

Exercícios: CLASSES

Para cada exercício crie um novo projeto ou separe as classes por pacotes (packages).
Teste todos os programas na classe principal (classe que contém o método main).

Exercício 1. Crie uma classe para representar uma pessoa, com os atributos privados de **nome**, **data de nascimento** e **altura**. Crie os métodos **públicos** necessários para **sets** e **gets** e também um método para **imprimir todos dados de uma pessoa**. Crie um método para **calcular a idade da pessoa**.

Exercício 2. Crie uma classe chamada AdivinhaNumero. Essa classe deve ter um atributo privado int chamado numero. Crie os métodos get e set para o atributo. Crie um método privado chamado sorteiaNumero, que criará um valor aleatório de 1 a 100 para o atributo numero. Utilize esse método em um construtor para gerar esse número. Crie um método público chamado **dica** que receba como parâmetro **int tentativa**. Esse método orientará o usuário com as mensagens “o número é maior” ou “o número é menor”, até o usuário adivinhar o número. Se após 5 tentativas o usuário não acertar o número, o método emite uma mensagem revelando o número sorteado.

Exercício 3. Escreva uma classe classeLampada com um construtor de forma que instâncias desta só possam ser criadas se um estado inicial for passado para o construtor. Esse estado pode ser o valor **booleano** que indica se a lâmpada está acesa (true) ou apagada (false). Crie um atributo status do tipo string. Crie os métodos GET/SET correspondentes. Crie um objeto lampada1 com status **true** e outro objeto lampada2 com status **false**. Informe ao usuário se a lâmpada está acesa ou apagada, através do atributo status.

Exercício 4. Crie uma classe chamada Arma que tenha dois atributos inteiros que representem a capacidade e a quantidade de munição disponível. Os atributos devem ser privados. Crie nessa classe:

- _ Métodos get para os atributos.
- _ Construtor que não receba parâmetros e inicialize a capacidade com o valor 8.
- _ Construtor que receba apenas um parâmetro inteiro e inicialize a capacidade com esse valor.
- _ Construtor que receba dois parâmetros inteiros e inicialize a capacidade e a munição com esses valores.
- _ Um método chamado disparar sem parâmetros que, se houver munição na arma, decrementa a quantidade de munição e retorna true; ou simplesmente retorna false caso não haja munição.
- _ Um método chamado recarregar sem parâmetros que faz com que a munição receba o máximo da capacidade.
- _ Um método chamado recarregar que recebe a quantidade de munição que deverá ser utilizada para recarregar a arma. O método deverá retornar um inteiro que representa a quantidade de munição que por ventura tenha sobrado após o

recarregamento. Evidentemente, se a quantidade passada por parâmetro for menor ou igual do que a quantidade de munição que falta para recarregar totalmente a arma, o método deverá retornar 0.

Exercício 5. Crie uma classe **classeTelevisao** com atributo **volume** e um atributo **canal**, ambos do tipo **private int**. Crie os GETs/SETs correspondentes. Crie um construtor para a classe que inicialize os atributos com o valor 12 para canal e 0 para volume. Crie uma classe **classeControleRemoto** que pode controlar o volume e trocar os canais da televisão. O controle de volume permite:

- aumentar ou diminuir a potência do volume de som em uma unidade de cada vez. Limites 0 pra inferior e 50 para superior;
- aumentar e diminuir o número do canal em uma unidade. Limites de 10 a 100;
- trocar para um canal indicado;
- consultar o valor do volume de som e o canal selecionado.