

# Disciplina de Programação Orientada a Objetos - POOS3

## Curso Superior de ADS - 3º Semestre

(Professores Dênis Leonardo Zaniro e Fernando Vieira Duarte)

### Prova 1 (Valor: 10,0)

Observações:

- A. A prova possui duração correspondente a 4 aulas (8h - 11h35).
- B. Os exercícios podem ser resolvidos em um computador (ou notebook).
- C. Os exercícios da parte teórica devem ser resolvidos em algum processador de texto e as soluções devem ser salvas em um arquivo PDF.
- D. Todos os exercícios devem ser salvos em uma pasta cujo título seja seu nome completo.

#### Parte teórica

- 1) (1,0) Analise as afirmações a seguir e assinale V caso a afirmação seja verdadeira e F caso a afirmação seja falsa:
- I. ( ) Variáveis estáticas em uma classe não são inicializadas automaticamente pela Máquina Virtual Java.
  - II. ( ) Membros *final* de uma classe não inicializados em sua declaração devem ser inicializados em, pelo menos, um dos construtores da classe.
  - III. ( ) O construtor deve possuir o mesmo nome da classe, pode especificar parâmetros e também pode especificar tipo de retorno.
  - IV. ( ) Java permite que vários métodos com o mesmo nome sejam definidos. Para isso, basta que os métodos possuam tipos de retorno diferentes.
  - V. ( ) O coletor de lixo fica sob controle da Máquina Virtual Java. É possível definir o instante em que ele será executado invocando o método `System.gc()`.
  - VI. ( ) Os construtores de superclasse não são herdados por subclasses.
  - VII. ( ) Um relacionamento “tem um” é implementado via herança.
  - VIII. ( ) Quando uma subclasse define um método utilizando a mesma assinatura de algum método da superclasse, diz-se que a subclasse sobrecarrega esse método da superclasse.
  - IX. ( ) Suponha a existência de um método `x()` (método sem parâmetros) definido na classe `Pessoa`. Suponha que a classe `PessoaFisica` estenda a classe `Pessoa` e defina um método `x(int a)` (método com parâmetro inteiro). Nesse caso, existe uma sobrecarga de métodos.

- X. ( ) Considere a existência de uma classe chamada Pessoa e outra classe chamada PessoaFisica, que estende a classe Pessoa. Suponha a existência de um método x(), sem parâmetros, na classe Pessoa cujo tipo de retorno seja Pessoa, e um método x(), sem parâmetros, na classe PessoaFisica cujo tipo de retorno seja PessoaFisica. Nesse caso, existe sobreposição de métodos e todas as regras para se alcançar a sobreposição estão sendo respeitadas.

2) (1,0) Preencha as lacunas de cada uma das sentenças a seguir:

- a. Os objetos têm a propriedade de \_\_\_\_\_ - embora objetos possam saber comunicar entre si por interfaces bem definidas, normalmente não têm permissão de saber como outros objetos são implementados.
- b. Os programadores Java concentram-se na criação de \_\_\_\_\_, que contêm atributos e o conjunto de métodos que manipulam esses atributos e fornecem serviços para clientes.
- c. O projeto orientado a objetos tira proveito de relacionamento de \_\_\_\_\_, em que novas classes de objetos são derivadas absorvendo características de classes existentes e, em seguida, adicionando características únicas dessas mesmas classes.
- d. O tamanho, a forma, a cor e o peso de um objeto são considerados exemplos de \_\_\_\_\_ da classe do objeto.
- e. \_\_\_\_\_ é uma forma de capacidade de reutilização de software em que novas classes adquirem os membros de classes existentes e aprimoram essas classes com novas capacidades.
- f. Os membros \_\_\_\_\_ e \_\_\_\_\_ de uma superclasse podem ser acessados (ter visibilidade) pela superclasse e também pelas subclasses.
- g. Em um relacionamento \_\_\_\_\_ ou \_\_\_\_\_, um objeto de uma subclasse também pode ser tratado como um objeto de sua superclasse.
- h. Em um relacionamento \_\_\_\_\_ ou \_\_\_\_\_, um objeto de classe tem referências a objetos de outras classes como membros.
- i. Quando um objeto de uma subclasse é instanciado, um \_\_\_\_\_ de superclasse é chamado implicitamente ou explicitamente.
- j. Os construtores de subclasse podem chamar construtores de superclasse via palavra-chave \_\_\_\_\_.

### Parte prática

- 3) (Valor: 4,0) Elabore uma aplicação em Java que permita realizar alguns cálculos envolvendo figuras geométricas. Considere inicialmente as seguintes figuras: retângulos, triângulos e esferas. A seguir, são apresentados os requisitos que devem ser atendidos:

- a. Qualquer figura geométrica deve saber calcular a sua área.
  - b. Apenas as figuras de duas dimensões podem calcular seu perímetro, e apenas as figuras de três dimensões podem calcular seu volume.
  - c. Abaixo, são apresentadas as regras para os cálculos a serem efetuados:
    - i. A área do retângulo é dada por  $b * h$ , onde  $b$  é um dos lados e  $h$  é o outro lado. Seu perímetro é dado por  $2 * b + 2 * h$ .
    - ii. A área de um triângulo é dada por  $\sqrt{s * (s - a) * (s - b) * (s - c)}$ , onde  $a$ ,  $b$  e  $c$  são os lados do triângulo, e  $s$  é a metade de seu perímetro. Seu perímetro é calculado como  $a + b + c$  (soma de seus lados).
    - iii. A área de uma esfera é dada por  $4 * PI * r^2$ , onde  $r$  é o raio, e o seu volume é dado por  $4 * PI * r^3 / 3$ . Defina o valor de  $PI$  de tal forma que ele não possa ser alterado em qualquer parte do código. Além disso, seu valor deve ser compartilhado entre todas as figuras.
  - d. Crie uma classe chamada `GerenciaFigura` composta por métodos que saibam lidar com figuras geométricas. Nessa classe, deve haver um método que permita exibir a área de uma figura; um método que permita exibir o perímetro de uma figura; e, finalmente, um método que permita exibir o volume de uma figura (ao todo, são três métodos). A estrutura de classes deve ser projetada de tal forma que a classe `GerenciaFigura` não tenha que ser modificada se alguma figura específica for adicionada ao projeto. Além disso, considere a utilização de boas práticas de orientação a objetos.
  - e. Crie uma classe chamada `Principal`, composta pelo método `main()`, responsável pela instanciação de um retângulo e uma esfera a partir de dados predefinidos no código. Usando a classe `GerenciaFigura`, essa classe deve imprimir a área de ambas as figuras, o perímetro do retângulo e o volume da esfera.
- 4) (4,0) Elabore uma aplicação em Java que gerencie transações relacionadas a contas em uma instituição bancária. Considere os seguintes requisitos específicos:
- a. Nessa instituição, há contas poupanças e contas correntes e, para cada conta bancária, considere operações de saque e depósito. Quaisquer contas bancárias possuem número (gerado sequencialmente), nome do cliente e saldo, mas apenas contas correntes possuem limite. Portanto, operações de saque em contas correntes devem ser realizadas considerando-se o valor do limite (saldo + limite), e operações de saque em contas poupanças devem considerar apenas o valor do saldo. Caso o limite seja utilizado (contas correntes), o valor correspondente deverá ser descontado do saldo (saldo negativo).
  - b. Além de classes que representem contas bancárias, crie uma classe que gerencie uma lista de contas (defina um tamanho limite). Essa classe deve

permitir adicionar contas e realizar qualquer outro serviço que seja necessário. O código dessa classe deve ser elaborado de tal forma que, se novos tipos de conta forem incluídos ao projeto, ele não deverá ser alterado. Além disso, deve haver a garantia de que, em toda a aplicação, somente exista uma lista de contas. Em outras palavras, deve-se garantir que qualquer classe cliente manipule sempre a mesma lista.

- c. Crie uma classe chamada Principal, composta pelo método main(), que defina um valor de saque e um número de conta (valores predefinidos), e imprima uma mensagem informando se o saque foi realizado ou não (o saldo da conta deve ser atualizado adequadamente).
- 5) (+1,0) Escreva uma aplicação em Java que possua duas janelas gráficas (classes JanelaPrincipal e JanelaGenerica). A classe JanelaPrincipal deve permitir que o usuário informe um painel (“A” ou “B”) em um campo de texto e, com base no painel informado, essa classe deve carregar outra janela (classe JanelaGenerica) que seja composta pelo painel que foi informado. O painel “A” deve possuir um label denominado “A” e um campo de texto, e o painel “B” deve possuir um label denominado “B” e um botão também denominado “B”. O código de ambas as classes JanelaPrincipal e JanelaGenerica devem ser projetados de tal forma que eles não sejam modificados caso painéis sejam adicionados ou removidos na aplicação.

Ilustração:

