

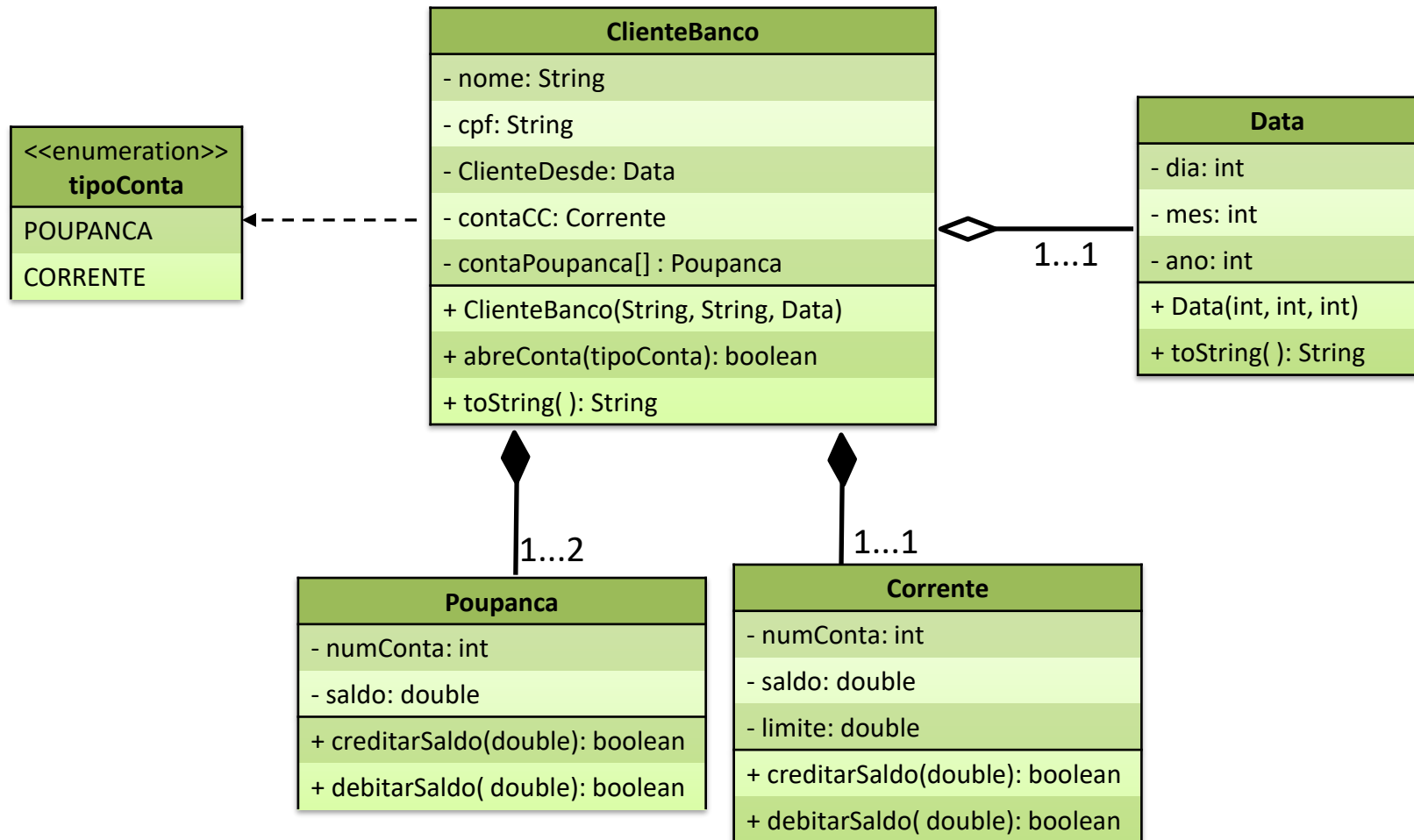
Programação Orientada a Objetos

Professor Isaac

Exercício de Herança

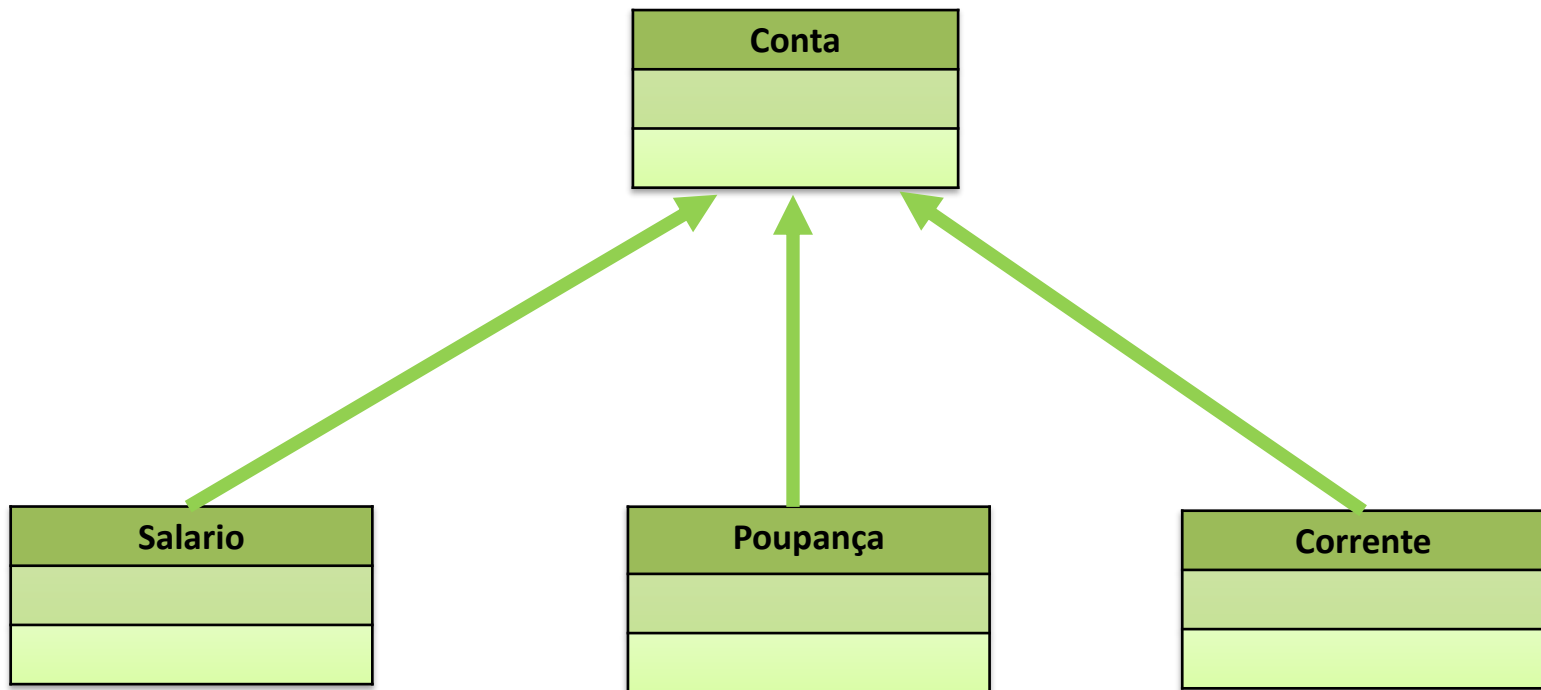
Exercício de Herança

- Reproveite o seu código da aula passada e **crie** uma **classe** chamada **Conta** para ser a **classe base** das classes Poupança e Corrente (utilize **herança**).



Exercício de Herança

- Reaproveite o seu código da aula passada e **crie** uma **classe** chamada **Conta** para ser a **classe base** das classes Salario, Poupança e Corrente (utilize **herança**).



Exercício de Herança

- O banco trabalha com três tipos de conta:
- ❑ Salário:
 - cobra taxa de 1% a cada débito realizado
 - Não permite débitos que deixem a conta com saldo negativo
- ❑ Corrente:
 - cobra taxa de 1% a cada débito realizado
 - Permite um saldo negativo de até o valor do atributo limite.
- ❑ Poupança:
 - Não cobra taxas de operações realizadas.
 - Não permite débitos que deixem a conta com saldo negativo.

São as mesmas regras da aula passada

Exercício de Herança

- Use o ENUM `tipoConta` no método `abreConta` para selecionar qual conta será aberta.
- Reutilizar os códigos da `Atividade 02` na classe `Conta`.
- Use o código disponível na `Aula 03_02` da classe `Data` dado nos exercícios 01 e 02.
- Permita o acesso ao conteúdo de `alguns atributos` criando o método `get`.
- Permita que se modifique o conteúdo de `alguns atributos` criando o método `set`.
- Crie os construtores que achar necessário.
- Crie mais métodos ou atributos que considere necessário.
- Declare com `final` atributos que considere constante.

São as mesmas regras da aula passada

Exercício: No método main()

- Crie objetos do tipo **clienteBanco** com diferentes tipos de contas e acesse os métodos dos objetos.

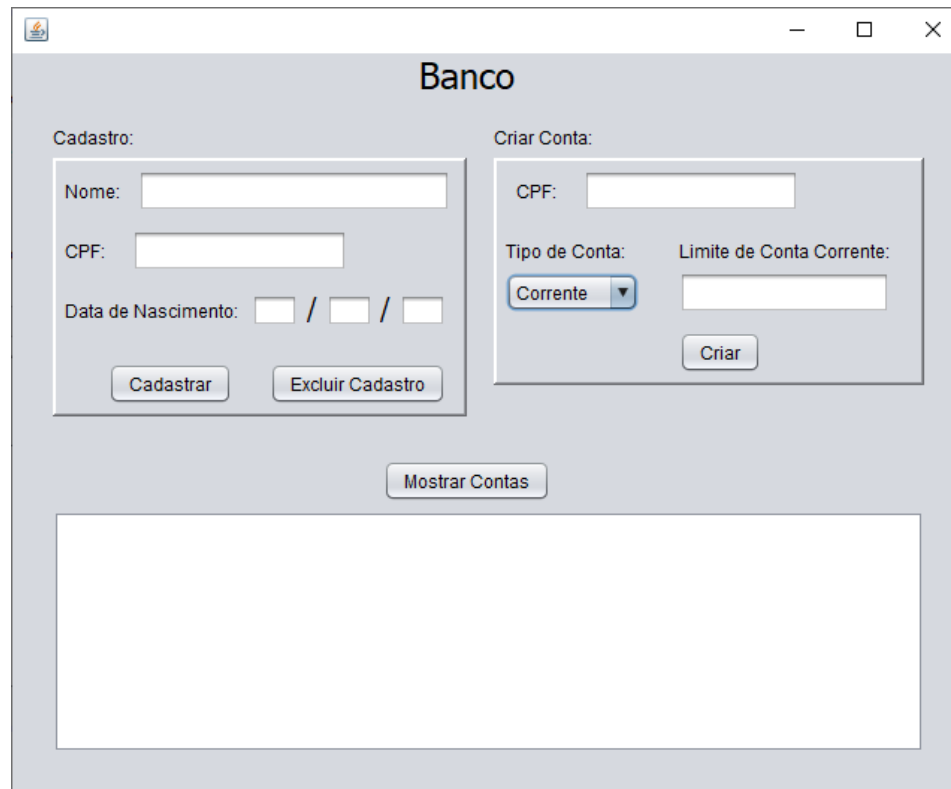
Quem quiser pode criar uma interface gráfica em vez de testar com o main()

Exercício extra

Exercício com Interface gráfica

Crie uma interface gráfica do usuário (GUI) para cadastrar clientes, excluir clientes e criar contas dos clientes.

- Exemplo:



O exemplo mostra uma janela de aplicativo intitulada "Banco". A interface é dividida em duas seções principais: "Cadastro" e "Criar Conta".

Cadastro:

- Nome:
- CPF:
- Data de Nascimento: / /
- Botões: "Cadastrar" e "Excluir Cadastro"

Criar Conta:

- CPF:
- Tipo de Conta:
- Limite de Conta Corrente:
- Botão: "Criar"

Abaixo dessas seções, há um botão "Mostrar Contas" e uma área vazia para a exibição dos dados.

Observação: Esta interface é apenas um exemplo, crie a **GUI** que você considere que possua **melhor usabilidade**

Exercício com Interface Gráfica

- ❑ Utilize as Classes das contas (Salario, Poupança e Corrente) para criar a conta.
- ❑ Pode adicionar mais componentes ou Telas.
- ❑ Crie um Executável do seu projeto (.jar)
- ❑ Entregue o projeto do Netbeans ou Eclipse comprimido como ZIP.

The image shows a Java Swing window titled "Banco". The window contains two main panels for user interaction:

- Cadastro:** A panel for user registration with fields for "Nome:", "CPF:", and "Data de Nascimento:" (formatted as MM/DD/YY). Below these fields are two buttons: "Cadastrar" and "Excluir Cadastro".
- Criar Conta:** A panel for creating a new account with fields for "CPF:", "Tipo de Conta:" (a dropdown menu currently showing "Corrente"), and "Limite de Conta Corrente:". Below these fields is a "Criar" button.

At the bottom center of the window is a "Mostrar Contas" button, and below it is a large empty rectangular area, likely intended for displaying a list of accounts.

Criando o Arquivo Executável

Para criar um executável no Netbeans siga os seguintes passos:

1. Vá na aba **Arquivo -> propriedade do projeto**
2. Abrirá uma janela, selecione **Encapsulamento**
3. Check as opções: **Comprimir Arquivo JAR, Contruir JAR após Compilação, Copiar Bibliotecas Dependentes**
4. Clique em **OK**.
5. Clique em **Limpar e Contruir** ou pressione **Shift + F11**

Dentro do seu Projeto haverá um arquivo chamado: **<nome do projeto>.jar**