**Lista de exercícios**

**Exercício 1: Clientes e contas**

1. Definir uma nova classe Cliente contendo os seguintes atributos:

- nome

- sobrenome

2. Definir uma nova classe Conta contendo os seguintes atributos:

- número da conta

- saldo

- titular (o titular é um objeto do tipo Cliente).

3. Criar um método chamado depósito que tenha como parâmetro de entrada

quantia em dinheiro. Como estamos fazendo um depósito, a quantia de dinheiro

é somada ao saldo. O método deve imprimir na tela o tipo de transação realizada

e o novo saldo.

4. Criar um método chamado saque que tenha como parâmetro de entrada

quantia em dinheiro. Como estamos fazendo um saque, a quantia é subtraída do

saldo. Caso o valor do saque a realizar seja maior que o saldo disponível, imprimir

na tela: “Saldo insuficiente”. Caso contrário, o método deve imprimir na tela o tipo

de transação realizada e o novo saldo.

5. Dentro do arquivo index, criar dois clientes, passando nome e sobrenome, e criar uma conta para cada um, inicializando número da conta, saldo e titular. Em seguida,

fazer um depósito e um saque em cada conta. Esclarecimento: definir, para cada

classe, os métodos getters e setters necessários.

**Exercício 2: Jogadores e treinamento**

1. Definir uma nova classe JogadorDeFutebol contendo os seguintes atributos:

- nome

- energia

- alegria

- gols

- experiência

2. Criar um método chamado fazerGol em jogador, que tire 5 pontos de energia e

dê 10 pontos de alegria ao jogador, além de aumentar o número de gols em 1. O

método deve imprimir “GOOOOL!” na tela.

3. Criar um método chamado correr em jogador, que tire 10 pontos de energia

dele. O método deve imprimir na tela: “Cansei”.

4. Definir uma nova classe SessaoDeTreinamento contendo os seguintes

atributos:

- experiência (ganha quando um treinamento é realizado)

5. Criar um método chamado treinarA que tenha como parâmetro de entrada um

jogador de futebol. Como é um treinamento, o jogador precisa:

- Correr

- Fazer um Gol

- Correr

Depois que o jogador fizer essas três coisas, a experiência dele deve aumentar,

somando um ponto na experiência. O método deve imprimir a experiência inicial

e a experiência final do jogador.

6. Dentro do arquivo index, criar dois jogadores de futebol, configurando os atributos

necessários, e criar uma sessão de treinamento, também com os atributos

necessários. Os dois jogadores devem realizar um treinamento.

Esclarecimento: definir, para cada classe, os getters e setters necessários.

**Exercício 3: Atletas e prova**

1. Definir uma nova classe Atleta contendo os seguintes atributos:

- nome

- nível

- energia

2. Definir uma nova classe Prova contendo os seguintes atributos:

- dificuldade

-energiaNecessaria

3. Criar, na classe Prova, o método podeRealizar, que tenha como parâmetro de

entrada um objeto do tipo

Atleta. O método deve retornar true caso o atleta possa realizar a prova e, caso

contrário, false. Um atleta pode realizar a prova se tiver um nível maior ou igual à

dificuldade da prova, além de energia suficiente.

4. Dentro do arquivo index, criar dois atletas, configurando os atributos necessários, e criar três provas para cada um, também com os atributos necessários. Depois,

verificar se os atletas podem realizar essas provas.

Esclarecimento: definir, para cada classe, os getters e setters necessários.

**Exercício 4: Tripé**

O tripé é um aparelho de três pés, com a parte superior circular ou triangular,

que permite estabilizar uma câmera e evitar os movimentos dela. Queremos

modelar o comportamento desse objeto.

1. Definir uma classe Tripé contendo os seguintes atributos:

- Dobrado: indica o estado atual do tripé (dobrado ou não)

- alturaMinima

- alturaMaxima

- alturaAtual.

2. Definir o método definirAltura(Integer novaAltura), que receba uma altura e

modifique o valor da altura

atual.

3. Definir o método dobrar(), que permita dobrar o tripé.

4. Definir o método desdobrar(), que permita desdobrar o tripé.

5. Definir o método guardar(), que permita deixar o tripé pronto para guardar. Ou

seja, ele deve estar dobrado e a altura atual deve ser a menor possível.

6. Definir o método prontoParaGuardar(), que permita verificar se o tripé está

pronto para ser guardado. Ou seja, ele deve estar dobrado e com a mínima altura

atual.

7. Definir o método usar(), que permita usar o tripé. Um tripé está pronto para usar

quando está desdobrado e com altura superior à metade da altura máxima.

8. Definir o método prontoParaUsar(), que permita verificar se o tripé está pronto

para ser usado. Ou seja, ele deve estar desdobrado e com altura superior à

metade da altura máxima.

9. Dentro do arquivo index, criar um tripé e fazer testes com os métodos definidos

anteriormente.

**Exercício 5: Veículos usados**

Queremos desenvolver um sistema informático para uma concessionária de

veículos usados. Essa concessionária decidiu que venderá apenas carros. Todos

os carros têm marca, modelo, ano de fabricação, cor e quilometragem.

Como estratégia de marketing, a concessionária quer ter um registro dos clientes

que compram os veículos. O departamento de marketing da concessionária

precisa saber o nome, o sobrenome e alguma informação de contato dos

clientes para poder oferecer outros carros no futuro.

Por outro lado, o departamento de contabilidade da concessionária quer ter um

registro das vendas realizadas.

Cada registro deve ter as seguintes informações:

-valorDaVenda

-veiculoVendido

-cliente

1. Implementar cada classe e definir os atributos. As classes que devem ser

implementadas são:

Concessionária, Cliente, Venda e Veículo.

2. Criar os construtores para as classes Cliente, Veículo e Venda.

- Cliente terá como parâmetros: nome, sobrenome e contato.

- Veículo terá como parâmetros marca, modelo, ano de fabricação, cor e

quilometragem.

- Venda terá como parâmetros cliente, veículo e valor da venda.

3. Definir, na classe Concessionária, o método registrarVenda(Veículo veículo,

Cliente cliente, Double valor), que adiciona ao registro de vendas da

concessionária uma nova venda de um veículo a um cliente por um valor

determinado.

4. Dentro do arquivo index, criar um carro, criar um cliente e registrar a venda desse

veículo a esse cliente pela Concessionária.

**Adicionais:**

**Adicional 1: (Atletas 2, a missão)**

Estender o exercício dos atletas para que:

1. Um atleta possa competir em um dos estádios do Rio. Cada estádio tem um

conjunto de provas

2. Dado um atleta, queremos saber como foi o desempenho dele em um estádio,

ou seja, que provas ele concluiu. O método recebe um atleta e retorna uma lista

com as provas concluídas.

3. Dados dois atletas e uma prova, queremos saber quem é o melhor. Ou seja, o

atleta que consiga concluir a prova e que tenha o maior nível atlético.

4. Dados dois atletas e um estádio, definir qual atleta ganhará a medalha de ouro.

Ou seja, aquele que consiga realizar melhor o maior número de provas.