



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 da AP3 – 1º semestre de 2012

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)

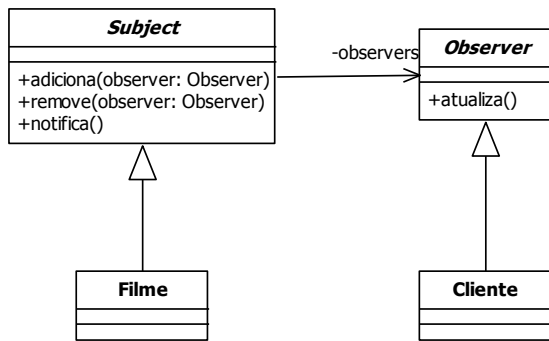
Um dos requisitos de um sistema de locadora determina que “os clientes da locadora podem informar o interesse em filmes que não estão disponíveis para aluguel naquele momento. Assim que os filmes estiverem disponíveis, os clientes interessados devem ser notificados”.

- (a) Responda com suas palavras qual é o padrão GoF mais apropriado para auxiliar na implementação desse requisito? Justifique a sua resposta.

Resposta: Padrão *Observer*. Este padrão tem exatamente o propósito de permitir a criação de dependência de um para muitos via mecanismo de notificação de mudanças.

- (b) Forneça um diagrama de classes que exiba a implementação deste requisito de acordo com o padrão GoF identificado.

Resposta:



Questão 2 (3 pontos)

Relacione cada elemento da coluna da esquerda com um e somente um elemento da coluna da direita.

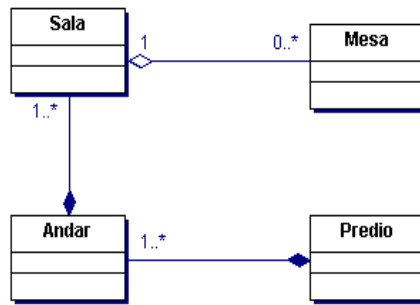
- | | |
|--------------------------------------|---|
| (a) diagrama de classes | (1) Explicita os processos de negócio do cliente. |
| (b) diagrama de casos de uso | (2) Explicita as possibilidades de interação entre os usuários e o sistema. |
| (c) diagrama de transição de estados | (3) Detalha o comportamento de um objeto no decorrer da sua vida. |
| (d) diagrama de sequencia | (4) Detalha uma determinada possibilidade de interação entre o usuário e o sistema. |
| (e) diagrama de atividades | (5) Explicita a estrutura estática interna do sistema. |
| (f) descrição de casos de uso | (6) Detalha a interação entre diferentes objetos do sistema para atender a uma funcionalidade específica. |

Resposta: a → 5; b → 2; c → 3; d → 6; e → 1; f → 4

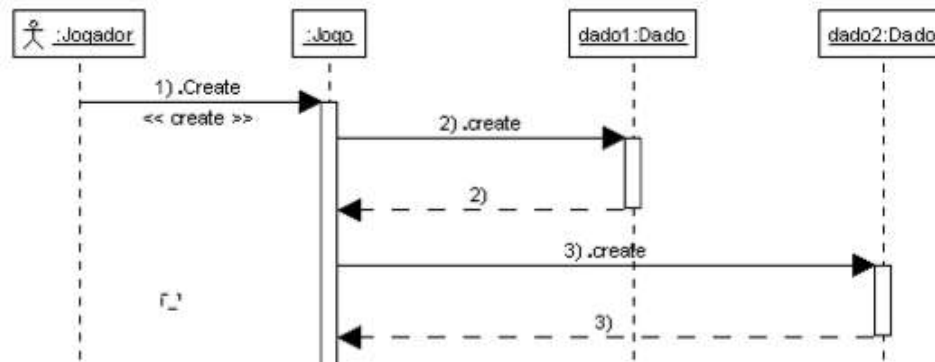
Questão 3 (2 pontos)

Responda com Verdadeiro (V) ou Falso (F) as afirmações abaixo:

- a) () Modelagem não é uma técnica utilizada para a disseminação de conhecimento entre os participantes do processo de desenvolvimento de um sistema de software.
- b) () Na especialização de Frameworks Caixa-Branca o framework recebe um conjunto de parâmetros que representa o comportamento específico da aplicação.
- c) () O Diagrama abaixo representa a interação ator sistema.



- d) () Em Orientação a Objetos, Herança é um conceito que fomenta reutilização através do reaproveitamento das definições da superclasse pela subclasse.
- e) () O Diagrama abaixo dá indícios da existência de uma classe Dado e um ator Jogo



Resposta: a → F; b → F; c → F; d → V; e → F

Questão 4 (3 pontos)

Dado os requisitos abaixo especifique:

- a) (1 Ponto) Diagrama de Caso de uso com no máximo UM ator.
- b) (1 Ponto) Diagrama de Classes com no máximo 4 classes.
- c) (1 Ponto) Mostre como se dá o rastreamento entre os termos presentes nos requisitos e os elementos de modelagem.

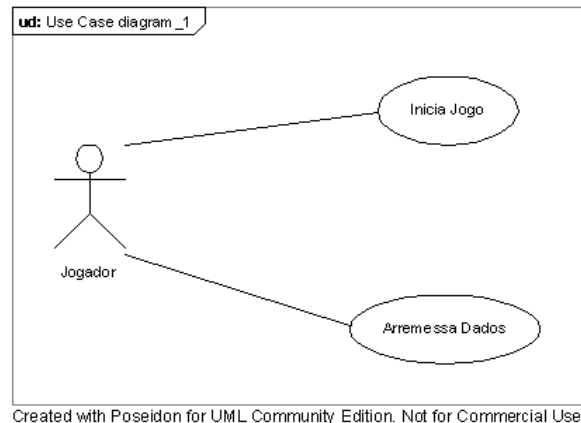
Requisitos:

- a) Este projeto tem como objetivo implementar um sistema que simula um jogo de dados. O jogo começa quando o jogador arremessa dois dados. Caso a soma das faces dos dados seja 7, o jogador é vencedor, caso contrário é perdedor.
- b) Os dados possuem 6 lados e em cada lado há um número de 1 a 6 (sem repetições)
- c) REQ01: O Jogador pode iniciar um jogo novo
- d) REQ02: O Jogador arremessa os dados.

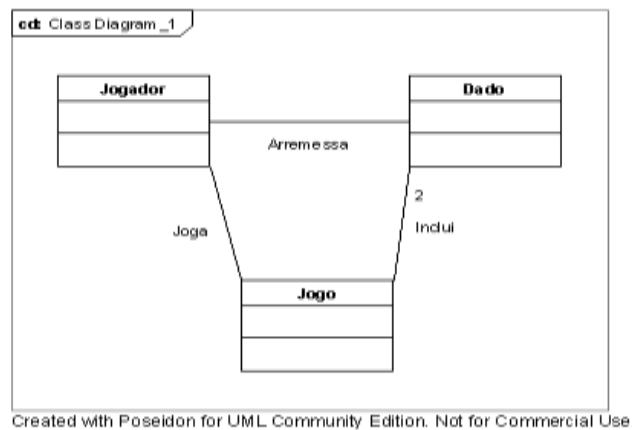
- e) REQ Não Funcionais: O sistema deverá ser desenvolvido utilizando linha de comando, ou seja, não há interface gráfica.

Resposta

a-



b-



c-

Ator Jogador – Presente em Requisitos.a => O jogo começa quando o **jogador** arremessa ...

Caso de Uso Inicia Jogo – Presente em Requisitos.c => REQ01: O Jogador pode **iniciar um jogo**

Caso de Uso Arremessa Dados – Presente em Requisitos.d => REQ02 : O Jogador **arremessa** os dados

Classe Jogador - Presente em Requisitos.a => O jogo começa quando o **jogador** arremessa ...

Classe Jogo - Presente em Requisitos.c => REQ01: O Jogador pode iniciar um **jogo**

Classe Dado - Presente em Requisitos.d => REQ02 : O Jogador arremessa os **dados**