

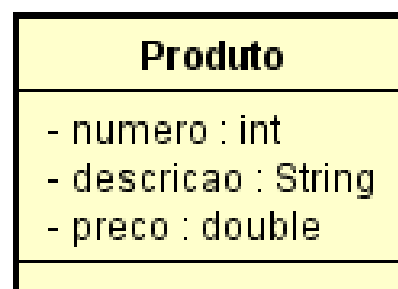
2ª Prova Prática N1

TAP – Técnicas Avançadas de Programação

Nome Estudante: _____ Data: 15/04/2021

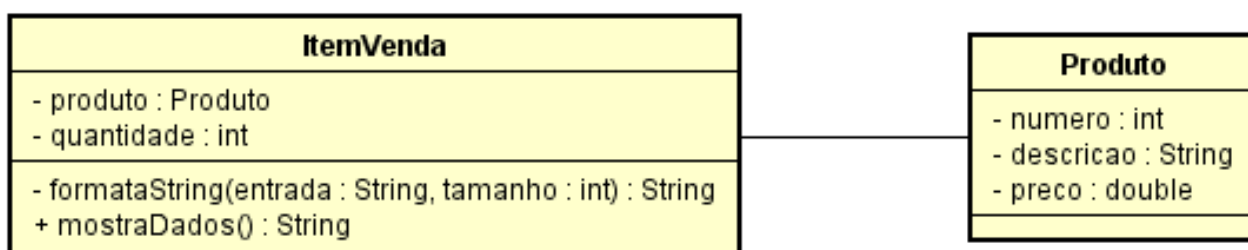
- 1) Crie uma **classe Produto** que representará um **produto em uma loja de suprimentos de informática**. Esta classe deve possuir os seguintes atributos, todos **privados**:

- Número único do produto na loja (inteiro)
- A descrição do produto (String)
- O preço unitário do produto (double)



Sua classe deve ter um método **construtor** que inicialize os três atributos e métodos **get e set** para cada um dos atributos, de forma que assim possamos manipulá-los.

- 2) Crie uma **classe ItemVenda** que representará um **item em uma venda da loja de suprimentos de informática**. Esta classe deve possuir os seguintes atributos, todos privados:
- Produto (objeto do tipo Produto)
 - Quantidade (inteiro)



Esta **classe** deverá possuir os seguintes métodos:

- Um método **privado formataString()**, que **retorna uma String** formatada e recebe como **parâmetro a String original** e um **inteiro** que será o tamanho da nova String após a formatação. Este método faz o seguinte:
 - Se o tamanho da string X passada como parâmetro é menor que o tamanho T passado como parâmetro, serão adicionados espaços em branco ao final de X até

que o tamanho de X fique igual a T. Ao final, será retornada X com os espaços em branco inseridos.

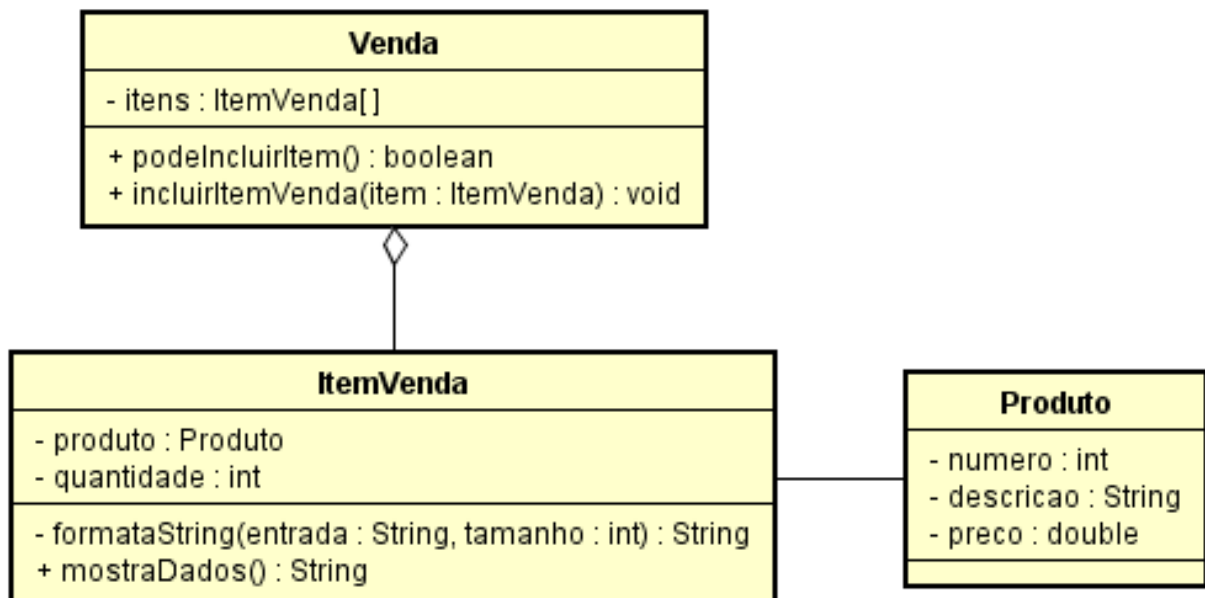
- Caso contrário, não acontece nada e será retornado como resultado a mesma String de entrada.
- Um método **público mostrarDados()**, que não recebe nada como parâmetro e retorna a String que corresponde a uma linha de um item de venda, formatada da seguinte maneira:
 - O número do produto deve ser uma String com 4 posições,
 - A descrição do produto deve ter 20 posições,
 - A quantidade do produto deve ter 4 posições,
 - O preço ajustado do item da venda (quantidade * preço), que deverá ter 10 posições.

Dica: Para formatar a linha **use o método privado formataString()**.

Esta classe também deve ter uma **interface**, um método **construtor** que inicialize os dois atributos e métodos **get e set** para cada um dos atributos, de forma que assim possamos manipulá-los.

3) Crie uma **classe Venda** que representará uma venda em uma loja de suprimentos de informática. Uma venda deve incluir o seguinte atributo privado:

- Itens (Vetor de ItemVenda)



Esta **classe** deverá possuir os seguintes métodos:

- Um método **público podeIncluirItem()**, que não recebe nada como parâmetro e retorna um **boolean** que retorna **verdadeiro** se no vetor ainda for possível incluir itens de venda e retorna **falso**, caso contrário.
- Um método público **incluirItemVenda()**, que recebe como parâmetro um item de venda a ser inserido no vetor e não retorna nada. Este item recebido como parâmetro deverá ser inserido na próxima posição livre do vetor.

Esta classe também deve ter uma **interface**, um método **construtor** que inicialize o vetor de Itens e métodos **get e set**, de forma que assim possamos manipular o atributo.

- O **construtor** desse método deve receber como **parâmetro** um **inteiro Y** e inicializar o vetor de itens com Y posições.

- 4) Escreva a **classe Main com um método main** para instanciar o vetor de Produtos da loja, como se segue:

Número	Descrição	Preço
1	Pen Drive	R\$ 22,00
2	Mouse	R\$ 38,00
3	Teclado	R\$ 110,00
4	Webcam	R\$ 400,00
5	Fone de Ouvido	R\$ 75,00
6	Tablet	R\$ 1200,00
7	Impressora	R\$ 420,00
8	Cartucho de Tinta	R\$ 48,00
9	Microfone	R\$ 65,00
10	Notebook	R\$ 4500,00

Quando o programa é iniciado, deverá ser mostrada uma **listagem de produtos na loja**. Na sequência será cadastrada uma nova venda da seguinte forma:

- A venda pode ter **no máximo 5 itens**;
- O vendedor pode inserir quantos itens quiser na venda desde que não ultrapasse a quantidade máxima permitida de itens. Se ultrapassar, deverá ser exibida a mensagem de erro: **"Não é mais possível incluir itens nesta venda!"**. Para verificar isso, use o método **podeIncluirItem()** da classe Venda. Se ainda for possível incluir itens na venda, use o método **incluirItemVenda()** da classe **Venda**;
- Sempre será perguntado se o vendedor deseja incluir mais um item. Se ele escolher que sim, será perguntado o número do item e a quantidade. Se for informado o

número de um item que não existe na loja, será mostrada a seguinte mensagem de erro: **"Não foi possível encontrar o produto!"**;

- Quando o vendedor finalizar o cadastro da venda, deverá ser informado um resumo da venda, com todos os itens selecionados e o total da venda, de acordo com a listagem a seguir. Para formatar as linhas de cada item de venda, use o método **mostraDados()** da classe **ItemVenda**.

ITENS PARA VENDA

1 - Pen Drive - R\$22.0
2 - Mouse - R\$38.0
3 - Teclado - R\$110.0
4 - Webcam - R\$400.0
5 - Fone de Ouvido - R\$75.0
6 - Tablet - R\$1200.0
7 - Impressora - R\$420.0
8 - Cartucho de Tinta - R\$48.0
9 - Microfone - R\$65.0
10 - Notebook - R\$4500.0

CADASTRAR NOVA VENDA

Numero do Item: 1
Quantidade: 2
Deseja continuar (s/n)? s
Numero do Item: 20
Não foi possivel encontrar o produto!
Deseja continuar (s/n)? s
Numero do Item: 3
Quantidade: 4
Deseja continuar (s/n)? s
Numero do Item: 8
Quantidade: 2
Deseja continuar (s/n)? s
Numero do Item: 9
Quantidade: 1
Deseja continuar (s/n)? s
Numero do Item: 2
Quantidade: 1
Deseja continuar (s/n)? s

Não é mais possível incluir itens nesta venda!
Deseja continuar (s/n)? n

+-----+			
	RESUMO DA VENDA		
+-----+			
	NUM DESCRICAO	QTD VALOR	
+-----+			
	1 Pen Drive	2 R\$44.0	
	3 Teclado	4 R\$440.0	
	8 Cartucho de Tinta	2 R\$96.0	
	9 Microfone	1 R\$65.0	
	2 Mouse	1 R\$38.0	
+-----+			
TOTAL: R\$683.0			