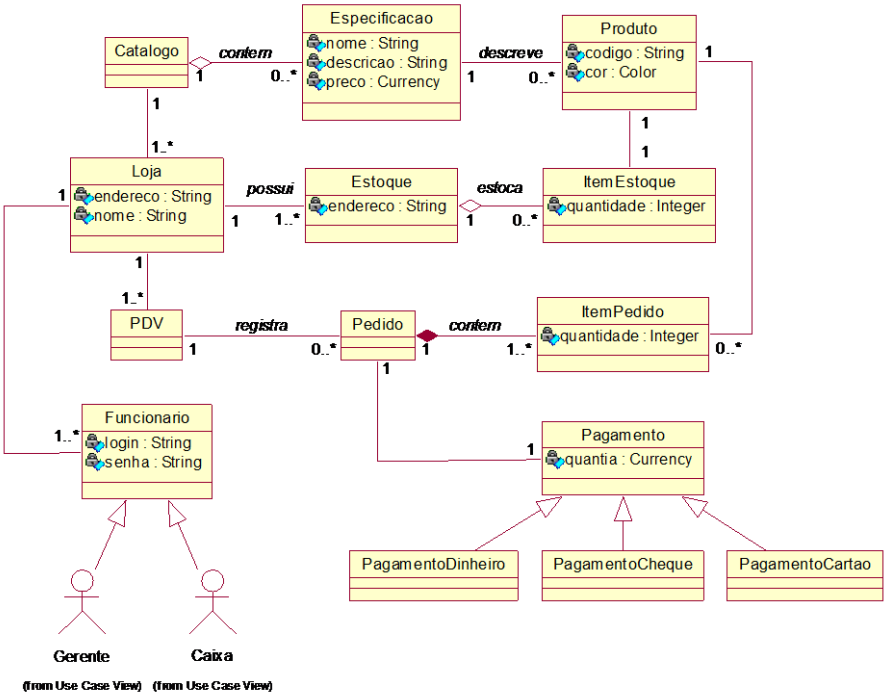
Fundação CECIERJ - **Vice Presidência de Educação Superior a Distância**

Curso de Tecnologia em Sistemas de Computação
Disciplina de Arquitetura e Projeto de Sistemas II
Gabarito – AP1 2° semestre de 2007.

Questão 1 (3 pontos)

Considere o seguinte diagrama de classes conceitual:



(a) Determine o grau de dependência direto e indireto para a classe "ItemPedido" em relação às demais classes do diagrama (Lembrete: a navegabilidade nos casos de todo-parte é sempre no sentido do todo para a parte. Ou seja, em situações de composição ou agregação, a parte não conhece o todo).

(b) Assumindo que o espaço-estado do atributo "quantia" da classe "Pagamento" seja de R$ 0 a R$ 1000,00, é permitido que a classe "PagamentoCheque" modifique esse espaço-estado para de R$ 50,00 a R$ 500,00? Justifique a sua resposta.

(c) Para que serve o mecanismo de invariante de classe? Descreva as invariantes da classe "ItemPedido".

Resposta:

(a) Grau de dependência **direto** para a classe ItemPedido: 1 (classe Produto). Grau de dependência **indireto** para a classe ItemPedido: 3 (classes Produto, Especificação e ItemEstoque).

(b) Sim. Tendo em vista que todo PagamentoCheque *é um* Pagamento, e que a variação de espaço-estado em casos de herança deve ser mais restritiva nas subclasses, a modificação proposta é válida. Ou seja, o intervalo entre 50 e 500 está dentro do intervalo entre 0 e 1000, como era de se esperar pois o PagamentoCheque *é um* Pagamento.

(c) O mecanismo de invariante de classe serve para garantir que as restrições de espaço-estado sejam respeitadas. As invariantes de classe devem ser válidas para todos os objetos em equilíbrio (sem método em execução) da classe. As invariantes da classe ItemPedido são: (1) o atributo quantidade ser maior que zero; (2) existir um produto associado; e (3) a quantidade em estoque desse produto ser maior ou igual ao atributo quantidade.

Questão 2 (3 pontos)

Considere um sistema de controle bancário onde a classe "Conta" tem um método que é responsável pela transferência de dinheiro para outra conta corrente: "transfere(valor, contaDestino)". Neste cenário, somente transferências de valores menores que R$ 5000,00 podem ser feitas.

(a) Defina as pré-condições e pós-condições do método "transfere(valor, contaDestino)".

(b) Seria correto criar uma classe que herde da classe "Conta" e somente permita transferências de até R$ 4000,00? Justifique a sua resposta.

(c) Se, além do método "transfere(valor, contaDestino)", existissem também os métodos "deposita(valor)" e "saca(valor)", a interface da classe "Conta" poderia ser considerada uma interface com comportamento replicado? Justifique a sua resposta.

Resposta:

QUESTÃO ANULADA

Questão 3 (2 pontos)

Em relação ao processo unificado, defina e exemplifique:

(a) Desenvolvimento iterativo

(b) Desenvolvimento evolutivo

(c) Desenvolvimento ágil

(d) Fases (concepção, elaboração, construção e transição)

Resposta:

(a) O **desenvolvimento iterativo** consiste em organizar o desenvolvimento em "mini-projetos" – um a cada iteração –, de duração curta e fixa, com atividades de análise, projeto, programação e testes e cujo produto de cada iteração é um software parcial. Exemplo: criar iterações de 4 semanas e atribuir alguns casos de uso para serem desenvolvidos em cada iteração.

(b) No **desenvolvimento evolutivo**, as especificações evoluem a cada iteração, com a construção de uma parte de software, de forma que o conhecimento sobre o software aumente. Exemplo: ao término de cada iteração, apresentar o software funcional ao usuário e evoluir a especificação em função da percepção do usuário ao utilizar o software.

(c) o **desenvolvimento ágil** fundamenta-se em respostas rápidas e flexíveis a mudanças, com replanejamento contínuo do projeto e entregas incrementais e constantes do software (refletindo tais mudanças). Exemplo: Ao apresentar o software produzido na iteração para o usuário, acolher prontamente as modificações necessárias, replanejando o projeto para que essas modificações sejam incorporadas ao software na próxima iteração.

(d) As fases do processo unificado têm o intuito de retratar a ênfase principal das iterações. A **fase de concepção** é a menor fase do projeto e tem escopo e estimativas ainda vagas. Exemplo: 1 mês com início da análise e algum estudo de possibilidades de projeto. A **fase de elaboração** é responsável pela conclusão de grande parte das atividades de análise e projeto, diminuição significativa das incertezas e da criação da *baseline* da arquitetura. Exemplo: 3 meses para a finalização da análise e projeto de alto nível. A **fase de construção** é a maior fase do projeto, responsável pelo estabelecimento das *baselines* de testes do produto. Exemplo: Em 4 meses, o projeto será detalhado e implementado. Finalmente, a **fase de transição** é responsável pelo estabelecimento da *baseline* de liberação do produto. Exemplo: No último mês, o código será testado e preparado para ser entregue para o cliente.

Questão 4 (2 pontos)

Em relação à orientação a objetos, defina e exemplifique:

(a) Classe

(b) Objeto

(c) Herança

(d) Polimorfismo

Resposta:

(a) **Classe**: representação computacional de entidades ou processos do mundo real. São compostas de atributos (características – informações) e métodos (comportamentos – processos) e instanciam objetos. Exemplo: classe Gerente, com atributos nome e idade e métodos calculaSalario() e getIdade()

(b) **Objeto**: instanciação de uma classe. Possui um conjunto de serviços (interface) e sua implementação (estruturas de dados – atributos, e implementação de operações – métodos). Exemplo: objeto da classe Gerente com nome = "João" e idade = 25.

(c) **Herança**: mecanismo que promove a reutilização de software por meio do reconhecimento da similaridade entre classes de objetos, formando uma hierarquia. Define uma relação do tipo "é um", onde uma classe compartilha a estrutura e o comportamento definidos em uma ou mais classes. Exemplo: classe Gerente herda da classe Funcionario.

(d) **Polimorfismo**: propriedade derivada do fato de que objetos de diferentes classes podem reagir a uma mesma mensagem de forma diferente. Dessa forma, cada classe implementa um método específico para uma operação, possibilitando a definição de protocolos comuns. Exemplo: método getSalario em Gerente leva em consideração a idade do gerente, enquanto em funcionário somente o tempo de trabalho é considerado.