

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

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Construtores, sobrecargas](#) / [\[EP\] ContaCorrente \(construtor\) 2](#)

Descrição

[Visualizar envios](#)

[EP] ContaCorrente (construtor) 2

Data de entrega: quarta, 16 Jun 2021, 23:59

Arquivos requeridos: ContaCorrente.java ([Baixar](#))

Tipo de trabalho: Trabalho individual

Conta Corrente (construtor) 2

Este exercício é uma versão adaptada e estendida do exercício *Conta Corrente com Construtores* de Guilherme Aldeia (2020).

Escreva uma classe com o nome **ContaCorrente** para representar uma conta corrente conforme especificado a seguir.

Construtores da classe **ContaCorrente** (todos os construtores são públicos):

- **ContaCorrente(String nome)**
 - construtor que armazena o nome do titular da conta e inicializa o saldo com zero.
- **ContaCorrente(String nome, double saldo)**
 - construtor que armazena o nome do titular da conta e inicializa o saldo com o valor especificado no parâmetro.

Métodos da classe **ContaCorrente**:

- **public double getSaldo()**
 - retorna o saldo da conta.
- **public String getTitular()**
 - retorna uma string com o nome do titular da conta.
- **public boolean depositar(double valor)**
 - recebe um valor para ser depositado, verifica se é um valor positivo válido. Apenas valores positivos podem ser depositados. O método deve retornar true ou false, indicando se o depósito ocorreu (true) ou não (false).
- **public boolean sacar(double valor)**
 - recebe um valor para ser retirado da conta. Apenas valores positivos podem ser retirados. Além disso, a operação só ocorre se a conta tiver saldo disponível. O método não realiza retirada parcial, portanto, é necessário que a conta tenha saldo disponível suficiente para que a operação seja realizada. O método deve retornar true ou false, indicando se a retirada ocorreu (true) ou não (false).
- **public static boolean transferir(ContaCorrente de, ContaCorrente para, double valor)**
 - Realiza a transferência do valor informado da primeira conta (de) para a segunda (para), se isso for possível. Se não for possível, o saldo de nenhuma das contas deve ser alterado. O método deve retornar true ou false, indicando se a transferência ocorreu (true) ou não (false).
- **public static int getQuantidadeContas()**
 - Retorna a quantidade de instâncias de ContaCorrente (dica: use um campo static para realizar a contagem de instâncias)

Importante: O programa principal já existe no sistema de correção automática. Submeta apenas a classe **ContaCorrente**. A classe ContaCorrente não pode realizar impressão de dados, utilizar import, 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:

- nomes das operações executadas (a cada conta instanciada, o programa de correção armazena em um vetor):
 - instanciar_nome [nome_titular]
 - instanciar_nome_saldo [nome_titular] [saldo]

- getTitular [índice da conta]
- getSaldo [índice da conta]
- depositar [índice da conta] [valor]
- sacar [índice da conta] [valor]
- transferir [índice da conta 1] [índice da conta 2] [valor]
- getQuantidadeContas

Saída:

- resultados obtidos após executar cada uma das operações

◀ [EP] Paralelogramo

Seguir para...

[EP] Frações (construtor) 2 ▶

[VPL](#)



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](#)