

LISTA 2

1. Implemente uma Classe Pessoa, contendo os atributos encapsulados e ainda o construtor padrão e pelo menos mais duas opções de construtores conforme sua percepção. Atributos: nome, endereço e telefone.
2. Considere, como subclasse da classe Pessoa (desenvolvida no exercício anterior) a classe Fornecedor. Considere que cada instância da classe Fornecedor tem, para além dos atributos que caracterizam a classe Pessoa, os atributos valorCredito (correspondente ao crédito máximo atribuído ao fornecedor) e valorDivida (montante da dívida para com o fornecedor). Implemente na classe Fornecedor, para além dos usuais métodos seletores e modificadores, um método obterSaldo() que devolve a diferença entre os valores dos atributos valorCredito e valorDivida. Depois de implementada a classe Fornecedor, crie um programa de teste adequado que lhe permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa.
3. Considere, como subclasse da classe Pessoa, a classe Empregado. Considere que cada instância da classe Empregado tem, para além dos atributos que caracterizam a classe Pessoa, os atributos codigoSector, salarioBase (vencimento base) e imposto (percentagem retida dos impostos). Implemente a classe Empregado com métodos seletores e modificadores e um método calcularSalario. Escreva um programa de teste adequado para a classe Empregado.
4. Implemente a classe Administrador como subclasse da classe Empregado. Um determinado administrador tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo ajudaDeCusto (ajudas referentes a viagens, estadias, ...). O salário de um administrador é equivalente ao salário de um empregado usual acrescido das ajudas de custo. Escreva um programa de teste adequado para esta classe.
5. Implemente a classe Operario como subclasse da classe Empregado. Um determinado operário tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo valorProducao (que corresponde ao valor monetário dos artigos efetivamente produzidos pelo operário) e comissao (que corresponde à percentagem do valorProducao que será adicionado ao vencimento base do operário). O salário de um operário é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.
6. Implemente a classe Vendedor como subclasse da classe Empregado. Um determinado vendedor tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo valorVendas (correspondente ao valor monetário dos artigos vendidos) e o atributo comissao (porcentagem do valorVendas que será adicionado ao vencimento base do Vendedor). O salário de um vendedor é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.

7. Implemente as seguintes as classes:

- a) Uma classe Pessoa com os seguintes atributos nome, sobrenome. A classe de poder ler e alterar os seus atributos. A classe Pessoa deve ainda ter um método chamado getNomeCompleto que retorna a concatenação do atributo nome como atributo sobrenome. Além disso, a classe de possuir um construtor sem parâmetros e um outro construtor que recebe como parâmetros o nome e o sobrenome da pessoa e alterar respectivamente os atributos nome e sobrenome.
- b) Uma subclasse de Pessoa, chamada Funcionario. A classe funcionário deve ter os atributos matrícula e salario. O salario de um funcionário jamais poderá ser negativo. Todo o funcionário recebe o seu salario em duas prestações, sendo 60 % na primeira prestação e 40 % na segunda prestação. Por isso classe Funcionario têm de implementar o método getSalarioPrimeiraPrestacao e o método getSalarioSegundaPrestacao.
- c) Uma subclasse de Funcionário chamada Professor. Todo o professor recebe o seu salário em uma única prestação
- d) Implemente uma classe TestaPessoa que instancia os seguintes objectos
 - pessoa1 (Pessoa)
Nome: Pedro
Sobrenome: Silva
 - pessoa2 (Funcionario)
Nome: Oscar
Sobrenome: Ribas
Salario: 182.000 Akz
 - pessoa3 (Professor)
nome: Oscar
sobrenome: Cardozo
salario: 1.500 Akz

Depois disso, execute as seguintes operações na seguinte ordem:

- d) Exibir a saída do método getNomeCompleto para os 3 objetos.
- e) Exibir a saída dos métodos getSalarioPrimeiraParcela e getSalarioSegundaParcela para os objetos pessoa2 e pessoa3.

8. Implemente as seguintes classes:

- a) (3 val) Implemente uma classe equipamento com o atributo ligado. A classe equipamento deve ainda conseguir ligar e desligar o equipamento
 - b) (5 val) Implemente uma classe EquipamentoSonoro que para além do atributo ligado têm ainda o volume que varia de 0 a 10, e um atributo stereo que indica se o equipamento está em stereo ou não. A classe deve permitir ler e alterar o volume e deve ainda implementar logica para um método **mono ()** e para um método **stereo ()**. Ao ligar o EquipamentoSonoro através do método liga, o seu volume é automaticamente ajustado para 5.
 - c) (1 val) Implemente a logica necessária ser possível vermos a informação sobre o estado de um EquipamentoSonoro.
 - d) (1 val) Crie uma classe TestaEquipamento, crie alguns equipamentos e teste toda a logica implementada anteriormente.
 - e) (4) Implemente um método que recebe um conjunto de equipamentos e que retorna aqueles que estão desligados e com o volume no nível 5. invoque esse método no main.
-
8. Implemente classe Transporte com os atributos ligado (tipo boolean) e velocidade (tipo int) e métodos liga() e desliga(). O método liga torna o atributo ligado true e o método desliga torna o atributo ligado false, além de tornar a velocidade zero. Crie também métodos get/set para modificar o atributo velocidade, sendo que a velocidade não pode ser negativa.
 - a) Uma subclasse de Transporte chamada Carro. Carro deve ter o atributo quilometragem (tipo int) e os métodos necessários para lê-lo e alterá-lo. A quilometragem não pode ser negativa, nem ultrapassar o valor 999999. A velocidade do Carro não pode ser negativa, nem ultrapassar 200.
 9. Implemente uma classe conta com o atributo saldo e métodos depositar e levantar para diminuir e aumentar o atributo saldo respectivamente, a partir de um valor especificado. O atributo saldo pode ser negativo.
 - a) Implemente uma subclasse de Conta chamada Poupanca. Poupanca deve ter o atributo diaRendimento do tipo int que armazena o dia do mês em que ocorre o rendimento da poupança. Além disso, ainda deve conter os métodos necessários para ler e alterar o atributo diaRendimento. O atributo saldo da Poupanca não pode ser negativo.

10. Uma sociedade **XPTO** pretende criar o **banco XPTO** e para parte do software pediu ajuda aos alunos de Programação III para modelar o sistema utilizando a POO. No banco XPTO existirão diferentes tipos de funcionários (**gerentes, telefonistas e secretarias**).

Todos os funcionários têm um nome e salário. Por sua vez os gerentes têm um username e senha para poderem ter acesso ao sistema do banco.

As secretarias têm um número de extensão e as telefonistas têm um código de estação de trabalho. Todos os funcionários do banco XPTO exceto os gerentes terão uma bonificação de 10 % do salário.

Como os gerentes executam tarefas com um grau maior de responsabilidade maior a sociedade XPTO decidiu que a bonificação para estes devia ser de 25 % do salário. A sociedade XPTO pediu ainda que o sistema deva permitir que os funcionários do banco consigam consultar os seus dados tais como nome, salário, bonificação etc...

No final teste o seu sistema criando diferentes tipos de funcionários.

11. Considere uma aplicação que contabiliza entradas num cinema. Considere que a aplicação tem uma classe denominada **Sessao** e nesta classe é necessário guardar além da data (que inclui a hora, minuto, ...) o título do filme, a duração e o número de clientes.

- a) Crie a classe de forma a que possua as propriedades referidas e um construtor que aceite os parâmetros indicados.
- b) Considere que o programa inclui um vector de sessões.
- c) Crie um método que devolva o número de sessões para um determinado título de filme e chame-o no método **Main()**.
- d) Crie um método que devolva o número de clientes para um determinado título de filme e chame-o no método **Main()**.

12. Considere uma aplicação que cria um catálogo de uma perfumaria.

- a) Crie uma classe **Perfume** que deve ter as propriedades **Designacao**, **Fabricante**, **Preco** e **Categoria**. Na mesma classe crie também um construtor que recebe os quatro atributos.
- b) Na classe do programa produza um método **PerfumesDaCategoria()** que devolve os perfumes de uma certa categoria passada por parâmetro.

13. Modele e implemente uma classe que represente uma versão muito simples do conceito Gato. Um Gato tem como características o nome, a idade e o peso.
- Implemente o método de comparação (`equals`), por forma a considerar que dois gatos são iguais se as suas características forem iguais.
 - Implemente o método de apresentação (`toString`), por forma a produzir uma cadeia de caracteres onde seja apresentado o nome, a idade e o peso do gato.
 - Implemente um programa (`main`) que ilustre a utilização dos métodos anteriores.
14. Faça um programa que calcule a área de uma figura geométrica. Aceite quatro tipos de figuras geométricas: quadrado, retângulo, triângulo e círculo.