



Lista de Revisão VII

Responda as questões abaixo:

1. Considere o seguinte caso na programação orientada a objetos: em um sistema de gestão de uma biblioteca, temos as classes "ItemBiblioteca" como classe base, "Livro" e "DVD" como classes derivadas. A classe "ItemBiblioteca" contém atributos comuns como "título" e "número de cópias", enquanto as classes derivadas possuem métodos específicos, como "consultarDisponibilidade()" para "Livro" e "reproduzir()" para "DVD".

Com base nesse cenário complexo, qual das seguintes afirmações é verdadeira?

- a) A herança na programação orientada a objetos possibilita que as classes derivadas obtenham tanto atributos quanto métodos da classe base, como exemplificado pelas classes "Livro" e "DVD" em relação à classe "ItemBiblioteca".
- b) A herança é um princípio obsoleto na programação orientada a objetos e deve ser evitada em favor da composição.
- c) Na programação orientada a objetos, a herança é exclusivamente utilizada para transferir métodos entre classes, não influenciando os atributos.
- d) A herança é um conceito meramente teórico na programação orientada a objetos e não tem aplicabilidade prática em sistemas complexos.
- e) A herança na programação orientada a objetos é restrita apenas à transferência de atributos entre classes, não afetando os métodos.



2. A programação orientada a objetos possui uma série de vantagens como Reutilização de Código, Manutenção Simplificada, Consistência entre análise e desenvolvimento. Para que estas vantagens sejam realmente obtidas, o conceito de encapsulamento é essencial, conforme apresenta Cay Horstmann em sua obra *Conceitos de Computação com Java*. Dentro deste contexto, qual é o principal objetivo do encapsulamento?

Referência:

HORSTMANN, Cay. Conceitos de Computação com Java. São Paulo: Grupo A, 2009. E-book. ISBN 9788577804078. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577804078/>. Acesso em: 07 nov. 2023.

- a) Ocultar os detalhes de implementação de uma classe e fornecer uma interface controlada para interagir com seus objetos
- b) Tornar os dados de uma classe acessíveis a qualquer parte do programa
- c) Reduzir o tamanho do código fonte de uma classe
- d) Garantir que uma classe nunca possa ser estendida por outras classes
- e) Permitir que classes que herdam da classe base possam também ser herdadas de outras classes

3. A lógica de programação é essencial para a criação de algoritmos, que são essencialmente a especificação de um conjunto limitado de passos ou comandos que precisam ser seguidos em uma ordem específica e dentro de um tempo limitado para alcançar um resultado desejado, ou seja, resolver um problema. Com base nesse entendimento de algoritmos, considere as afirmações a seguir:

- I) A sequência lógica é a realização das operações ou instruções na ordem correta para alcançar um objetivo ou resolver um problema.
- II) Um algoritmo pode ter apenas uma única sequência lógica de instruções que leva à sua solução.
- III) A lógica de programação é a arte de ordenar pensamentos para alcançar um objetivo específico.



IV) Uma instrução é um conjunto de diretrizes ou normas estabelecidas para executar uma tarefa que culmina em uma ação específica.

A afirmação correta é somente a constante em:

- a) I e II
- b) I e III
- c) II e IV
- d) I, III e IV
- e) II, III e IV

4. A programação orientada a objetos apresenta uma forma mais “natural” de se implementar sistemas, modelando os objetos do mundo real como classes. A respeito dos conceitos do POO, avalie as seguintes asserções:

I. A herança permite que características comuns de entidades modeladas, sejam generalizadas em uma classe base, também chamada de superclasse. A partir dela, outras classes podem ser especializadas, onde cada uma apresentará características da superclasse, acrescidas de algumas particularidades, comportamentos e/ou características específicas.

II. Objetos representam a instância de alguma entidade do domínio do problema analisado, onde objetos similares são definidos em uma classe.

III. O encapsulamento permite que subclasses possam invocar métodos que possuem a mesma assinatura, mas implementações diferentes, especializadas por cada classe derivada.

Após a análise, aponte abaixo, apenas a alternativa que apresenta itens corretos.

- a) I, II e III
- b) I e II
- c) II e III
- d) I e III
- e) II, apenas



5. A programação orientada a objetos é um paradigma de programação que utiliza conceitos como classe e objeto para organizar e estruturar o código. De acordo com Pressman (2016), *"um objeto é uma instância de uma classe e é a unidade básica de um sistema orientado a objetos"*. Nesse contexto, é fundamental compreender a diferença entre classe e objeto e como eles se relacionam na programação orientada a objetos.

Qual a principal diferença entre classe e objeto na programação orientada a objetos?

- a) Uma classe é uma estrutura que define as propriedades e comportamentos de um objeto, enquanto um objeto é uma instância específica dessa classe.
- b) Uma classe é um objeto específico que possui propriedades e comportamentos definidos, enquanto um objeto é uma estrutura genérica que contém métodos e atributos.
- c) Uma classe é um conjunto de objetos relacionados entre si, enquanto um objeto é uma estrutura abstrata que define a hierarquia de um sistema.
- d) Uma classe é um objeto estático que não pode ser instanciado, enquanto um objeto é uma entidade dinâmica que pode ser modificada em tempo de execução.
- e) Uma classe é um objeto que representa um tipo de dado, enquanto um objeto é uma estrutura que armazena informações sobre o tipo de dado utilizado.

6. O encapsulamento é um dos pilares da orientação a objetos. Ele é utilizado a fim de que uma classe possa ocultar detalhes de implementação que devem ser conhecidos apenas por ela.

Considerando o texto apresentado, avalie as proposições a seguir.

I. Classes possuem construções que implementam propriedades e comportamentos de seus objetos. Somente as construções que implementam seus comportamentos é que podem ser encapsuladas.

II. A interface pública de uma classe é constituída por todos os membros que ela encapsula.



III. O encapsulamento de detalhes de implementação de uma classe promove o baixo acoplamento entre ela e as demais classes.

É correto apenas o que se afirma em:

- a) I
- b) II
- c) III
- d) I e II
- e) II e III

7. A herança é um dos pilares da orientação a objetos. Ela viabiliza a reutilização de código: quando uma superclasse define membros, eles podem ser herdados pelas suas subclasses.

Considerando o texto apresentado, avalie as proposições a seguir.

I. Ainda que a classe B herde da classe A, é possível que existam membros definidos em A não herdados por B.

II. Se a classe B é subclasse da classe A, ela também é superclasse de A.

III. Se a classe B é subclasse da classe A, ela herda todos os construtores de A.

É correto apenas o que se afirma em:

- a) I
- b) II
- c) III
- d) I e II
- e) II e III

8. Na programação orientada a objetos, a interface é um conceito que define um conjunto de métodos que uma classe deve implementar. Por outro lado, a implementação desses métodos é deixada para as classes concretas que implementam essa interface.



Nesse contexto, avalie as asserções a seguir e a relação proposta entre elas.

Interfaces em programação orientada a objetos são usadas para criar um contrato com as classes que irão implementá-la, obrigando que todos os métodos definidos na interface sejam codificados nas classes, promovendo assim o princípio da abstração.

PORQUE

Uma classe pode implementar múltiplas interfaces, o que permite a uma classe adquirir as funcionalidades de várias interfaces, sem a necessidade de herança múltipla, evitando assim problemas associados a ambiguidades de herança.

A respeito dessas asserções, assinale a opção correta.

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

9. Uma estrutura de repetição é uma instrução de fluxo de controle presente em linguagens de programação que permite que trechos de código sejam executados repetidamente até que uma determinada condição booleana seja atendida.

Considerando essa definição, avalie as afirmações a seguir:

I - Em uma estrutura de repetição do tipo PARA, o controle do laço é estruturado apenas pelo uso de uma variável lógica, que é iniciada como VERDADEIRA, encerrando o laço ao ter seu valor modificado para FALSO.

II - Na estrutura de repetição do tipo ENQUANTO, o teste do controle é realizado por um teste lógico que é realizado no início do laço. Se essa condição for FALSA, logo na primeira execução, o laço não será executado nenhuma vez.



III - Apenas as estruturas de repetição do tipo PARA e REPITA possuem controle do laço por meio de variáveis lógicas.

IV - Na estrutura de repetição do tipo REPITA o laço é executado no mínimo uma vez, já que a verificação da condição de parada é realizada no final do laço.

É correto afirmar que:

- a) Apenas as afirmativas I e II estão corretas.
- b) Apenas as afirmativas II e III estão corretas.
- c) Apenas as afirmativas III e IV estão corretas.
- d) Apenas as afirmativas I e III estão corretas.
- e) Apenas as afirmativas II e IV estão corretas.