

# DA1MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2021.2

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Classes, objetos, encapsulamento, UML](#) / [\[EP\] Pedidos de produtos](#)

Descrição

[Visualizar envios](#)

## [EP] Pedidos de produtos

**Data de entrega:** segunda, 14 Jun 2021, 23:59

**Arquivos requeridos:** Pedido.java ([Baixar](#))

**Tipo de trabalho:** Trabalho individual

Uma empresa precisa de um sistema para gerenciamento de pedidos. Nessa empresa, há algumas restrições para os pedidos:

- Os pedidos podem ter no máximo 5 códigos diferentes de produtos;
- Um pedido entregue não pode ser cancelado;
- Um pedido cancelado não pode ser entregue;
- Novos produtos só podem ser adicionados a pedidos que ainda não tenham sido entregues ou cancelados (e que não tenham atingido o limite de 5 produtos);
- Um pedido pode ser somado a outro. Por exemplo, o pedido B pode ser somado ao pedido A. Neste caso, todos os produtos do pedido B são incluídos no pedido A. Os códigos de produtos que estão no pedido B e já estavam em A, não são incluídos novamente (não pode ter códigos de produtos duplicados em um pedido). Por exemplo, pedido A = [11, 22, 33] e pedido B = [22, 44, 66]. Ao somar o pedido B em A, o resultado é [11, 22, 33, 44, 66] (o código 22 presente nos dois pedidos não foi duplicado).
- A soma de pedidos só é permitida se, ao final da soma, o total de produtos diferentes for no máximo 5 (que é o limite aplicado a qualquer pedido). Não há soma parcial de pedidos. Portanto, se um pedido B não pode ser integralmente somado ao pedido A, nenhuma alteração é realizada no pedido A.

Implemente a classe **Pedido**, conforme especificado a seguir. Serão apresentados os métodos necessários na classe Pedido. Esses métodos podem ser usados pelo sistema de correção e por isso devem seguir a especificação apresentada no enunciado. Você pode incluir atributos e métodos auxiliares adicionais que considerar necessários durante a implementação.

Classe **Pedido** (deve ser implementada no **pacote empresa**):

- `public boolean adicionarProduto(int codProduto)`: adiciona um produto com o código especificado conforme regras da empresa. O retorno do método é `true` se o produto for adicionado e `false` caso contrário.
- `public int[] entregar()`: realiza a entrega de um pedido conforme regras da empresa. A empresa apenas realiza entregas de pedidos que não foram cancelados e que possuem pelo menos um produto. Caso a entrega seja realizada pela empresa, o retorno do método é um vetor com os códigos dos produtos do pedido. O tamanho do vetor deverá ser igual à quantidade de produtos no pedido. Dessa forma, o vetor não possuirá posições vazias. Caso a entrega não seja realizada, o método retorna `null`.
- `public boolean cancelar()`: cancela um pedido de acordo com as regras da empresa. O retorno do método é `true` se o pedido for cancelado e `false` caso contrário.
- `public boolean somarPedido(Pedido p)`: soma o pedido no parâmetro `p` ao pedido da instância atual conforme regras da empresa. Se a soma não for possível, os pedidos não devem ser modificados. O retorno da função é `true` se a soma for realizada, ou `false` caso contrário.

**Importante:** Submeta apenas a classe pública **Pedido** descrita no enunciado (a classe deve estar no pacote "**empresa**"). A classe submetida não pode utilizar `import`, realizar impressão de dados, utilizar `java.util` e `System.out`.

## Casos de teste

Formato dos casos de teste (que aparecem ao avaliar as classes no sistema de correção automática):

Entrada:

- operações no formato: [índice pedido] [código operação] (código do produto ou índice de outro pedido)
  - operação 1: adicionar produto
  - operação 2: entregar
  - operação 3: cancelar
  - operação 4: somar pedido

Saída:

- métodos executados e saídas obtidas

[VPL](#)

[◀ \[EP\] Lâmpada Inteligente](#)

Seguir para...

[\[EP\] Operações com Strings ▶](#)



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

## Informação

[Conheça a UFABC](#)

[Conheça o NTI](#)

[Conheça o Netel](#)

## Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos



[Universidade Federal do ABC](#) - [Moodle](#) (2020)

[Obter o aplicativo para dispositivos móveis](#)