

## Exercício 1: Sistema de Biblioteca

**Objetivo:** Desenvolver um software básico de biblioteca que permita gerenciar livros, adicionando-os e listando-os.

### Descrição:

Você foi contratado para desenvolver um sistema básico para uma *biblioteca*. O sistema deve ser capaz de gerenciar informações sobre *livros* e permitir que novos livros sejam adicionados à biblioteca e que todos os livros presentes na biblioteca sejam listados. As informações a serem mantidas sobre livros são: (título, autor, ano de publicação). O bibliotecário deseja obter as informações de todos os livros cadastrados na biblioteca.

Dica: utilize a classe `List`, e seus métodos `add` e `get` para manipular a lista de livros.

```
List<Livro> livros= new ArrayList<>();
```

### Exemplo de uso:

Implemente um programa principal (classe `Main`) que:

1. Crie pelo menos 3 objetos da classe `Livro` diferentes
2. Crie um objeto `Biblioteca` e adicione os livros criados a ela.
3. Utilize o método `listarLivros()` da classe `Biblioteca` para exibir todos os livros presentes na biblioteca no console.

## Exercício 2: Sistema Escolar

**Objetivo:** Criar um sistema de gerenciamento de alunos que permita adicionar alunos a uma turma e listar todos os alunos de uma turma.

### Descrição:

Você foi solicitado a desenvolver um sistema de gerenciamento para uma escola. O sistema deve gerenciar informações sobre *alunos* e *turmas*, permitindo que novos alunos sejam adicionados a uma turma específica e que todos os alunos dessa turma sejam listados. As informações a serem mantidas dos alunos são: (nome, idade, e matrícula). Já para as turmas as informações são (nome, código).

Aluno ("Lucas", 22, 1234)

Turma("Turma A", 1A)

### Exemplo de uso:

Implemente um programa principal (classe `Main`) que:

1. Crie 05 alunos com diferentes valores para `nome`, `idade` e `matricula`.
2. Crie 02 turmas com seu nome de turma e adicione os alunos a ambas..

3. Utilize o método `listarAlunos()` da classe `Turma` para exibir todos os alunos presentes em cada turma.

### Exercício 3: Sistema de Compras

**Objetivo:** Desenvolver um sistema de compras que permita adicionar produtos a um carrinho de compras e listar todos os produtos no carrinho.

#### Descrição:

Você foi contratado para desenvolver um sistema básico de compras. O sistema deve gerenciar informações sobre *produtos* e permitir que novos produtos sejam adicionados a um *carrinho de compras* e que todos os produtos presentes no carrinho sejam listados. O aplicativo também deverá ser capaz de informar o valor total dos produtos contidos no carrinho de compras.

Dica: Para formatar uma String para exibir valores

Considere as informações: nome, preço e quantidade para os produtos.

#### Exemplo de uso:

Implemente um programa principal (classe **Main**) que:

1. Crie alguns produtos diferentes.
2. Crie e adicione pelo menos 03 produtos ao carrinho de compras.
3. Após, informe no console, os produtos que estão no carrinho e seus sub-totais, bem como o valor total da compra:

	Nome	Valor Unit	Qtd	Sub-total
Ex.	Mouse Gamer	R\$ 60,00	2	R\$ 120,00
				Total: R\$ 120,00

4 - Você foi contratado para desenvolver uma aplicativo Java para uma empresa chamada HealthCare. Esse software deverá ter uma classe chamada Paciente que possui um construtor que recebe o seu peso em quilos e sua altura em metros, ambos utilizando o tipo double. Crie um método chamado calcularIMC() que calcula o índice de Massa Corporal de acordo com a fórmula  $IMC = \text{peso (quilos)} / (\text{altura} * \text{altura (metros)})$ . Crie também um método chamado diagnostico() que utiliza o método calcularIMC() e retorna uma String de acordo com as seguintes faixas de valor:

- **Baixo peso muito grave** = IMC abaixo de 16 kg/m<sup>2</sup>
- **Baixo peso grave** = IMC entre 16 e 16,99 kg/m<sup>2</sup>
- **Baixo peso** = IMC entre 17 e 18,49 kg/m<sup>2</sup>
- **Peso normal** = IMC entre 18,50 e 24,99 kg/m<sup>2</sup>
- **Sobrepeso** = IMC entre 25 e 29,99 kg/m<sup>2</sup>
- **Obesidade grau I** = IMC entre 30 e 34,99 kg/m<sup>2</sup>
- **Obesidade grau II** = IMC entre 35 e 39,99 kg/m<sup>2</sup>
- **Obesidade grau III (obesidade mórbida)** = IMC igual ou maior que 40 kg/m<sup>2</sup>

Escreva uma classe chamada Principal em que se criam 3 instâncias da classe Paciente com valores diferentes e se imprime no console o resultado dos dois métodos criados.

Bônus: Crie um atributo nome, para a classe paciente, e um método setNome(), para dar nome aos pacientes. Após, escreva no console a mensagem configurada anteriormente junto ao nome do paciente.

Ex.:

O diagnóstico do pac. Lucas é Peso normal = IMC entre 18,50 e 24,99 kg/m²

5 - Cansado de conferir os descontos em sua folha de pagamento manualmente, João da Silva lhe contratou para escrever um aplicativo que faça o cálculo do valor de imposto de renda a ser recolhido de seu salário. A tabela de descontos do IRPF 2024 é a seguinte:

Faixas do IR	Parcela salarial em cada faixa	Alíquota
Faixa 1:	Até R\$ 2.112	isento
Faixa 2:	De 2.112,01 até 2.826,66	7,5%
Faixa 3:	De 2.826,67 até 3.751,06	15%
Faixa 4:	De 3.751,07 até 4.664,68	22,5%
Faixa 5:	Acima de R\$ 4.664,68	27,5%

Tabela-1 Faixa IRPF 2024

A tabela abaixo mostra um exemplo de cálculo para um salário de R\$ 4.000,00:

Faixas do IR	Parcela salarial em cada faixa	Alíquota	Imposto pago sobre a parcela
1	até R\$ 2.112	isento	zero
2	R\$ 714,65	7,5%	R\$ 53,59
3	R\$ 924,39	15%	R\$ 138,66
4	R\$ 248,93	22,5%	R\$ 56,01
Total	Salário: R\$ 4.000	Alíquota efetiva: 6,2%	Total pago: R\$ 248,26

Sabendo disso escreva uma classe *Pessoa* deve receber via construtor os atributos nome(String) e salário (double);

Em seguida crie uma classe *FaixaImpostoDeRenda*. Essa classe retorna um inteiro conforme a faixa da tabela-1. Crie outra classe *ImpostoDeRenda* que receberá como parâmetro um objeto da classe *Pessoa*, em seu construtor. A classe *ImpostoDeRenda* deverá ter um método *calcular( )*, calculando o IR a pagar e definindo a faixa de IR da pessoa e também o método *imprimir( )*, que deverá imprimir as informações na tela.

Após na classe principal crie alguns objetos do tipo *Pessoa* e *ImpostoDeRenda* e imprima no console.

Ex.:

Lucas seu salário é R\$ 5700,00 sua faixa de imposto é 5 e o valor do imposto de renda a ser pago é: R\$ 682,54

Bônus.

Adquira as informações de nome e salário da classe *Pessoa* via console, utilizando a classe *Scanner*;

Ex.:

```
Scanner scanner = new Scanner(System.in);  
var salario = scanner.nextLine().toString();  
var nome = scanner.nextDouble();  
scanner.close();
```

Mais informações:

<https://g1.globo.com/economia/imposto-de-renda/noticia/2024/05/08/imposto-de-renda-2024-veja-a-tabela-de-aliquotas-e-saiba-como-fazer-o-calculo.ghtml>