

## Programação 2 e Introdução à Programação 2 - Lista 1 2020.4

### GIT (pontuação total 2,0 pontos)

Esta atividade tem como objetivo fixar os conceitos de GIT através da execução de partes de um tutorial como se segue. Para comprovação da execução desta atividade, você deverá gravar vídeos capturando sua tela e sua voz executando o tutorial passo a passo. O vídeo deverá estar disponível em plataforma pública (YouTube ou similar) e o link deverá ser postado como texto na submissão da atividade.

- 1) (1,0) Acesse o site [https://learngitbranching.js.org/?locale=pt\\_BR](https://learngitbranching.js.org/?locale=pt_BR) selecione o nível "Main" e grave a execução dos itens marcados em vermelho.



- 2) (1,0) Acesse o site [https://learngitbranching.js.org/?locale=pt\\_BR](https://learngitbranching.js.org/?locale=pt_BR) selecione o nível remote e grave a execução de todos os itens, conforme apresentado na figura abaixo.



### Estoque (pontuação total 2,0 pontos)

3) Escreva as seguintes classes em Java. Você tem a liberdade de definir os tipos dos atributos, então pense bem para usar um tipo adequado ao propósito (ex.: atributo com o nome 'produto' deve ser do tipo Produto que vc definiu):

a) Produto:

- Deverá conter os atributos privados: codigo, nome.
- Deverá conter um construtor sem parâmetros e outro contendo os parâmetros (codigo, nome).
- Deverá conter o método:
  - Método que altera o nome do produto
    - mudarNome(nome)

b) Estoque:

- Deverá conter os atributos privados: produto, qtdAtual e qtdMinima.
- Deverá conter um construtor sem parâmetros e outro contendo os parâmetros (produto, qtdAtual e qtdMinima).
- Deverá conter os métodos:
  - Método que altera o valor da quantidade mínima
    - mudarQtdMinima(qtdMinima)
  - Método soma a1o valor atual o valor passado
    - reporEstoque(qtd)
  - Método subtrai do valor atual o valor passado
    - darBaixa(qtd)
  - Método que escreve os valores atuais dos atributos
    - relatorio()

c) Escreva uma classe TesteEstoqueMain que vai realizar as seguintes operações:

- Instanciar os objetos:
  - produto01
    - codigo: 1
    - nome: Teclado
  - produto02
    - codigo: 2
    - nome: Mouse
  - estoque1
    - produto: produto01
    - qtdAtual: 20
    - qtdMinima: 15
  - estoque2
    - produto: produto02

- qtdAtual: 30
- qtdMinima: 10

d) Execute as operações na seguinte ordem:

- Dar baixa em 10 unidades de estoque1;
- Fazer a reposição de 19 unidades de estoque2;
- Executar o método relatorio das instâncias estoque1 e estoque2;
- Mudar o nome do produto01 para "Teclado virtual";
- Executar o método relatorio da instância de estoque responsável pelo atributo produto02;

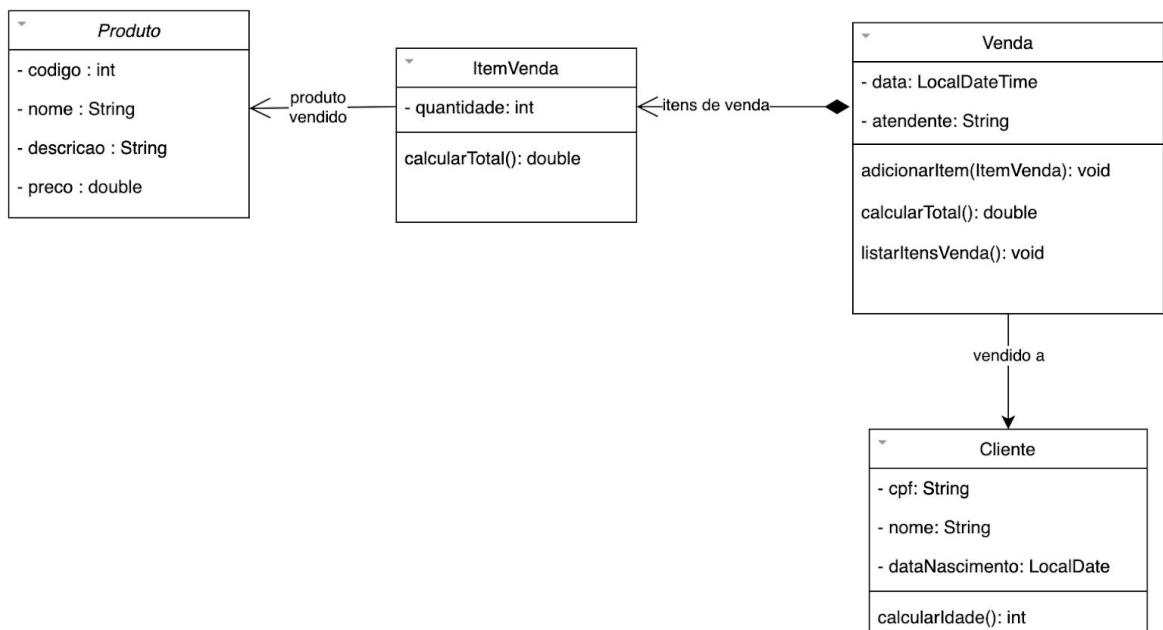
### **java.time (pontuação total 2,0 pontos)**

4) Estude o pacote `java.time.*` e, usando as classes dele, escreva um programa que:

- a) (0,2 pontos) Pegue uma data que o usuário digitou no teclado e transforme em um objeto do tipo apropriado que será usado em outras partes do seu programa
- b) (0,2 pontos) Imprima a diferença (em dias) da data de hoje para a data que o usuário digitou no teclado
- c) (0,2 pontos) Some 110 dias à data digitada pelo usuário e imprima qual a nova data obtida
- d) (0,2 pontos) Altere o mês da data informada para ser igual ao mês de agosto
- e) (0,2 pontos) Imprima a data alterada no formato "dd-MMM-YYYY". Para isso você deve usar a classe `java.time.format.DateTimeFormatter`.
- f) (1,0 ponto) Nesta mesma classe que você escreve seu programa, crie um método que recebe duas datas (`java.time.LocalDate`) e calcula a quantidade de dias úteis (excluindo sábados e domingos) contidos entre essas duas datas. Para testar o método mencionado, peça que o usuário forneça duas datas usando o teclado e então calcule a quantidade de dias úteis entre essas datas através da invocação do método.

## Classes e relacionamentos (pontuação total 4,0 pontos)

5) Com base no diagrama UML abaixo, implemente as 4 classes em Java.



- **(0,5) Produto**
  - implemente os métodos get/set para cada atributo
- **(0,5) Cliente**
  - implemente os métodos get/set para cada atributo
  - método `calcularIdade()`: retorna a idade do cliente com base no atributo `data Nascimento`
- **(0,5) ItemVenda:**
  - implemente os métodos get/set para cada atributo
  - método `calcularTotal()`: deve multiplicar o preço do produto do item de venda pela sua quantidade
- **(1,5) Venda:**
  - implemente os métodos get/set para cada atributo
  - método `adicionarItem(ItemVenda item)`: adiciona um item de venda ao objeto venda
  - método `calcularTotal()`: efetua a soma de todos os itens de venda adicionados ao objeto venda
  - método `listarItensVenda()`: imprima os itens de venda conforme exemplo a seguir:

OBS: Siga as tabulações e alinhamentos conforme exemplo. Observe que os valores numéricos estão alinhados à direita, enquanto os textuais estão alinhados à esquerda. Use o método `System.out.printf(...)` para atingir esse resultado

de forma rápida e organizada. Vide <https://bit.ly/2L6olzV> ou <https://bit.ly/37GEvxb>.  
Vídeo com exemplos de printf: <https://bit.ly/39OKr9R>.

### Tabela 1. Exemplo de impressão de uma Venda

Data da Venda: 23/12/2020

Cliente: José Maria Marques | CPF: 123456 | idade: 26

Atendente: Maria Dolores Sampaio

Nome do produto	Preço	Quantidade	Total
-----	-----	-----	-----
Produto 01	R\$ 10.00	10	R\$ 100.00
Produto 02	R\$ 5.00	5	R\$ 25.00
Produto 03	R\$ 20.00	10	R\$ 200.00
Produto 04	R\$ 15.00	2	R\$ 30.00
-----	-----	-----	-----
Total da Venda:			R\$ 355.00

- **(1,0) Classe contendo método main:**

A. Escreva uma classe TesteVendaMain que vai realizar as seguintes operações:

- Instanciar 4 objetos do tipo Produto contendo os dados apresentados na **Tabela 1**
- Instanciar 4 objetos do tipo ItemVenda contendo os dados apresentados na **Tabela 1** e associando-os diretamente aos objetos do tipo Produto criados no item anterior através da invocação de métodos do tipo set
- Instanciar 1 objeto do tipo Cliente contendo os dados apresentados na **Tabela 1**
- Instanciar 1 objeto do tipo Venda contendo os dados apresentados na **Tabela 1** associando-o diretamente ao objetos do tipo Cliente criado no item anterior através da invocação de método do tipo set

B. Uma vez instanciados os objetos do Item A, execute as operações na seguinte ordem:

- Adicione cada um dos objetos ItemVenda no objeto Venda chamando o método "adicionarItemVenda"
- Chame o método *listarItensVenda()* do objeto Venda instanciado para realizar a impressão de informações conforme apresentado na **Tabela 1**.

