

Aula 04

Exercícios

1. Crie uma classe `Calculadora`, que possui métodos para realizar operações básicas (soma, subtração, multiplicação e divisão).

Devem ser implementados os métodos:

- `somar`
- `subtrair`
- `multiplicar`
- `dividir`

Todos os métodos receberão dois números do tipo `double` e devem retornar o resultado da operação.

Crie também uma classe de teste para:

- Criar um objeto da classe `Calculadora`
- Informar que será feita uma soma e solicitar dois números. Apresentar o resultado.
- Informar que será feita uma subtração e solicitar dois números. Apresentar o resultado.
- Informar que será feita uma multiplicação e solicitar dois números. Apresentar o resultado.
- Informar que será feita uma divisão e solicitar dois números. Apresentar o resultado.

2) Implemente a classe abaixo

ContaBancaria
- numero: int
- agencia: int
- titular: String
- saldo: double

- a) Adicione um construtor na classe, que receba três parâmetros: número da conta, agência e titular. O saldo inicial sempre deve ser zero.
- b) Implemente métodos getters para todos os atributos.
- c) Implemente o método `depositar`. Ele deve receber um valor que será adicionado ao saldo. Este método não tem retorno.
- d) Implemente o método `sacar`. Ele deve receber um parâmetro que indica o valor que será removido do saldo. Este método não tem retorno.
- e) Implemente o método `imprimirInfosConta`. Este método não tem nenhum retorno e nem parâmetros. Deve apenas mostrar em tela todos os dados da conta.
- f) Crie uma classe de teste para:
- Instanciar um objeto da classe `ContaBancaria`.
 - Depositar R\$ 200,00
 - Depositar \$ 80, 40
 - Sacar R\$ 100,00
 - Sacar R\$ 20,50

Verifique o saldo da conta após cada operação. O resultado apresentado é conforme o esperado?

3) Crie uma classe chamada `FichaAtendimento`. Essa classe será utilizada para controlar uma fila de clientes em um estabelecimento. A classe possui apenas um atributo: `qtdClientes`. Toda vez que instanciarmos um objeto dessa classe, o atributo deve iniciar com zero.

Implemente os seguintes métodos na classe:

- `adicionarCliente`: esse método não tem parâmetro e não tem retorno. O objetivo é adicionar uma pessoa na quantidade de clientes na fila.
- `atenderCliente`: esse método não tem parâmetro e não tem retorno. O objetivo é remover uma pessoa da quantidade de clientes na fila, pois será atendido.
- `getQtdClientes`: é o método `get` para o atributo.
- `verificarFilaVazia`: este método não tem parâmetro e seu retorno é do tipo `boolean`. Retornará `true` se houver ao menos um cliente na fila. E retornará `false` se não houver nenhum cliente na fila.

- `imprimeTamanhoFila`: este método escreve em tela quantos clientes estão na fila naquele momento. O método não possui parâmetros, nem retorno.

Crie uma classe de teste para verificar a implementação.

Use o seguinte cenário de teste:

Cenário de teste	
Entrada	Saída esperada
Adicionar um cliente na fila	
Adicionar uma cliente na fila	
Imprimir o tamanho da fila	Deve informar que duas pessoas estão na fila.
Atender um cliente	
Imprimir o tamanho da fila	Deve informar que uma pessoa está na fila.
Adicionar uma cliente na fila	
Atender um cliente	
Adicionar uma cliente na fila	
Verificar se a fila está vazia	Deve retornar false

4) Implemente a classe abaixo

Pedido
- <code>valorTotal</code> : double
- <code>totalItens</code> : int

Essa classe representa um pedido de um cliente em uma loja online.

a) Acrescente o método abaixo:

```
adicionarItem(double valorUnitario, int qtd);
```

Este método deve calcular o valor a ser adicionado ao `valorTotal`. Para tanto, primeiro deve-se multiplicar o valor unitário pela quantidade de itens para só então adicionar ao valor total.

O atributo `totalItens` também deve ser atualizado de acordo com as informações recebidas.

b) Implemente o método abaixo que deve mostrar em tela qual o valor total da compra naquele momento e a quantidade de itens.

```
imprimirResumoConta();
```

c) Crie uma classe de teste para verificar a implementação feita.

Utilize o seguinte cenário:

Cenário de teste	
Entrada	Saída esperada
Adicionar 4 unidades de um produto que custa R\$ 5,40 a unidade.	
Imprimir o resumo da conta	Valor total: R\$ 21,60 Total itens: 4
Adicionar 1 unidade de um produto que custa R\$ 19,30 a unidade.	
Imprimir o resumo da conta	Valor total: R\$ 40,90 Total itens: 5
Adicionar 3 unidades de um produto que custa R\$ 12,10 a unidade.	
Imprimir o resumo da conta	Valor total: R\$ 77,20 Total itens: 8