PRÁTICA LABORATORIAL 08 – PROGRAMAÇÃO ORIENTADA A OBJETOS

Objetivos:

- Classes
- Objetos
- Métodos

EXERCÍCIOS

1. Escreva um programa onde cria uma classe **Pessoa** com atributos de nome, idade e altura.
   Defina os métodos de acesso (getters) para todos os atributos.
   Na classe Main:
   a) Crie duas instâncias da classe **Pessoa**, defina os seus atributos usando o construtor.
   b) Imprima o seu nome, idade e altura.

2. Escreva um programa onde cria uma classe **Edifício** com atributos de nome, rua, cidade, cor da fachada, número de andares e garagem (boolean: true se tem garagem ou false se não tem garagem).
   Defina os métodos de acesso (getters) para todos os atributos. Defina os métodos de acesso (setters) para a cor da fachada.
   Na classe Main:
   a) Crie uma instância da classe Edifício, definindo os seus atributos usando o construtor.
   b) Imprima na consola as específicações do edifício.
   c) Troque a cor do edifício (usando o set apropriado).
   d) Imprima na consola as específicações do edifício.
   e) Tente trocar outro atributo (como por exemplo: rua ou cidade). Irá perceber que não é permitido, pois na altura da criação da classe, ao não definirmos um método de acesso set para esses atributos, bloqueamos a sua alteração.

3. Escreva um programa onde cria uma classe **Retângulo** com atributos de cor, largura e altura.
   Desenvolva dois métodos para calcular a área e o perímetro do Retângulo (não aceitam parâmetros e tem retorno do resultado).
   Na classe Main:
   a) Instâncie um Retângulo, definindo os seus atributos usando o construtor.
   b) Imprima na consola o perímetro e a área do mesmo.

4. Escreva um programa onde cria uma classe **Círculo** com atributos de cor e raio.
   Desenvolva dois métodos para calcular a área e o perímetro do Circulo (não aceitam parâmetros e tem retorno do resultado).
   Na classe Main:
   a) Instância dois Círculos, definindo os seus atributos usando o construtor.
   b) Imprima na consola o perímetro e a área do maior círculo.

5. Crie uma classe chamada **Carro** com os seguintes atributos: marca, modelo, cor e ano de fabrico.
   Crie um método na classe **Carro** chamado ligar que imprime a mensagem "O {marca} {modelo} {cor} está ligado…". Por exemplo: “O Mercedes A250e branco está ligado…”.
   Na classe Main:
   a) Instância dois Carros, definindo os seus atributos usando o construtor.
   b) Invoque o método ligar para ambos os carros.

6. Crie uma classe chamada **Livro** com o seguintes atributos: título, autor, categoria, número de páginas e ISBN.
   Crie um método chamado exibirDetalhes( ) que imprime na consola todos os atributos do livro.
   Na classe Main:
   a) Instância dois Livros, definindo os seus atributos usando o construtor.
   b) Invoque o método de exibirDetalhes para ambos os livros.

   A partir deste exercício, crie o método **exibirDetalhes( )** para todas as classes.

7. Crie uma classe chamada **Aluno** com os seguintes atributos: nome, idade, email, curso e média.
   O construtor deve receber os cinco atributos. Crie métodos de acesso get para todos os atributos. Crie métodos de acesso set para o curso e para a média. Crie um método felizAniversario( ) que aumenta 1 à idade do Aluno. Crie um método boolean situacaoAprovacao( ) que com base na média retorna true se o aluno tiver aprovação ou false se o aluno estiver reprovado (se a média for maior que 9.5, considera-se aprovado).
   No Main:
   a) Instância dois Alunos, definindo os seus atributos usando o construtor, um deles com média de 15 e outro com média de 7.5.
   b) Invoque o método situacaoAprovacao( ) de forma a imprimir se um aluno está ou não aprovado.

8. Crie uma classe chamada **Produto** com os seguinte atributos: nome, preço e quantidade em stock.
   O construtor deve receber o nome e o preço. A quantidade em stock deve começar, por predefinição, a 0.
   Crie métodos de acesso get para todos os atributos. Crie método de acesso set para o preço.
   Crie um método para adquirirStock, que recebe um inteiro como parâmetro e incrementa ao stock existente, deve apresentar mensagem de confirmação.
   Crie um método para venderProduto, que recebe um inteiro como parâmetro que é a quantidade a ser vendida, analise se temos stock suficiente para satisfazer o pedido. Se tivermos stock, efetuamos a venda, apresentamos uma mensagem de sucesso incluindo o total da compra, e decrementamos ao stock. Se não tivermos stock, apresentamos mensagem de indisponibilidade de stock, e não alteramos o stock, uma vez que a venda não é efetuada.
   No Main:
   a) Instancie três produtos, definindo os seus atributos usando o construtor.
   b) Adquira stock para cada um deles.
   c) Tente fazer uma compra válida.
   d) Tente fazer uma compra inválida.
   e) Troque o preço de um produto.
   f) Tente fazer uma compra válida.
   g) Imprima o stock dos três produtos (verifique se estão como esperado)

9. Crie uma classe chamada **Funcionário** com os seguintes atributos: nome, email, departamento e salário.
   O construtor deve receber todos os atributos. Crie métodos de acesso get para todos os atributos. Crie método de acesso set para o departamento.
   Crie o método aumentarSalario que recebe um valor real por parâmetro que representa a percentagem do aumento a ser aplicado (por exemplo: recebe 15, representa um aumento de 15%).
   No main:
   a) Instância um funcionário, definindo os seus atributos usando o construtor.
   b) Exiba todos os dados do funcionário.
   c) Aumente o salário.
   d) Exiba novamente os dados do funcionário.

10. Crie uma classe chamada **ContaBancaria** que tenha os seguintes atributos: iban, titular e saldo.
    O construtor deve receber o iban e o titular. O saldo deve ser definido, por predefinição, a 0.
    Crie o método **depositar**( ) que recebe um valor real por parâmetro e incrementa esse valor ao saldo. Apresente uma mensagem de confirmação, como por exemplo: "1000€ depositados na conta 12345".
    Crie o método **levantar**( ) que recebe um valor real por parâmetro e decrementa esse valor ao saldo. Tenha em atenção que, para levantar uma determinada quantia, a conta deve ter saldo suficiente. Caso a conta tenha saldo suficiente, apresenta mensagem de confirmação, como por exemplo: "500€ levantados da conta 12345". Caso não tenha saldo suficiente, deve apresentar mensagem de erro: "Saldo insuficiente na conta 12345. Levantamento de 4000€ foi recusado".
    Crie o método **transferencia**, que recebe a Conta de Destino e o valor a transferir por parâmetro. Tenha em atenção que, para transferir uma determinada quantia, a conta deve ter saldo suficiente. Caso a conta tenha saldo suficiente, apresenta mensagem de confirmação, como por exemplo: "500€ transferidos da conta 12345 para a conta 99887". Caso não tenha saldo suficiente, deve apresentar mensagem de erro: "Saldo insuficiente na conta 12345. Transferência de 4000€ para a conta 99887 foi recusado". O valor deve ser adicionar e retirado às contas, respetivamente.
    No Main:
    a) Instancie três contas, definindo os seus atributos usando o construtor.
    b) Deposite alguns valores nas contas.
    c) Tente efetuar um levantamento válido.
    d) Tente efetuar um levantamento inválido.
    e) Tente efetuar uma transferência inválida.
    f) Tente efetuar uma transferência válida.
    g) A cada operação efetuada pode ir exibindo os detalhes das três contas para acompanhar as evoluções dos saldos.

Bom trabalho! ☺