desta forma, o ideal é que toda classe pertença a um único domínio.
- (b) Um exemplo seria uma classe Fatura com métodos para envio por diferentes meios de comunicação. Neste caso, a Fatura deixa de ser genérica e perde coesão por adicionar métodos que não pertencem ao seu domínio. Além disso, caso se deseje enviar outro tipo de documento pelos meios de comunicação usados em Fatura, não será possível, pois o código de envio está diretamente acoplado com a classe Fatura.
- (c) Não. Classes do domínio de aplicação são criadas exatamente com o intuito de adaptar outras classes ou servir de cola entre classes para possibilitar a construção de uma aplicação específica. Por essa razão, elas não são projetadas para serem reutilizadas.

### Questão 3 (2 pontos)

Com relação à modelagem orientada a objetos:

- (a) O que são atores e como se relacionam a casos de uso?
- (b) Como casos de uso e atores são identificados?

Resposta:

- (a) Atores representam qualquer elemento externo que possa interagir direta ou indiretamente com o sistema. Podem ser outros sistemas, seres humanos ou dispositivos de hardware. Cada ator tem interesse em um conjunto de

operações/funções. Um caso de uso apresenta uma operação/função de interesse de um ator.

- (b) Casos de uso e atores são identificados a partir de documentos que descrevem domínios de aplicação e/ou requisitos do sistema, e de entrevistas com especialistas do domínio e/ou clientes do sistema.

Questão 4 (3 pontos)

Com relação ao desenvolvimento de software em geral:

- (a) Qual a importância da modelagem para o desenvolvimento de software?
- (b) Quais são as dificuldades da Engenharia de Requisitos?
- (c) Como o desenvolvimento orientado a objetos se difere do desenvolvimento tradicional?

Resposta:

- (a) Modelos são úteis para o entendimento de problemas, a difusão do conhecimento entre os desenvolvedores e a possibilidade de testar hipóteses antes de realizá-las. Eles são criados devido à limitação do ser humano em lidar com a complexidade.
- (b) Algumas das dificuldades são: a comunicação com o usuário, mudanças constantes, previsão de situações futuras, entendimento completo do problema e diversas formas de representação, dentre outras.
- (c) O desenvolvimento orientado a objetos se caracteriza por ser centrado em dados e processos, diferentemente do desenvolvimento tradicional que 'é centrado em processos ou dados de forma excludente. E' também caracterizado por um desenvolvimento interativo incremental, diferentemente do desenvolvimento tradicional linear-sequencial.