

FICHA de Exercícios N°6

1. Observe as classes a seguir.
 - a. Estudante
 - b. Carro
 - c. Aula
 - d. Aviao
 - e. Palestra

- 2.1 Identifique e agrupe as classe abstratas e concretas;
- 2.2 Indique pelo menos 8 atributos para cada classe concreta;
- 2.3 Para as classes Estudante, Carro e Aula quais atributos são indispensáveis, atributos que não podem faltar.

2. Crie a classe Agricultor com os seguintes atributos: nome, data de nascimento, especialização, cultura favorita (exemplo: milho, arroz, etc).
 - a. Use o princípio de encapsulamento para garantir a segurança dos dados;
 - b. Crie um construtor que inicialize com todos atributos;
 - c. Crie os métodos de acesso (getters e setters);
 - d. Crie um método **dadosAgricultor()** que imprima todos os dados de um determinado agricultor.
 - e. Crie um método que identifica os agricultores com as mesmas especializações;
 - f. Crie um método **agricultorIdeal()** que receba culturas e identifica quais agricultores são especializados nestas culturas.

3. Crie a classe teste para o exercício anterior
 - a. Crie 4 agricultores e garanta que pelo menos dois deles sejam especializados na mesma cultura;
 - b. Imprima os dados dos agricultores com especializações diferentes;
 - c. Identifique o agricultor ideal para o plantio da cultura de milho;

4. Crie a classe Vendedor que contenha os seguintes atributos o nome vendedor, a data de nascimento, o tipo de produto que vende, o número total de vendas efectuadas, a quantidade total dos produtos vendidos e o salário.
 - a. Use o princípio de encapsulamento para garantir a segurança dos dados;
 - b. Crie um construtor que inicialize com todos atributos;

- c. Crie os métodos de acesso (getters e setters);
 - d. Crie o método **efectuarVenda()** que recebe o nome do produto e a quantidade;
 - e. Crie o método **dadosVendedor()** que imprima os dados do vendedor ocultando o salário ;
 - f. Crie o método **aumentarSalario()** que aumenta o salário caso o número total de vendas alcance 40% da quantidade total vendida.
 - g. Crie o método **melhorVendedor()** que identifique o melhor vendedor de acordo com as seguintes regras:
 - i. A quantidade total dos produtos vendidos deve ser duas vezes superior ao número total de vendas;
 - ii. O número de vendas total de vendas deve ser superior a 15.
5. Crie uma classe teste vendedor para o exercício anterior.
- a. Crie 5 objectos da classe Vendedor e efectue vendas com todos os objectos;
 - b. Identifique o melhor vendedor e aumente 20 % do seu salario;
 - c. Identifique o vendedor com mais vendas efectuadas;
 - d. Identifique o vendedor com o salario mais baixo e aumente para 10%;
 - e. Mostre os dados dos vendedores com o número de vendas inferior a 5.
6. O banco BOM, concederá um crédito especial aos seus clientes, variável com o saldo médio do último ano. Crie uma classe Conta com os seguintes atributos: numDeConta, saldoActual, tipoDaConta. Crie uma classe Cliente que possui um atributo do tipo Conta e mais 3 outros atributos que achar necessários a sua escolha.
7. Permita que seja possível efectuar as seguintes operações:
- a) Criar 10 Clientes com as suas contas.
 - b) Pesquisar clientes em função de um dos seus atributos;
 - c) Listar todos os clientes;
 - d) Ordenar o cliente em função do saldo que possuem na sua conta.
8. Readapte o exercício 7 de modo que permita que um cliente possa ter varias contas.
- a) Crie varias contas e atribua a um cliente;
 - b) Ordenar as contas do cliente em função do saldo;
 - c) Pesquisar um cliente em função do número de uma das suas contas;