

Checkpoint 2

Deadline: 05/05/2021

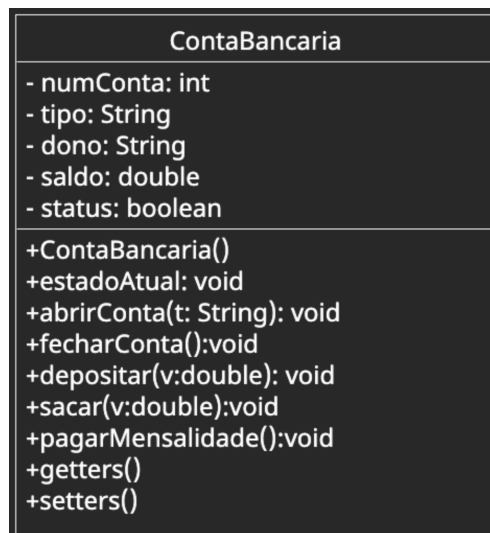
Objetivos:

- Trabalhar com os conceitos de Classes, Objetos, Métodos e Construtores em Java;
- Aplicar os conceitos de Estruturas Condicionais e de Repetição na resolução dos problemas

Descrição

Vamos imaginar a seguinte situação: José e Maria possuem um dinheiro e gostariam de depositá-los em uma conta bancária. Porém, sabemos que existem dois tipos de conta: Conta Corrente e Conta Poupança. José possui R\$ 300,00 e gostaria de depositar em uma conta poupança. Maria tem R\$ 500,00 e gostaria de depositar em uma Conta Corrente. Porém, ambos precisam abrir uma conta bancária para depositar o dinheiro.

Considere o seguinte diagrama de Classe:



A classe **ContaBancaria** terá os seguintes atributos:

- **numConta** - representa o número da conta com um número inteiro
- **tipo** - o tipo representa o tipo de conta que o cliente pode ter: "cc" para conta corrente ou "cp" para conta poupança
- **dono** - atributo que armazenará o nome do titular da conta
- **saldo** - atributo que armazena o saldo da conta
- **status** - atributo que determina se a conta está ativa ou não através de um valor lógico

Restrições

Ao abrir uma conta corrente, o cliente ganhará R\$ 50,00 como bônus e R\$ 150,00 para conta poupança.

Sempre que uma conta for criada, o método Construtor iniciará o saldo com zero e o status será falso.

O método `estadoAtual` deverá imprimir as informações da conta: conta, tipo, dono, saldo, status.

Checkpoint 2

Deadline: 05/05/2021

O método `abrirConta` deverá receber por parâmetro o tipo da conta, configurar o tipo (cc ou cp) e o status (verdadeiro ou falso), além de atribuir os valores de bônus, dependendo do tipo da conta (conta corrente ou conta poupança).

Para encerrar uma conta, o saldo deverá estar zerado (R\$ 0,00), ou seja, não pode ter saldo positivo e nem negativo.

O método `depositar` permite o cliente realizar depósitos, porém, é preciso verificar se a conta está aberta (ativa), ou seja, o status deve ser verdadeiro.

O método `sacar` permite ao cliente realizar saques da conta, porém, é preciso verificar se a conta está aberta e ter saldo positivo. Não havendo saldo, o cliente não poderá sacar nenhum valor.

O método `pagar mensalidade` ao ser invocado deverá debitar do cliente o valores de R\$ 12,00 para conta corrente e R\$ 20,00 para conta poupança. Vale ressaltar que o cliente só poderá pagar a mensalidade se a mesma estiver ativa.

Classe de Teste

Crie um classe de Teste que contenha dois objetos: um para ilustrar a criação de uma conta corrente e uma conta poupança.

- Após definir os valores dos atributos, invoque o método `estadoAtual` para visualizar o status da conta.
- Considere que ambas estejam **abertas**.
- Simule um depósito de R\$ 100,00 para a conta corrente
- Simule um depósito de R\$ 500,00 para a conta poupança
- Simule um saque de R\$ 100,00 na conta poupança
- Simule um saque de R\$ 150,00 na conta corrente
- Simule um saque de R\$ 1000,00 na conta poupança
- Simule o fechamento de uma das contas