ATIVIDADE FINAL

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

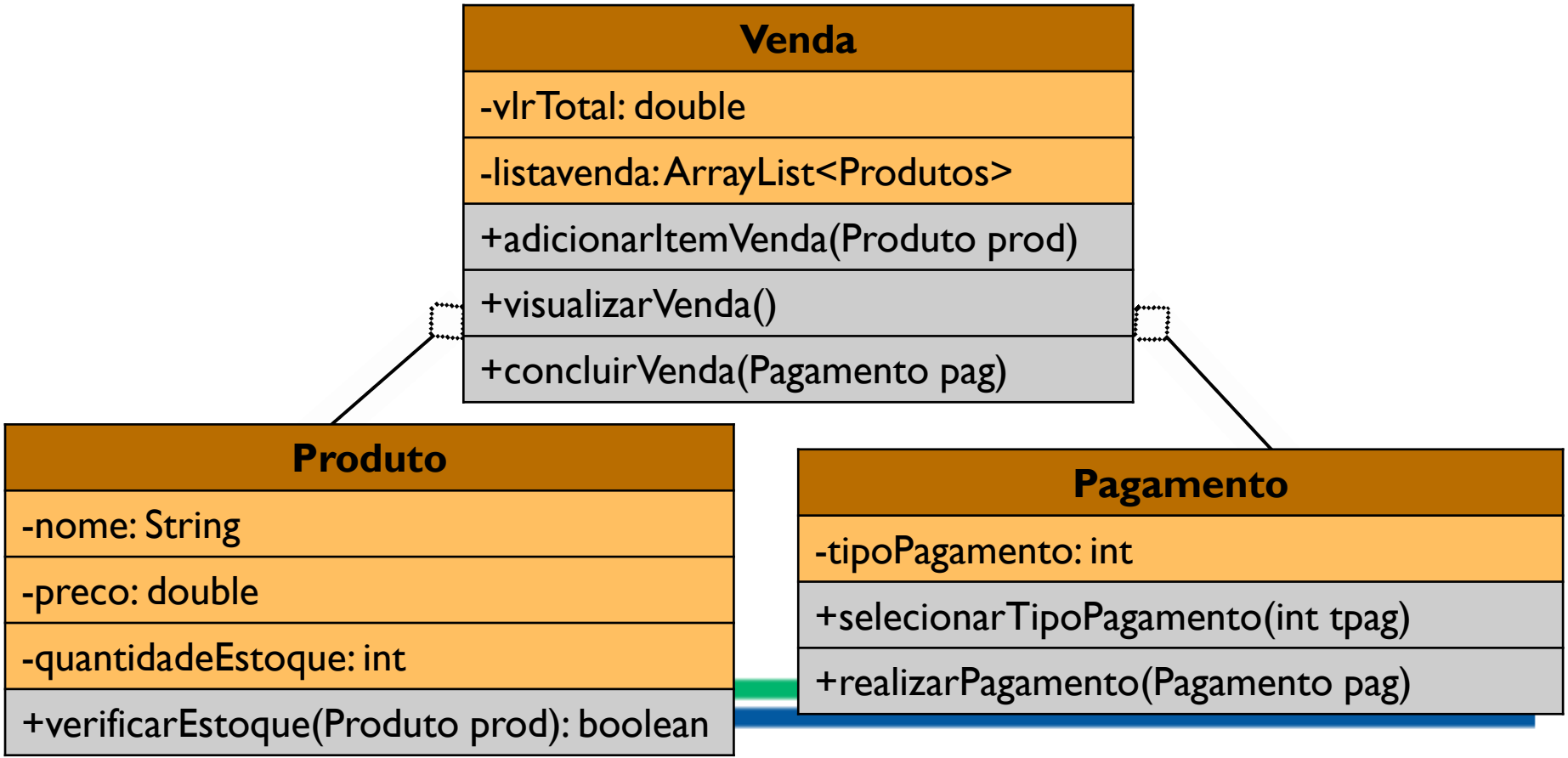
HABNER FABRÍCIO BOESING
habner.boesing@unoesc.edu.br

ATIVIDADE FINAL
CONSTRUÇÃO E RELACIONAMENTO DE CLASSES

- Implemente e programe as classes conforme a descrição a seguir:
- "O supermercado vende diferentes tipos de produtos. Cada produto tem um preço e uma quantidade em estoque. Um pedido de um cliente é composto de itens, onde cada item especifica o produto que o cliente deseja e a respectiva quantidade. Esse pedido pode ser pago em dinheiro, cheque ou cartão."

ATIVIDADE FINAL
CONSTRUÇÃO E RELACIONAMENTO DE CLASSES

- Para facilitar segue também o diagrama de classes contendo a estrutura solicitada no enunciado, especificando as classes, os atributos e seus tipos correspondentes e os métodos juntamente com parâmetros, quanto tiverem.



ATIVIDADE FINAL
CONSTRUÇÃO E RELACIONAMENTO DE CLASSES

- Produto:
  - **verificarEstoque(Produto prod)**: Este método deve receber por parâmetro um objeto do tipo Produto, o qual deve ter seu estoque analisado. O retorno deve ser do tipo boolean, resultando true caso o estoque seja maior que zero ou false caso esteja zerado.
- Pagamento:
  - **selecionarTipoPagamento(int pag)**: Este método deve receber por parâmetro um valor inteiro indicando a escolha do pagamento, 1 para Dinheiro, 2 para Cheque ou 3 para Cartão e armazenar este valor inteiro na variável tipoPagamento.
  - **realizarPagamento(Pagamento pag)**: Este método deve receber por parâmetro um objeto do tipo Pagamento e através do seu atributo tipoPagamento, exibir na tela a informação que o pagamento foi realizado juntamente com o tipo de pagamento escolhido.

ATIVIDADE FINAL
CONSTRUÇÃO E RELACIONAMENTO DE CLASSES

- Venda:
  - **adicionarItemVenda(Produto prod)**: Este método deve receber um produto como parâmetro e adicioná-lo em uma lista. Importante também que ao adicionar um novo produto nesta lista a variável vlrTotal seja atualizada somando o valor total de cada produto adicionado.
  - **visualizarVenda()**: Método sem parâmetro que tem como principal objetivo exibir na tela a lista de produtos que foram adicionados pelo método adicionarItemVenda().
  - **concluirVenda()**: Este método recebe por parâmetro um objeto do tipo Pagamento, e deve exibir o valor total da venda, assim como chamar o método realizarPagamento() para que seja exibida a opção de pagamento utilizada ao concluir a venda. Ao terminar a venda a lista contendo os produtos da venda deve ser esvaziada para que possa aceitar os produtos para uma nova venda.
- Classe Principal:
  - Nesta classe é onde serão instanciados os objetos das demais classes para que os mesmos sejam criados e posteriormente tenham seus métodos executados para demonstrar o funcionamento do processo de venda.

ATIVIDADE FINAL
CONSTRUÇÃO E RELACIONAMENTO DE CLASSES

- Getters e Setters:
  - Os atributos devem seguir as regras de encapsulamento, logo devem ser do tipo privado e os métodos Getters e Setters devem ser implementados para que seja possível alterar os valores destes atributos.
- Métodos Contrutores:
  - Os métodos construtores servem para definir quais serão os métodos solicitados para a criação do objeto, podem ser vazios, como também podem solicitar os atributos como parâmetro, dependendo a forma de implementação de cada classe.

ATIVIDADE FINAL
CONSTRUÇÃO E RELACIONAMENTO DE CLASSES

- Resultados esperados:
  - Projeto Java contendo as classes programadas de acordo com o diagrama de classes. Na programação deve constar pelo menos uma venda completa realizada de acordo com os requisitos anteriormente informados.
- Critérios avaliados:
  - Correta criação das classes.
  - Definição correta dos atributos de cada classe e relacionamentos conforme indicação do diagrama.
  - Implementação dos métodos conforme instruções repassadas.
  - Uso correto das técnicas de encapsulamento.
  - Aplicação correta dos modificadores de acesso.
  - Entrega da atividade no prazo estabelecido.
- Forma de entrega:
  - Projeto Java compactado contendo as estruturas de pastas e suas respectivas classes criadas no desenvolvimento do projeto.