ECM251 - Linguagens I

Laboratório - Encapsulamento e Construtores

Prof. Murilo Zanini de Carvalho
Prof. Tiago Sanches da Silva

Antes de começar!

Clone seu repositório do Github

- Lembre-se sempre antes de iniciar uma aula, clonar seu repositório remoto e realizar as atividades nele.
- Para cada atividade desenvolvida, criar um novo diretório.

Retirado de (https://miro.medium.com/max/4000/0*MZMl76wKo2FQLqG0.png), em 07/03/2021

Nosso Problema!

MauaBank

Vamos construir uma aplicação chamada MauaBank.

Ela vai ser a chave para um Thintech(mais detalhes sobre thintechs: https://thintech.co.uk), que nos procurou para realizar essa implementação.

Retirado de (photo-1579621970795-87facc2f976d), em 15/03/2021

MauaBank

A construção da aplicação vai acontecer de forma gradual, ao longo das aulas.

Cada novo conceito que for apresentado, será utilizado para melhorar/refatorar nosso código desenvolvido até o momento.

Mesmo utilizando o controle de versão, vamos versionar nosso projeto das modificações realizadas por laboratório, não apenas as modificações gerais realizadas no projeto.

Retirado de (photo-1560746420-1b4dc6d92d17), em 15/03/2021

Requisitos do Projeto

Requisitos do Projeto - Versão 1.0

Para esta versão do projeto, a aplicação vai rodar em modo console, localmente. Ainda assim, ela deve respeitar algumas premissas:

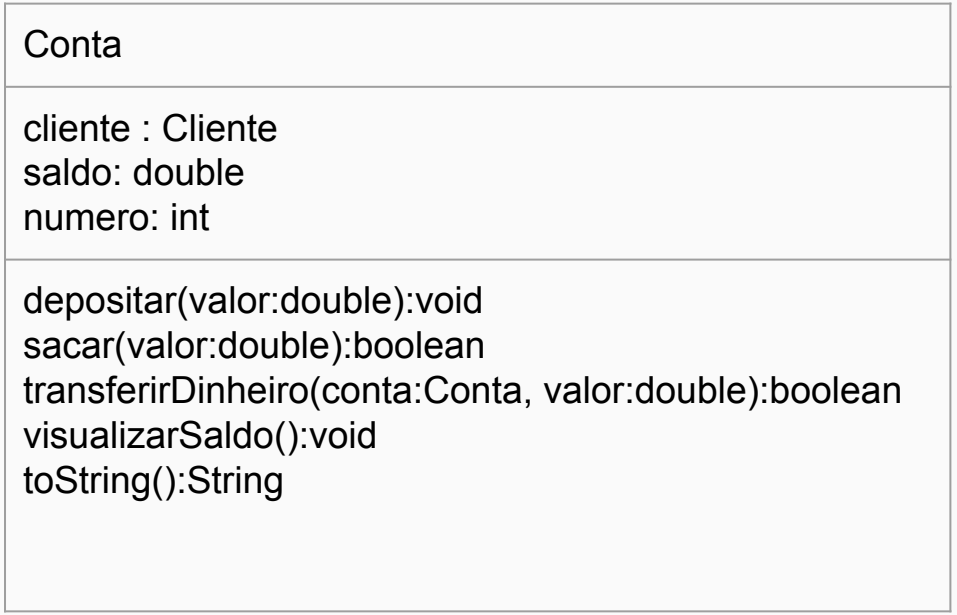
- Apenas um usuário logado no sistema, deve ser sua conta que fica ativa no sistema.
- O usuário pode pagar contas no sistema, sacar dinheiro e transferir dinheiro para outras contas (por hora, apenas retirar o dinheiro da conta do usuário).
- O sistema deve ser representado por uma classe dedicada para ele. Toda exibição de dados deve acontecer apenas nessa classe (entrada e saída).
- Sua função main(), deve apenas inicializar o sistema.
- As contas pagas devem possuir uma data de vencimento. Se a conta já estiver vencida, verificar qual o valor de multa deve ser aplicado ao valor dela.

Diagrama de Classes

Diagrama de Classes

Servem para representar uma classe, com seus atributos e seus métodos.

Ela faz parte de uma linguagem para descrição de classes chamada UML (mais em breve).



HandsOn!

Hands On

Criar a classe Sistema, que deve possuir o método executar(), que será o responsável pela interação no sistema.

Implementar a classe Titulo, que representa uma conta que deve ser paga.

Mover toda a interação com o usuário para dentro da classe Sistema.

Realizar as modificações necessárias no sistema para encapsular a classe Conta e a classe Cliente.

Entrada de Dados com o Usuário pelo Teclado

Documentação Oficial: https://docs.oracle.com/en/java/javase/11/

https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Scanner.html

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

LocalDate

Para trabalhar com datas. Verificar sua documentação: https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/time/LocalDate.html

Perguntas?