



# Exercícios Eng Sw

## APOO

Prof. Marcio Victorino – [mcvictorino@uol.com.br](mailto:mcvictorino@uol.com.br)

Q1.

78 Em uma operação de sobrecarga, uma classe derivada pode redefinir operações de sua classe base.

79 Ao se utilizar o encapsulamento, não é necessário saber como ele funciona internamente, apenas como transmite os seus atributos.

Q2.

❑ Abstração é um conceito segundo o qual o sistema ou software é dividido em partes distintas. Compõe o ferramental necessário para um programa mais legível com uma melhor manutenção e melhor desempenho por meio da programação orientada a objetos.

❑ Abstração é a habilidade de se concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de software.

Q3.

69 A modelagem orientada a objetos pode demonstrar, entre outros aspectos, como eles podem ser classificados, de que forma herdam atributos e operações de outros objetos, como são compostos por meio de modelos de agregação e de que forma interagem por meio de modelos de comportamento simples.

91 No paradigma orientado a objetos, o polimorfismo está relacionado à herança de métodos. O Java possui duas modalidades de polimorfismo: overriding, que consiste em alterar o método em uma subclasse mantendo-se sua mesma assinatura da superclasse; e overloading, no qual há métodos de mesmo nome associados por meio da herança, porém, com argumentos diferentes.

Q4.

92 Quanto maior for o número de camadas, menor será o desempenho do software como um todo.

93 Cada camada tem comunicação (interface) com todas as demais camadas, tanto inferiores quanto superiores.

105 Em programação orientada a objetos, polimorfismo é o princípio pelo qual duas ou mais classes derivadas da mesma superclasse podem invocar métodos que têm a mesma assinatura, mas comportamentos distintos.

Q5.

97 Se uma subclasse herdar características de duas ou mais superclasses, ocorrerá uma herança múltipla.

107 Na programação orientada a objetos, o polimorfismo permite que cada componente de um programa agregue toda a informação relevante para sua manipulação como uma unidade. Com esse mecanismo ocultam-se detalhes de uma estrutura complexa, que podem interferir no processo de desenvolvimento da classe.

108 A herança em programação orientada a objetos é um relacionamento pelo qual uma classe, chamada de subclasse, herda todos os comportamentos e estados possíveis de outra classe, chamada de superclasse ou classe base.

Q6.

102 Na análise orientada a objetos, o encapsulamento auxilia a prevenção de interferências indesejadas de outros objetos sobre os métodos e atributos de uma classe.

105 Considerando-se que um anexo faz parte de apenas um requerimento e que um requerimento pode não apresentar nenhum anexo ou apresentar vários, é correto utilizar um relacionamento do tipo composição entre as classes “Requerimento” e “Anexo”.

Q7.

70 Em linguagens orientadas a objeto, um tipo abstrato de dados é materializado por uma estrutura de dados concreta modelada por uma classe.

71 O polimorfismo de sobrecarga é realizado por meio da aplicação de parâmetros diferentes em operadores distintos com o mesmo nome e semânticas distintas.

72 O método a ser utilizado no binding dinâmico, em tempo de execução, é determinado pela classe do objeto. Esse processo denomina-se resolução de sobrecarga.



Q8.

87 Por meio do encapsulamento, para impedir o acesso direto ao atributo de um objeto, são disponibilizados externamente apenas os métodos que alteram esse objeto. Por exemplo, não é preciso conhecer todos os detalhes dos circuitos de uma câmera digital para utilizá-la; a parte externa da câmera encapsula os detalhes, provendo para o usuário uma interface mais amigável.

88 Uma classe é capaz de instanciar um objeto de uma classe abstrata, para utilizar seus métodos e manipular seus atributos.

Q9.

120 Classes são constituídas por interfaces e nelas definidas. O nome de uma classe deve ser único e tem escopo dentro da interface na qual é declarado.

121 Um objeto apresenta três características básicas, o estado, a identidade e o comportamento. A parte de dados de um objeto é definida por um conjunto de mensagens, e a porção funcional, por um conjunto de atributos.

122 Objeto é o agrupamento de classes similares que apresentam os mesmos atributos e operações. Na definição de uma classe, é necessário estabelecer a que objeto ela ocorre como instância.

Q10.

- 123 Polimorfismo permite estabelecer uma interface comum que define os atributos que um objeto pode receber em um grupo de classes criadas por herança. Assim, ao longo de uma hierarquia de classes, uma operação pode ter o mesmo nome compartilhado, e cada pacote, uma implementação diferente.
- 124 Mensagem é uma solicitação entre classes para invocar certa operação, particularmente quando ocorre polimorfismo. Identidade é uma propriedade de um objeto que o distingue de todos os demais, sendo preservada até mesmo quando o estado do objeto muda completamente.
- 125 Herança é um mecanismo de reutilização de classes e atributos definidos em classes gerais por classes mais específicas que pode ser utilizado para expressar tanto associações quanto generalizações. Em herança linear múltipla, ocorrem múltiplas superclasses para uma mesma subclasse. As superclasses irão se comportar como classes concretas somente para o objeto instanciado na subclasse de herança múltipla, permanecendo abstratas para instâncias regulares.

Q11.

111 Uma classe pode ser vista como uma descrição generalizada de uma coleção de objetos semelhantes.

Q12. Acerca da arquitetura cliente-servidor multicamadas, julgue os itens subseqüentes.

63 Em arquiteturas cliente-servidor multicamadas, na maior parte das aplicações, o browser é adotado como cliente universal.

64 Nessa arquitetura, quando são consideradas três camadas, a primeira camada deve ser implementada por meio do servidor de aplicação.

65 Em arquitetura multicamadas, o servidor de aplicação nada mais é do que um programa que fica entre o aplicativo cliente e o sistema de gerenciamento de banco de dados.

Q13.

66 Essa arquitetura implica custo inicial de implantação maior que o do modelo convencional cliente-servidor, pois é necessário que haja separação física de cada camada em uma máquina, garantindo que a ocorrência de erro em uma camada não se propague para as demais.

67 Uma desvantagem dessa arquitetura é o aumento na manutenção da aplicação, pois alterações na camada de dados, por exemplo, acarretam mudanças em todas as demais camadas.

68 Nessa arquitetura, as camadas podem ter vários nomes. Por exemplo, a camada de aplicação pode ser chamada de camada de negócio, e a camada de dados pode receber o nome de camada EIS (enterprise information system).

Q14.

84 Na arquitetura cliente-servidor com três camadas (three tier), a camada de apresentação, a camada de aplicação e o gerenciamento de dados ocorrem em diferentes máquinas. A camada de apresentação provê a interface do usuário e interage com o usuário, sendo máquinas clientes responsáveis pela sua execução. A camada de aplicação é responsável pela lógica da aplicação, sendo executada em servidores de aplicação. Essa camada pode interagir com um ou mais bancos de dados ou fontes de dados. Finalmente, o gerenciamento de dados ocorre em servidores de banco de dados, que processam as consultas da camada de aplicação e enviam os resultados.

# Sistema Cliente-Servidor

- Sistemas cliente-servidor em duas camadas foram dominantes durante aproximadamente toda a década de 90 e são utilizados até hoje.
- Sistemas cliente-servidor em três camadas. Essas camadas normalmente recebem os seguintes nomes:
  - camada de apresentação
  - camada da lógica do negócio
  - camada de acesso
- Tier x Layer:
  - Tier: implica em separação física.
  - Layer: as camadas não precisam rodar em máquinas diferentes.
  - Fowler (2006, p.39) FOWLER, M. Padrões de Arquitetura de Aplicações Corporativas. 1. ed. São Paulo:Bookman, 2003. 493p.



- 1) E, C.
- 2) E, C.
- 3) C, E.
- 4) C, E, C.
- 5) E, E, C.
- 6) C, C.
- 7) C, C, E.
- 8) E, E.
- 9) E, E, E.
- 10) E, E, E.

- 11) C.
- 12) C, E, C.
- 13) E, E, C.
- 14) C.



FIM