

# Programação orientada a objetos: atributos e métodos

Prof. Rommel Dias

**Exercício 1** *Crie uma classe chamada Invoice que possa ser utilizado por uma loja de suprimentos de informática para representar uma fatura de um item vendido na loja. Uma fatura deve incluir as seguintes informações como atributos: o identificador (String) do item faturado, a quantidade comprada do item e o preço unitário do item.*

*Na classe que possui a função principal, declare e inicialize um objeto do tipo Invoice. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço por item não for positivo ele deve ser configurado como 0.0. Além disso, forneça um método que calcula o valor da fatura e depois retorna o valor como um double.*

*Implemente um método para mostrar as informações (atributos) da classe. Em seguida, declare e inicialize um vetor com informações para n Invoice, onde n é um dado de entrada. Mostre as informações de todos os objetos do tipo Invoice cadastrados.*

**Exercício 2** *Crie uma classe Data para representar datas observando os pontos a seguir.*

- *represente uma data usando três atributos: o dia, o mês, e o ano;*
- *implemente um método que retorna a representação da data como String. Considere que a data deve ser formatada mostrando o dia, o mês e o ano separados por barra (/).*

**Exercício 3** *Uma das operações que podemos efetuar com datas é a comparação para ver se uma data ocorre antes de outra. O algoritmo para comparação é muito simples, e seus passos estão abaixo. Nesse algoritmo, consideramos que dia1, mês1 e ano1 são os dados da primeira data, e que dia2,*

*mês2 e ano2 são os dados da segunda data. Escreva um método vemAntes na classe Data que receba como argumento outra instância da classe Data e implemente o algoritmo acima, retornando true se a data encapsulada vier antes da passada como argumento e false caso contrário. Se as datas forem exatamente iguais, o método deve retornar true.*

**Exercício 4** *Uma imobiliária trabalha com dois tipos de imóveis (casa e apartamento). Todo e qualquer imóvel possui (**atributos**) um tipo (1 para casa e 2 para apartamento), um proprietário (nome), um tamanho (em metros quadrados), um identificador se está em área nobre ou não, a idade do imóvel (em anos) e um valor de compra.*

*Implemente um método para calcular o IPTU de um imóvel. Usa-se as seguintes regras para o cálculo:*

Tipo	Tamanho	IPTU (% sobre valor de compra)
Apartamento	> 100	0,05
Apartamento	≤ 100	0,02
Casa	> 100	0,03
Casa	≤ 100	0,01

*Também manipulando os dados fornecidos, pode-se calcular um possível preço de venda usando as seguintes condições:*

Área nobre	Tamanho	Valor de venda
Sim	> 100	80% o valor da compra
Sim	≤ 100	81% o valor da compra
Não	> 100	82% o valor da compra
Não	≤ 100	80% o valor da compra

*Sabendo disso:*

- *implemente a classe Imovel com os devidos atributos e métodos para o cálculo do IPTU e do valor de venda;*
- *implemente um método para mostrar as informações (atributos) do imóvel;*
- *declare e inicialize um vetor com informações para n imóveis, onde n é um dado de entrada;*
- *mostre os IPTUs e os valores de venda dos n imóveis;*

- *mostre o maior tamanho de um imóvel cadastrado;*
- *mostre o menor tamanho de um imóvel cadastrado;*
- *mostre quantos imóveis do tipo apartamento e do tipo casa foram cadastrados;*
- *mostre a média do valor de venda dos  $n$  imóveis;*
- *mostre quantos imóveis do tipo casa localizado em área nobre foram cadastrados;*
- *mostre o nome do proprietário do imóvel mais caro;*
- *mostre os nomes de todos os proprietários dos imóveis com idade superior a 10 anos;*
- *implemente um método que receba como parâmetro um imóvel. Cheque se ambos têm a mesma idade.*