

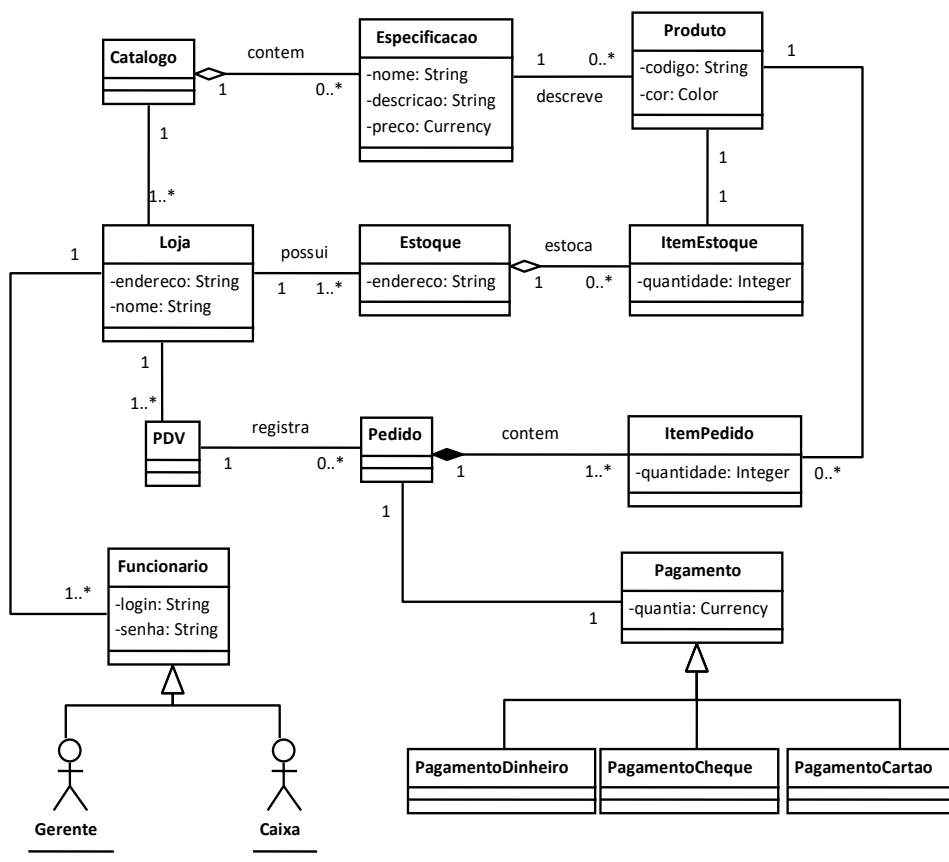
Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular ou celular.
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 (5 pontos): Considere os padrões GRASP e o modelo a seguir.

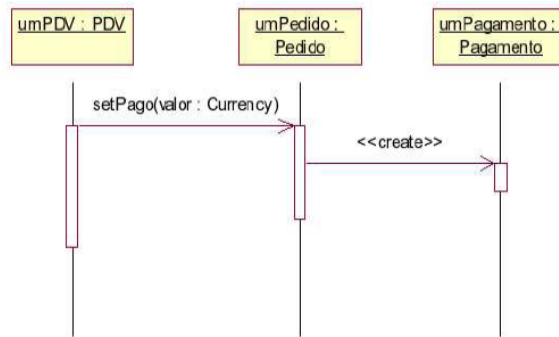


- (a) Responda com suas palavras qual classe deve ser a responsável pela criação de objetos da classe ItemPedido? Justifique a sua resposta citando os padrões utilizados nessa tomada de decisão.

Resposta: De acordo com os padrões *Creator* e *Low Coupling* a classe Pedido deve ser a responsável por criar objetos da classe ItemPedido. Isso ocorre pois a responsabilidade de criação de objetos da classe B deve ser atribuída à classe A se A contém objetos de B, favorecendo assim a manutenção do baixo acoplamento do sistema.

- (b) Forneça um diagrama de sequência que exiba a atribuição de responsabilidades para o pagamento de um pedido, representada pelo método paga(valor : Currency, umPedido : Pedido) a ser criado na classe PDV.

Resposta:



- (c) Responda com suas palavras qual padrão GoF poderia ser adotado na implementação do sistema para que exista somente uma instância de Catalogo para toda a rede de lojas?

Resposta: Padrão *Singleton*. O propósito do padrão *Singleton* é assegurar que uma classe tenha somente uma instância, e fornecer um acesso global a essa instância.

- (d) Qual é a classe com maior grau de dependência direto (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. Além disso, a ausência de seta nas associações representa navegabilidade bidirecional)? Justifique a sua resposta citando todas as suas dependências diretas.

Resposta: Classe Loja. Ela depende diretamente das classes PDV, Funcionario, Catalogo e Estoque.

- (e) 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\$ 500,00 a R\$ 800,00? Justifique a sua resposta.

Resposta: 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

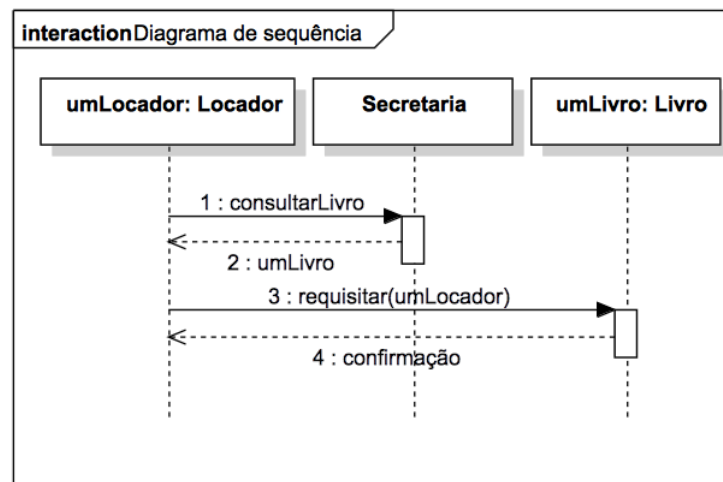
seja, o intervalo entre 500 e 800 está dentro do intervalo entre 0 e 1000, como era de se esperar pois o PagamentoCheque é um Pagamento.

Questão 2 (2 pontos)

Assinale V (verdadeiro) ou F (falso) justificando (em ambos os casos).

- a) () A especificação de requisitos formaliza as necessidades do desenvolvedor.
- b) () Diagramas de classe permitem explicitar relações de herança.
- c) () Diagramas de classe modelam a interação entre objetos.

Considere o diagrama de sequência abaixo, que descreve um caso de uso para requisitar livros no sistema de uma biblioteca.



- d) () Os retângulos no meio das linhas tracejadas verticais indicam que a classe possui métodos e atributos públicos.

Gabarito

- a) (F) A especificação de requisitos formaliza as necessidades do usuário.
- b) (V) Isso é feito a partir da indicação de generalizações.
- c) (F) Isso é feito por diagramas de sequência.
- d) (F) Esses retângulos indicam a ativação da classe invocada pela seta que se destina nesses e encerrada após a resposta identificada pela seta tracejada de origem nos mesmos.

Questão 3 (3 pontos) Sobre o estilo arquitetural Orientado a Objetos, caracterize suas diferentes visões (modelo 4+1).

Gabarito

- (i) Visão lógica: detalham-se os requisitos funcionais do sistema, representada por diagramas de classes.
- (ii) Visão de desenvolvimento: consiste em uma unidade de código (componente), formado por um conjunto de classes que interagem para realizar tarefas.
- (iii) Visão de processos: decompõe o sistema nas tarefas a serem realizadas, explicitando quais processos estão associados a quais componentes, enfocando atributos não funcionais.
- (iv) Visão física: mapeia os processos em unidades de processamento específicas.
- (v) Visão de cenário: unifica as demais visões, funcionando como uma guia ao projeto da arquitetura e servindo como uma forma de validar a arquitetura.