



Lista 12

Exercícios:

1. Usando a estrutura de classes de futebol dadas nos slides da aula, faça as seguintes atividades:
 - a) Recrie a estrutura dos slides (note, elementos em Maiúsculo são classes):

```
jogo
jogo.Futebol
jogo.Equipe
jogo.pessoas
jogo.pessoas.Jogador
jogo.pessoas.Juiz
jogo.pessoas.Treinador
jogo.equipamentos
jogo.equipamentos.Campo
jogo.equipamentos.Bola
```
 - b) Adicione os atributos **private** em cada uma das classes:
 - A classe Equipe tem 5 jogadores (goleiro, zagueiro, ala direito, ala esquerdo e atacante)
 - A classe Equipe tem um treinador
 - A classe Futebol tem duas equipes, um juiz, um campo e uma bola
 - A classe Bola tem uma marca
 - A classe Campo tem largura e comprimento
 - As classes Jogador, Treinador e Juiz têm nome
 - c) Crie um **construtor** para cada classe, que permita inicializar todos atributos privados.
 - d) Crie um método **get** e um **set** para cada um dos atributos.
 - e) Crie uma classe chamada **Exercicio1** que possui um método **main**.
No método **main** você deve ler os dados do usuário, necessários para criar todos objetos.
 - f) Usando apenas o objeto criado a partir de **Futebol**, mostre todos os dados de todos objetos.

2. Crie uma classe chamada **Correntista** que possua dois atributos privados, **nome** e **saldo** (quanto dinheiro o correntista tem na sua conta), e que possua métodos **get** e **set** para manipular os valores dos atributos. Também inclua um **construtor**.

Escreva um programa chamado **Exercicio2** para ler o **nome** do correntista e seu **saldo** inicial e criar um objeto da classe **Correntista** para armazenar estes dados lidos.

Após informar nome e saldo, o programa deve ler os **movimentos** (débitos ou créditos) feitos pelo correntista na sua conta. Durante a leitura de cada movimento, o usuário deve informar o **tipo** e **valor** do movimento.

Os tipos de movimento são:

- 1 **Débito** (valor a ser subtraído do saldo; por exemplo, um pagamento de conta)
- 2 **Crédito** (valor a ser somado ao saldo; uma transferência ao seu favor)

O programa deverá encerrar a leitura de movimentos ao ser lido um movimento de **tipo 3**.

O **objetivo** do programa é calcular o **saldo final** de conta, que corresponde a quantidade de dinheiro restante na conta após todos movimentos serem contabilizados. Antes de encerrar, informe o **nome do correntista** e seu **saldo final**.

3. Crie uma classe chamada **CrescimentoPopulacional**, e adicione quatro atributos **private** do tipo **double**, chamados: **populacaoA**, **populacaoB**, **taxaA** e **taxaB**. Eles representam *respectivamente* o número de habitantes de uma cidade A e de uma outra cidade B, e a taxa de crescimento anual da população da cidade A e da população de B.

Exemplo: se o atributo **populacaoA** é igual a **1000**, então a cidade A tem 1000 habitantes. Se o atributo **taxaA** tem o valor **1.10**, isso quer dizer que a população da cidade A cresce 10% ao ano. Para calcular o número de habitantes da cidade A após **um ano**, multiplicamos **populacaoA** por **taxaA**, ou seja, **1000*1.10 = 1100**.

Crie um **construtor** para inicializar os 4 atributos da classe **CrescimentoPopulacional**.

Crie o método **calculaAnos**() que retorna um número **int**, correspondente ao número de anos necessários para que a população da cidade A seja maior que a população de B, dadas as suas respectivas taxas de crescimento. O método deve fazer as seguintes tarefas:

- Verificar se **populacaoA** >= **populacaoB**
Se o número de habitantes de A for igual ou maior que de B, então o método retorna **0**. Nesse caso, a população de A já é maior que a de B, não é necessário fazer cálculos.
- Verificar se **taxaA** <= **taxaB**
Se a taxa de A for menor que a taxa de B, então a cidade A nunca crescerá a ponto de ser maior que B. Nesse caso, o método retorna **0**.
- Nos demais casos, o método deve calcular o número de anos necessários para que a população da cidade A seja maior que a população da cidade B dadas as suas respectivas taxas de crescimento.

Finalmente, mostre o número de anos.