

# **Disciplina de Programação Orientada a Objetos - POOS3**

## **Curso Superior de ADS - 3º Semestre**

**(Professor Dênis Leonardo Zaniro)**

### **Lista de exercícios (Conteúdo 6 - Herança, classes abstratas e polimorfismo)**

- 1) Elabore uma aplicação em Java que contenha classes para representar pessoas físicas e pessoas jurídicas. Sabe-se que toda pessoa física possui nome, CPF e endereço. Toda pessoa jurídica possui nome, CNPJ e endereço. Um endereço deve armazenar número, bairro e cidade. Independentemente do tipo, para que uma pessoa seja instanciada, todos os seus dados devem ser definidos por classes clientes. Crie um método que retorne todos os dados, na forma de uma *string*, tanto para pessoas físicas quanto para pessoas jurídicas. Crie uma classe que instancie uma pessoa física e uma pessoa jurídica e, para ambas as pessoas, imprima todos os seus dados. Considere a reutilização de código nesse exercício.
- 2) Elabore uma aplicação para calcular e imprimir salários mensais de professores em uma universidade. Todos os professores possuem nome e código (número sequencial gerado automaticamente), e o cálculo do salário varia de acordo com o seu tipo. Para professores horistas, o salário é calculado considerando-se a quantidade de aulas ministradas, valor recebido por cada aula (valor único para todos os horistas) e um valor referente à hora-atividade. Considere que todo professor horista trabalha por 5.25 semanas durante cada mês. Para professores mensalistas, o salário é composto por um valor fixo (salário base) adicionado do valor de hora-atividade. Além disso, somente alguns professores mensalistas também são orientadores de TCC ou orientadores de IC. Além do valor recebido como professor mensalista, orientadores de IC recebem o valor correspondente a 4 aulas para cada trabalho orientado. Já os orientadores de TCC recebem o valor correspondente a 3 aulas para cada 2 orientandos. Suponha que, além da existência de orientandos, houvesse diversas similaridades entre orientadores de TCC e orientadores de IC. Independentemente da forma de se calcular o salário, observe que todos os professores devem saber como se calcula seu salário mensal. O cálculo de salário de orientadores de IC e orientadores de TCC não deve ser redefinido por qualquer outra classe nessa hierarquia. Todos os dados necessários para o cálculo do salário devem ser informados durante a instanciação de cada professor. Crie uma classe de testes que permita instanciar professores de todos os tipos e imprimir seus salários.
- 3) Elabore uma aplicação composta por um conjunto de classes que representem progressões aritméticas e geométricas. Uma progressão aritmética é uma sequência tal que, cada número, somado a um valor fixo (razão da progressão), resulta no próximo número da sequência. Já uma progressão geométrica também é uma sequência, mas cada número deve ser multiplicado por um valor fixo (razão da progressão) para se encontrar o próximo termo da sequência. Toda progressão, ao

ser instanciada, deve possuir uma razão e o primeiro valor da sequência. Além disso, qualquer progressão deve fornecer os seguintes métodos:

- a. Método primeiro(): deve retornar o primeiro termo da sequência.
- b. Método corrente(): deve retornar o termo corrente da sequência.
- c. Método próximo(): deve retornar o termo seguinte da sequência e atualizar o termo corrente.

Resolva esse problema da forma mais flexível possível.

- 4) Elabore uma aplicação para simular uma calculadora simples, que suporte inicialmente 5 operações aritméticas (soma, subtração, multiplicação, divisão e módulo) envolvendo dois números decimais quaisquer. Embora essa calculadora realize inicialmente apenas as operações citadas, seu código deve ser projetado de tal maneira que ele não precise ser alterado caso novos tipos de operações sejam adicionados ao projeto. Em outras palavras, a classe de calculadora não deve ser dependente de operações específicas. Os resultados de seu processamento devem ser mostrados por meio da classe JOptionPane. A operação desejada e os operandos (dois números) devem ser armazenados como atributos para a calculadora. Com base nesses atributos, uma determinada operação deve ser realizada em um método. Crie uma classe cliente que permita realizar operações (por exemplo, realize uma soma e uma subtração entre dois números quaisquer) usando essa calculadora.
- 5) Modifique a solução do Exercício 4 de tal forma que nenhuma classe cliente tenha que conhecer detalhes específicos de operações. Crie uma classe, composta pelo método main(), que leia do usuário a operação desejada e informe o resultado na tela utilizando-se os serviços da classe de calculadora. Dica: Crie uma classe que possua apenas a responsabilidade de instanciar operações.
- 6) Elabore um conjunto de classes a serem utilizadas em uma ferramenta que gerencie figuras geométricas. Inicialmente, deve-se considerar apenas quadrados, triângulos e losangos. Uma das capacidades que todas as figuras devem possuir é saber calcular sua área. Considere: a área do quadrado é dada por:  $\text{lado} \times \text{lado}$ ; a área do triângulo é dada por:  $\text{base} \times \text{altura} / 2$ ; e a área do losango é dada por:  $D \times d / 2$ , onde D é a diagonal maior e d é a diagonal menor. Crie também uma classe que permita gerenciar uma lista de figuras. Essa classe deve realizar as seguintes operações: 1) Armazenar figuras em uma lista (*array*); 2) Retornar uma figura de acordo com a posição informada por classes clientes; e 3) Retornar, na forma de uma *string*, o tipo e o valor da área de todas as figuras armazenadas em sua lista. Observe que o código dessa classe deve ser projetado de tal maneira que ele não precise ser alterado para processar novos tipos de figuras (além dos tipos existentes). Crie uma classe que instancie algumas figuras e utilize os serviços da classe de gerenciamento.
- 7) Crie uma aplicação gráfica que permita inserir empregados em uma empresa e consultar seus salários. Todo empregado possui nome, salário base e total de horas extras efetuadas. Considere inicialmente dois tipos de empregados: engenheiros e médicos. Portanto, a janela gráfica deve possuir um botão que permita inserir um

empregado e um botão que permita consultar seu salário. Além disso, a janela gráfica deve possuir um componente que permita selecionar se o empregado é um médico ou um engenheiro. Para esse propósito, use o componente **JRadioButton** ou o componente **JComboBox**. Sabe-se que, para engenheiros e médicos, paga-se o valor de R\$ 30,00 por cada hora extra realizada. Apenas engenheiros possuem CREA, e apenas médicos possuem CRM. O salário final de um médico é o salário base adicionado do valor referente a horas extras e também adicionado de um bônus de 10%, calculado sobre o salário base. Elabore todas as classes necessárias para resolver esse problema de forma adequada. A classe de janela não deve conhecer quaisquer detalhes específicos de médicos ou engenheiros. Em outras palavras, essa classe deve ser projetada de tal forma que não haja necessidade de alterações em seu código, caso novos tipos de empregados sejam adicionados ao projeto.

- 8) Escreva uma aplicação gráfica que permita gerenciar um *array* de nomes de pessoas (*array* de strings). Nessa aplicação, todos os *arrays* criados devem possuir tamanho máximo de 100 elementos. Crie uma classe que represente uma janela composta por um campo para a entrada de nomes e três botões. O botão “Inserir nome” deve permitir adicionar um nome ao *array*; o botão “Imprimir posição” deve permitir imprimir a posição de um dado nome (informado como entrada); e o botão “Imprimir nomes” deve permitir imprimir todos os nomes armazenados no *array*. A janela gráfica deve saber realizar a ação correta de acordo com o botão clicado pelo usuário de forma tal que seu código não seja dependente de um botão específico. Em outras palavras, o código da janela gráfica não deve ser alterado caso novos botões sejam adicionados (não utilize estruturas condicionais). **(Desafio)**