



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 – AD1 2º semestre de 2007.

Questão 1 [1 ponto]

Conceitue Processo Unificado e explique seus três aspectos constituintes. Faça a ligação de cada um deles com sua importância prática do ponto de vista do cliente.

Resposta:

Processo Unificado é um processo de software que define um conjunto de atividades necessárias para identificar os requisitos do usuário e transformá-los em um sistema de software, caracterizado por ser dirigido a casos de uso, centrado na arquitetura, iterativo e incremental. Os três aspectos a serem considerados neste processo de software são o desenvolvimento iterativo, o desenvolvimento evolutivo e o desenvolvimento ágil.

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. Possibilita, dessa forma, a verificação constante pelo cliente dos requisitos implementados, com aumento de conhecimento sobre o software e redução de incertezas perante mudanças.

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. Diante de um cenário de mudanças (requisitos, ambiente e pessoas), a abordagem evolutiva tenta evitar correções, retrabalho e implementações de especificações “desnecessárias” do cliente (requisitos que não são realmente úteis ao software).

Já 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). Em relação ao cliente, basta verificar que um dos pontos do manifesto ágil é: “colaboração do cliente vem antes de negociação de contrato”.

Questão 2 [0,5 ponto]

Liste as principais características de cada uma das quatro fases do Processo Unificado.

Resposta:

Fase de Concepção: (i) menor fase do projeto; (ii) escopo ainda vago; e (iii) estimativas ainda vagas.

Fase de Elaboração: (i) grande parte das atividades de análise e projeto concluídas; (ii) diminuição significativa das incertezas; e (iii) *baseline* da arquitetura estabelecida.

Fase de Construção: (i) maior fase do projeto; e (ii) *baseline* de testes do produto estabelecida.

Fase de Transição: (i) *baseline* de liberação do produto estabelecida.

Questão 3 [1 ponto]

Sobre conceitos de Orientação a Objetos defina:

- (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.
- (b) Objeto: instanciamento de uma classe. Possui um conjunto de serviços (interface) e sua implementação (estruturas de dados – atributos, e implementação de operações – métodos).
- (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.
- (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.

Questão 4 [0,5 ponto]

Qual a importância da modelagem para o desenvolvimento de software? Relacione, em sua resposta, implicações de modelagem na complexidade do software.

Resposta:

Por meio de modelos, a modelagem aplicada ao desenvolvimento de software auxilia: (i) no entendimento de problemas e na compreensão dos requisitos; (ii) na difusão do conhecimento e na comunicação entre os membros da equipe de projeto (linguagem homogênea entre diferentes especialistas); e (iii) no teste de hipóteses antes de realizá-las. Além disso, permite expressar diferentes visões do sistema. A modelagem contribui, dessa forma, para a redução da complexidade do software ao atuar no âmbito da limitação humana – lidar com cenários de grande escala –, tanto em nível de domínio de aplicação, quanto ao longo do processo de desenvolvimento.

Questão 5 [0,5 ponto]

Em relação à Engenharia de Requisitos:

(a) explique três de suas dificuldades, enfatizando possíveis impactos sobre o cliente (quando da utilização do software);

(b) pesquise e discorra acerca de três técnicas de elicitação de requisitos.

Resposta:

(a) Dificuldades: (i) Comunicação com usuário: problemas de comunicação, ocasionados por ruídos no canal (meio) ou mesmo pela falta de compreensão do código utilizado (linguagem), são significativos para o entendimento dos requisitos a serem contemplados pelo projeto do sistema, o que pode resultar na agregação de funcionalidades erradas ou mesmo desnecessárias ao produto final; (ii) Mudanças constantes: solicitações realizadas pelos clientes são passíveis de alterações frequentes, devido a mudanças gerenciais, de lógica de negócio, do arcabouço tecnológico, dentre outras, e isso requer uma postura flexível e dinâmica dos engenheiros de software; e (iii) Diferentes formas de representação: a falta de padronização de uso de uma forma para se representar o problema a ser resolvido e os requisitos a serem contemplados pelo sistema podem gerar distorções entre desenvolvedores e gerentes e entre gerentes e clientes, ocasionando tensões desnecessárias, por falta de planejamento.

(b) Técnicas: (i) Entrevista: é uma técnica simples, muito conhecida e usada, mas que deve ser conduzida de forma estruturada a fim de evitar se tornar muito informal – menos produtiva – e consistir apenas em uma conversa. Engenheiros de requisitos devem elaborar formulários com questões direcionadas para organizar as informações que o cliente fornecerá, sem perder detalhes importantes, além de buscar evitar retrabalho ou esforços futuros desnecessário; (ii) Reuniões de Brainstorming: consistem em reuniões de trocas de idéias onde várias questões são levantadas e colocadas em pauta, começando do nível mais geral e prosseguindo-se para um detalhamento mais específico de idéias sobre a construção do sistema; e (iii) Prototipação: é uma técnica prática e bastante utilizada, na qual um entendimento inicial do sistema pode permitir a construção de um pequeno protótipo do sistema, a fim de exibir ao cliente o que foi assimilado *a priori* pelo engenheiro de requisitos acerca do que vai ser o sistema.

Questão 6 [0,5 ponto]

Quais são os diagramas da UML que compõem a análise orientada a objetos? Escolha um deles e descreva seus principais elementos constituintes.

Resposta:

Os diagramas da UML que compõem a análise orientada a objetos são: Diagrama de Casos de Uso, Diagrama de Classes e Diagrama de Seqüência. Considerando o Diagrama de Casos de Uso, seus principais elementos componentes são: (i) caso de uso (*use case*): apresenta uma funcionalidade do sistema que provê ao usuário (ator) um resultado observável. Representa um conjunto de ações que o sistema deve realizar para atender a uma funcionalidade. Serve de base para comunicação entre clientes e desenvolvedores e para guiar o processo de desenvolvimento; (ii) cenário: consiste de uma seqüência específica de ações e interações entre atores e o sistema. É uma instância de caso de uso e representa uma história particular de uso do sistema; (iii) atores: representam qualquer elemento externo que possa interagir direta ou indiretamente com o sistema. Podem ser outros sistemas, seres humanos ou dispositivos de hardware; e (iv) relações entre casos de uso: são os relacionamentos entre funções do sistema, unidirecionais e representadas por uma seta. Podem ser de inclusão (quando diversos casos de uso podem compartilhar o mesmo comportamento e este é, então, separado em um caso de uso específico, estabelecendo-se uma relação) ou de generalização e extensão (indica comportamento opcional/alternativo dependente de condição – ponto de variação).

Questão 7 [1 ponto]

Quais as diferenças entre análise e projeto de software?

Resposta:

A análise se concentra na descrição do problema, e seu objetivo deve ser “fazer o sistema certo”, ao passo que o projeto consiste na descrição da solução, visando “fazer certo o sistema”. Em suma, o projeto representa a atribuição de responsabilidades para solucionar o problema descrito na análise, respeitando os requisitos funcionais e não-funcionais do produto.

Questão 8 [0,5 ponto]

Explique o que é Projeto Arquitetural. Cite um exemplo diferente daquele apresentado na disciplina.

Resposta:

Projeto Arquitetural consiste no primeiro passo da fase de projeto, em busca da definição da arquitetura a ser utilizada no sistema a ser desenvolvido. Define a arquitetura global do sistema em termos de subsistemas, interconectados através de dados e fluxos de controle. A arquitetura de um sistema constitui um modelo da estrutura relativamente

pequeno e intelectualmente gerenciável, que descreve como os componentes do sistema trabalham juntos. Um exemplo de arquitetura é a arquitetura cliente-servidor, que permite a distribuição do acesso ao sistema por meio da existência de vários clientes acessando a um servidor central. Um projeto arquitetural segundo essa arquitetura poderia definir o banco de dados como residente no componente servidor da arquitetura e as regras de negócio como residentes no componente cliente da arquitetura.

Questão 9 [0,5 ponto]

Em projeto de software, quais os elementos que podem compor a descrição de casos de uso concretos? Explique cada um deles.

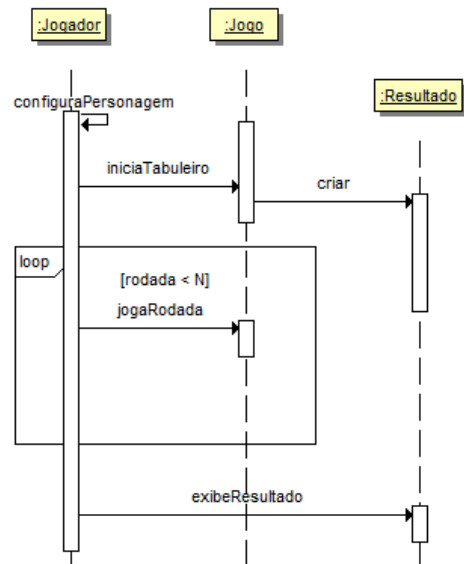
Resposta:

A descrição de casos de uso concretos – um dos artefatos do projeto – refina a descrição de casos de uso essenciais, incluindo: (i) maquete de interface com o usuário com a identificação das *widgets*; (ii) utilização das *widgets* identificadas durante a descrição da seqüência típica de eventos; e (iii) atualização da referência cruzada, descrevendo o caso de uso essencial que serviu como base.

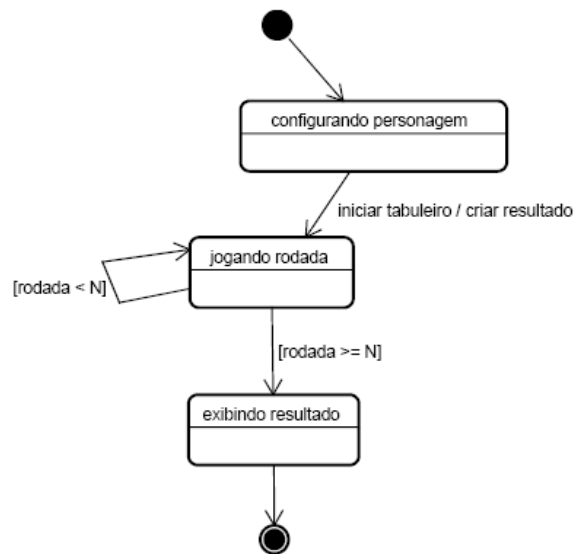
A técnica de maquetes é útil no refinamento de casos de uso. Uma maquete pode ser vista como um desenho inicial das telas do sistema e podem ser desenhadas utilizando tecnologia diferente da que será utilizada pelo sistema. O objetivo é validar as funcionalidades, e não o *design* propriamente dito. Além disso, todas as *widgets* identificadas na descrição dos casos de uso essenciais (análise) são utilizadas para verificar sua aplicação e necessidade e os relacionamentos entre casos de uso são explicitados pelas referências cruzadas atualizadas.

Questão 10 [0,5 ponto]

A partir do diagrama de seqüência abaixo, construa um diagrama de transição de estados para a classe Jogo.



Resposta:



Questão 11 [0,5 ponto]

Defina encapsulamento e apresente seus níveis, exemplificando cada um deles.

Resposta:

Encapsulamento é um mecanismo utilizado para lidar com o aumento de complexidade, buscando exibir “o que” pode ser feito sem informar “como” é feito e permitindo que a granularidade de abstração do sistema seja alterada, criando estruturas mais abstratas. Foi concebido para proteger o acesso direto (referência) aos atributos de uma instância fora da classe onde estes foram declarados. Esta proteção consiste em se usar modificadores de acesso mais restritivos sobre os atributos definidos na classe; depois, deve-se criar métodos com acessibilidade máxima para manipular os atributos da classe.

Os níveis de encapsulamento são: (i) nível 0: completa inexistência de encapsulamento (e.g., linhas de código efetuando todas as ações); (ii) nível 1: módulo procedimentais (e.g., procedimentos permitindo a criação de ações complexas); (iii) nível 2: classes de objetos (e.g., métodos isolando o acesso às características da classe); (iv) nível 3: pacotes de classes (e.g., conjunto de classes agrupadas, permitindo acesso diferenciado entre elas); e (v) nível 4: componentes (e.g., interfaces providas e requeridas para fornecer funcionalidades complexas).

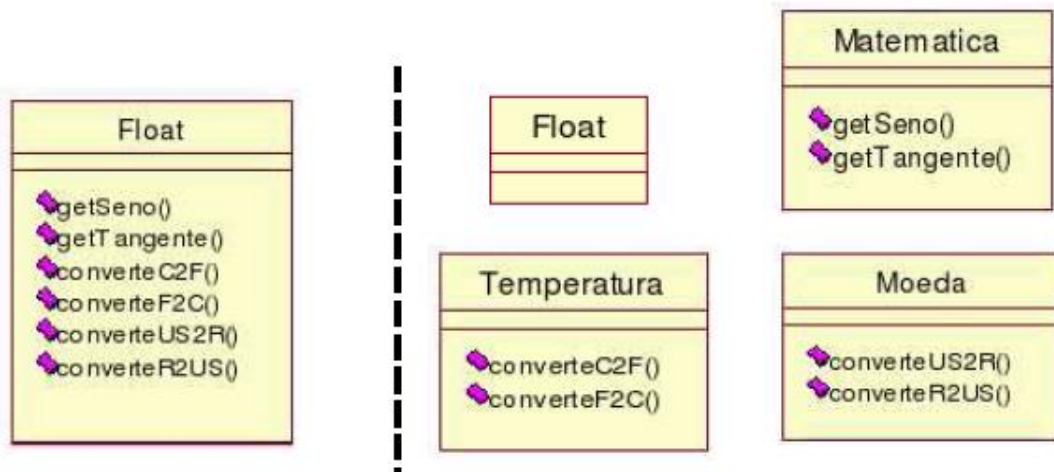
Questão 12 [0,5 ponto]

A coesão pode ser classificada em algumas categorias. Cite tais categorias e ilustre duas delas.

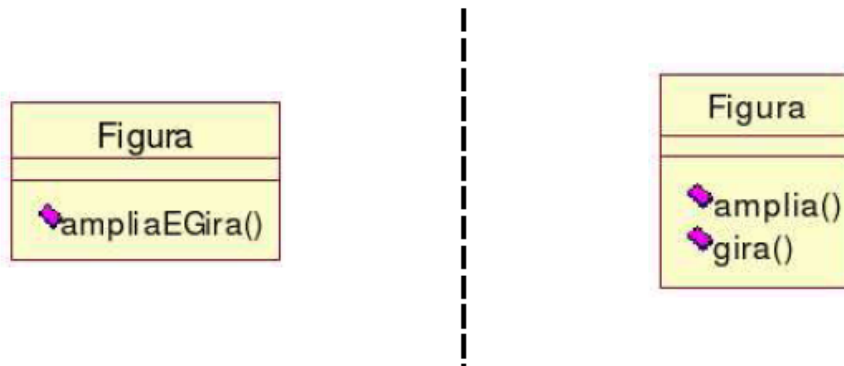
Resposta:

As categorias da coesão são: (i) coesão de instância mista; (ii) coesão de domínio misto; (iii) coesão de papel misto; (iv) coesão alternada; (v) coesão múltipla; e (vi) coesão funcional.

Um exemplo clássico de coesão de domínio misto é a inserção de métodos de manipulação numérica em uma classe que representa números reais (`Float`). A solução para esse problema é a separação das responsabilidades em classes de diferentes domínios, tirando a sobrecarga da classe `Float`.



Na coesão múltipla, um exemplo consiste em um método `ampliaEGira(int escala, int angulo)` em `Figura`, onde esse método faz mais de uma funcionalidade ao mesmo tempo. A solução para esse problema é dividir o método em dois métodos separados.



Questão 13 [0,5 ponto]

De acordo com os princípios de Projeto OO, você considera aceitável a criação de pré e pós condições em métodos abstratos de uma classe? E quanto aos métodos de uma interface?

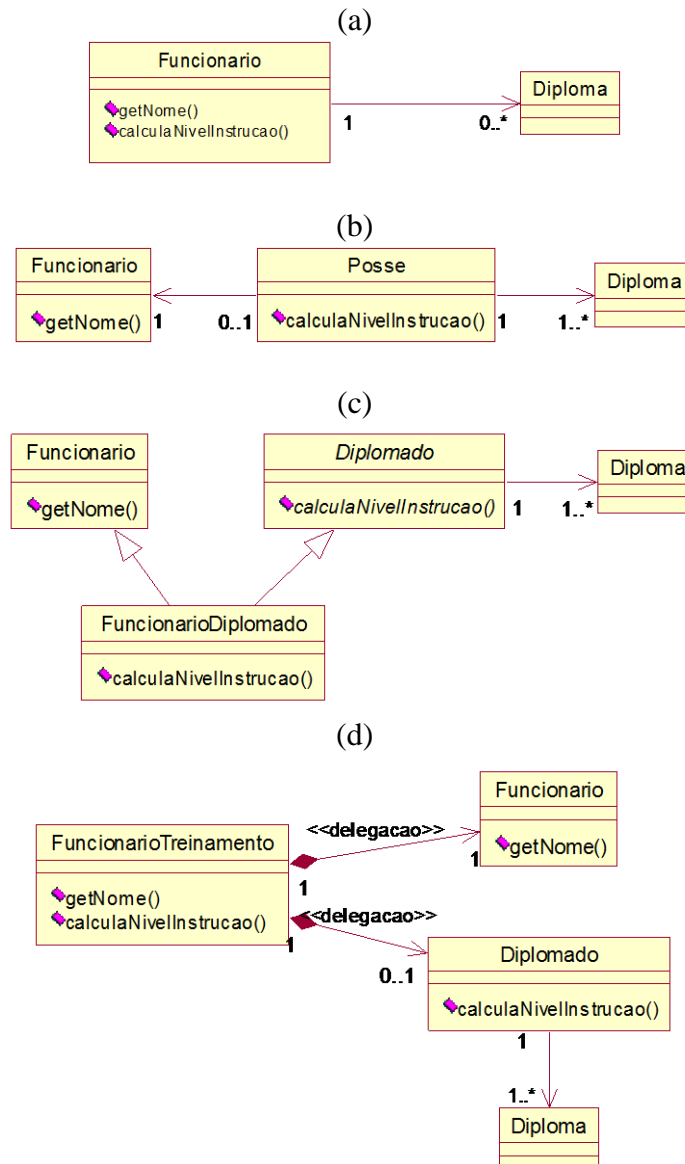
Resposta:

QUESTÃO ANULADA

Questão 14 [1 ponto]

Cenário do problema: O sistema de controle de treinamentos de uma empresa precisa identificar, para cada funcionário da empresa, toda a sua formação até aquele momento, para poder lhe oferecer cursos do seu interesse e condizentes com o seu nível de instrução. Para isso, existe um método chamado `calculaNivelInstrucao()` que fornece o

nível de instrução de um objeto da classe Funcionario (que é utilizada em outros sistemas da empresa) em função de objetos da classe Diploma, que representa os diplomas ou certificados obtidos pelo funcionário durante a sua carreira profissional. Analise os 4 modelos abaixo, citando para cada modelo as suas vantagens e desvantagens.



Resposta:

QUESTÃO ANULADA

Questão 15 [0,5 ponto]

Explique o que são heurísticas de projeto e para que servem.

Resposta:

QUESTÃO ANULADA

Questão 16 [0,5 ponto]

Selecione uma heurística de cada tipo (classes e objetos, topologias, relacionamentos, herança e projeto OO físico) e explique com suas palavras a contribuição provida por cada heurística selecionada.

Resposta:

QUESTÃO ANULADA