



## Atividade 2: Conceitos Gerais

Obs.: Para entrega: a pasta `src` com os arquivos `.java` de cada um dos projetos. Os Exercícios 1 e 2 devem estar no mesmo projeto. Submeta uma pasta compactada na atividade do Tidia.

### Exercício 1

Crie um pacote **pessoa** e dentro deste, uma classe **Pessoa**, com os seguintes atributos: **nome**, **telefone**, **email**. Além dos métodos modificadores (`get/set`), crie também os seguintes métodos:

- alterar e-mail e telefone (um método para alterar os dois atributos)
- imprimir os dados de uma pessoa

Os métodos `set` e o de alteração do e-mail e telefone não devem aceitar a alteração se os valores passados forem vazios.

Escreva a classe principal (pacote *default*), instancie três pessoas e faça chamadas dos métodos para testar. Faça uma validação nos métodos para alterar o telefone e e-mail, e verifique se eles são alterados se os parâmetros passados são vazios.

### Exercício 2

Estendendo o Exercício 1, crie uma classe chamada **Agenda** (no mesmo pacote **pessoa**) para armazenar 10 pessoas. Escreva métodos para:

- adicionar uma pessoa na agenda
- remover uma pessoa da agenda
- verificar se uma pessoa está cadastrada na agenda (m caso positivo, deve retornar a posição (vetor) em que a pessoa está, caso contrário, retornar 0)
- buscar o telefone de uma pessoa pelo nome (ou parte do nome)
- buscar o email de uma pessoa pelo nome (ou parte do nome)
- imprimir os dados de uma pessoa

Escreva o programa principal (pacote *default*), instancie pessoas e adicione na agenda. Em seguida, faça a chamada dos outros métodos para testar e mostre os resultados. Tente inserir mais de uma vez a mesma pessoa ou, remover/alterar uma pessoa que não está na agenda.

### Exercício 3

Crie um pacote **carro** e implemente uma classe **Carro** que possui as seguintes propriedades: modelo, tipo de combustível (ex.: álcool, gasolina), a quantidade de combustível no tanque e o consumo (em km/l, ou seja, quantos quilômetros o carro percorre com um litro de combustível). Escreva métodos para:

- abastecer o tanque com uma quantidade de combustível;
- retornar a quantidade atual de combustível no tanque;
- retornar a quantidade de combustível necessária para percorrer uma certa distância (recebida como parâmetro).
- simular o uso do carro para percorrer uma certa distância (recebida como parâmetro), reduzindo a quantidade de combustível do tanque, caso haja combustível suficiente. Neste

caso, o método deve retornar o valor 1 (indicando que a operação foi realizada). Caso contrário, se a quantidade de combustível no tanque for insuficiente, deve retornar -1.

Escreva também os métodos modificadores (get/set) necessários e defina a visibilidade dos atributos e métodos.

Escreva o programa principal e crie três (ou mais) carros (de diferentes modelos e consumo). Abasteça os carros e simule o uso dos mesmos fazendo com que percorram algumas distâncias. Em seguida, solicite ao usuário uma distância a ser percorrida e imprima os dados do carro mais econômico, e se este possui combustível suficiente para o uso ou se será necessário abastecê-lo.

#### Exercício 4

Crie uma classe **NumeroComplexo** para representar um número complexo. A classe deverá ter os seguintes métodos:

- **inicializaNumero**, que recebe dois valores como argumentos para inicializar os atributos da classe;
- **imprimeNumero**, que deve imprimir o número complexo encapsulado usando a notação  $a+bi$ , onde  $a$  é a parte real e  $b$  a imaginária;
- **ehIgual**, que recebe um número complexo (outra instância da classe **NumeroComplexo**) e retorna verdadeiro se os valores dos campos encapsulados forem iguais aos da instância passada como argumento, ou falso, caso contrário;
- **soma**, que recebe outra instância da classe **NumeroComplexo** e soma este número complexo com o encapsulado usando a fórmula:

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

- **subtrai**, que recebe outra instância da classe **NumeroComplexo** e subtrai o argumento do número complexo encapsulado usando a fórmula:

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

- **multiplica**, que recebe outra instância da classe **NumeroComplexo** e multiplica este número complexo com o encapsulado usando a fórmula:

$$(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$$

- **divide**, que recebe outra instância da classe **NumeroComplexo** e divide o número encapsulado pelo passado como argumento usando a fórmula:

$$\frac{(a+bi)}{(c+di)} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}i$$

Escreva o programa principal, crie cinco números complexos ou mais e teste todos os métodos, apresentando os resultados.