

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

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Herança, polimorfismo](#) / [\[EP\] Cartões](#)

Descrição [Visualizar envios](#)

## [EP] Cartões

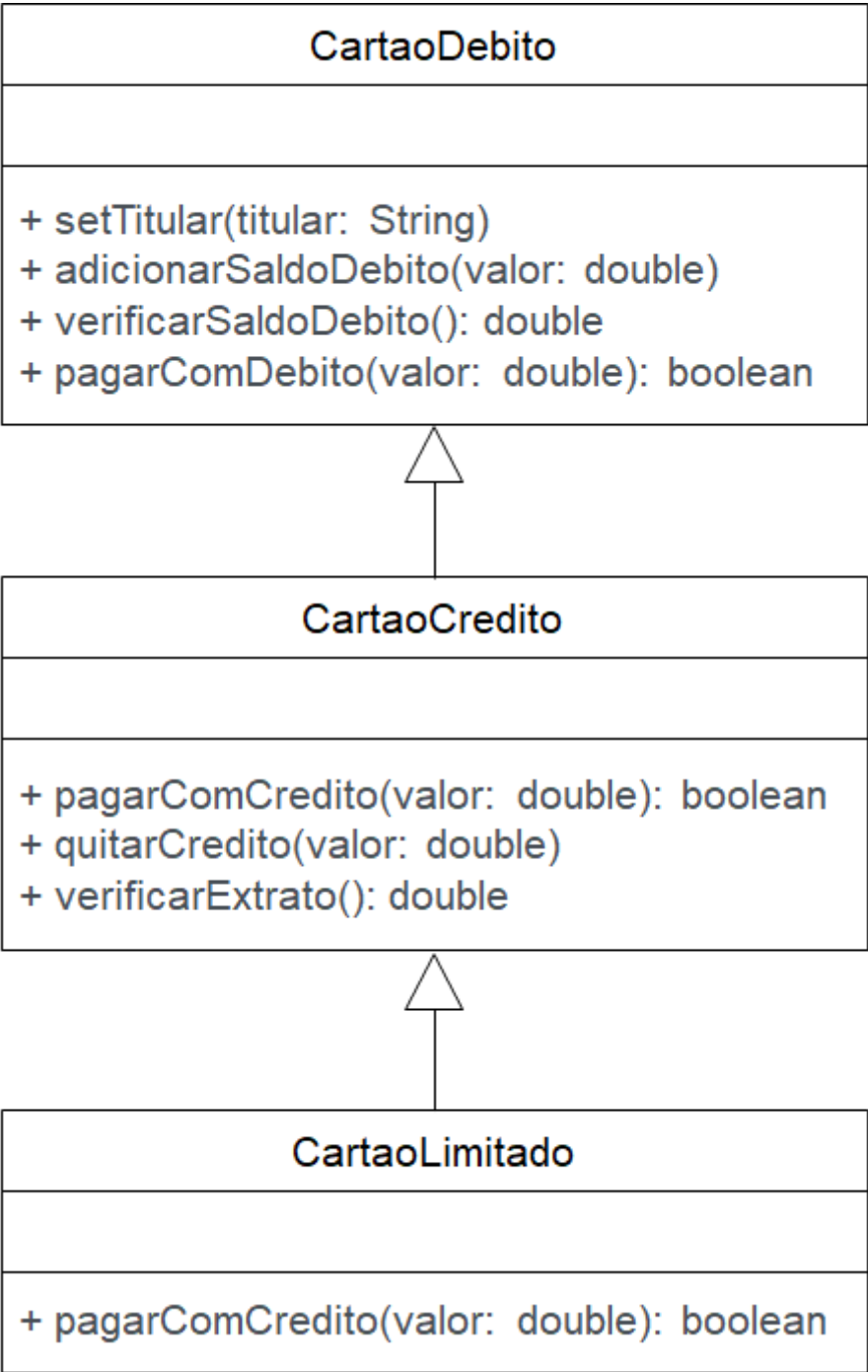
**Data de entrega:** segunda, 28 Jun 2021, 23:59  
**Arquivos requeridos:** CartaoDebito.java, CartaoCredito.java, CartaoLimitado.java ([Baixar](#))  
**Tipo de trabalho:** Trabalho individual

## Cartões

Esta é uma versão revisada do exercício Cartões de Guilherme Aldeia, Paulo H. Pisani. POO - QS/2020

## Tarefa

Nesse exercício, você criará as classes CartaoDebito, CartaoCredito e CartaoLimitado, conforme diagrama a seguir (**todas as classes devem estar no pacote cartões**):



Classe **CartaoDebito**:

- `setTitular(titular: String)`: armazena o nome do titular do cartão
- `adicionarSaldoDebito(valor: double)`: aumenta o saldo para débito. O saldo inicial é zero (quando o cartão é instanciado).
- `verificarSaldoDebito()`: `double` : retorna o saldo disponível

- `pagarComDebito(valor: double): boolean` : caso tenha saldo disponível, debita o valor passado no parâmetro. Quando o débito ocorre, retorna `true`, caso contrário, retorna `false`.

Classe **CartaoCredito**:

- `pagarComCredito(valor: double): boolean` : aumenta o *valor devido com crédito* e retorna `true`. Quando a classe é instanciada, o valor devido com crédito é zero.
- `quitarCredito(valor: double)`: quita o *valor devido com crédito* (seria como pagar a fatura do cartão). Após pagar, o crédito devido é diminuído (dependendo do valor passado no parâmetro do método).
- `verificarExtrato()`: `double` : retorna o *valor devido com crédito*.

Classe **CartaoLimitado**:

- `pagarComCredito(valor: double): boolean` : funcionamento similar ao método da classe `CartaoCredito`, mas agora possui um limite de crédito de \$200,00 (para pagamentos com crédito). Portanto, se o valor devido com crédito somado ao valor do pagamento passar do limite, o pagamento não é realizado. Nesse caso, o método retorna `false`. Caso contrário, retorna `true`.

**Importante:** Submeta apenas as classes públicas `CartaoDebito`, `CartaoCredito` e `CartaoLimitado` (todas no **pacote cartoes**). O programa principal já existe no sistema de correção automática. As classes submetidas não podem realizar impressão de dados ou utilizar `import`. Todas as classes devem possuir o construtor sem parâmetros, que será usado pelo sistema de correção automática para instanciar as classes.

## Casos de teste

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

### Entrada

- quantidade de cartões
- lista com tipos dos cartões a serem instanciados (`debito`, `credito`, `limitado`)
- operações no formato: `[índice cartão] [nome operação] (valor)`

### Saída

- classes instanciadas
- métodos executados e saídas obtidas (o caso de teste indica o índice do cartão envolvido em cada operação)

[VPL](#)

[◀ \[EP\] Quadriláteros](#)

Seguir para...

[\[EP\] REC - Vínculo com a Universidade ▶](#)



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](#)