Aula 5

Modelando um sistema

Orientação a Objetos : Criando um sistema bancário:

- Vamos supor que estamos criando um sistema para um banco;
- Inicialmente vamos modelar uma das classes mais importantes para nosso sistema que é a conta dos clientes;

Algumas perguntas sobre o modelo de nosso sistema:

- O que toda conta tem e é importante para nós?
- O que toda conta faz que podemos modelar inicialmente?
- Quais são as operações possíveis envolvendo uma conta bancária?

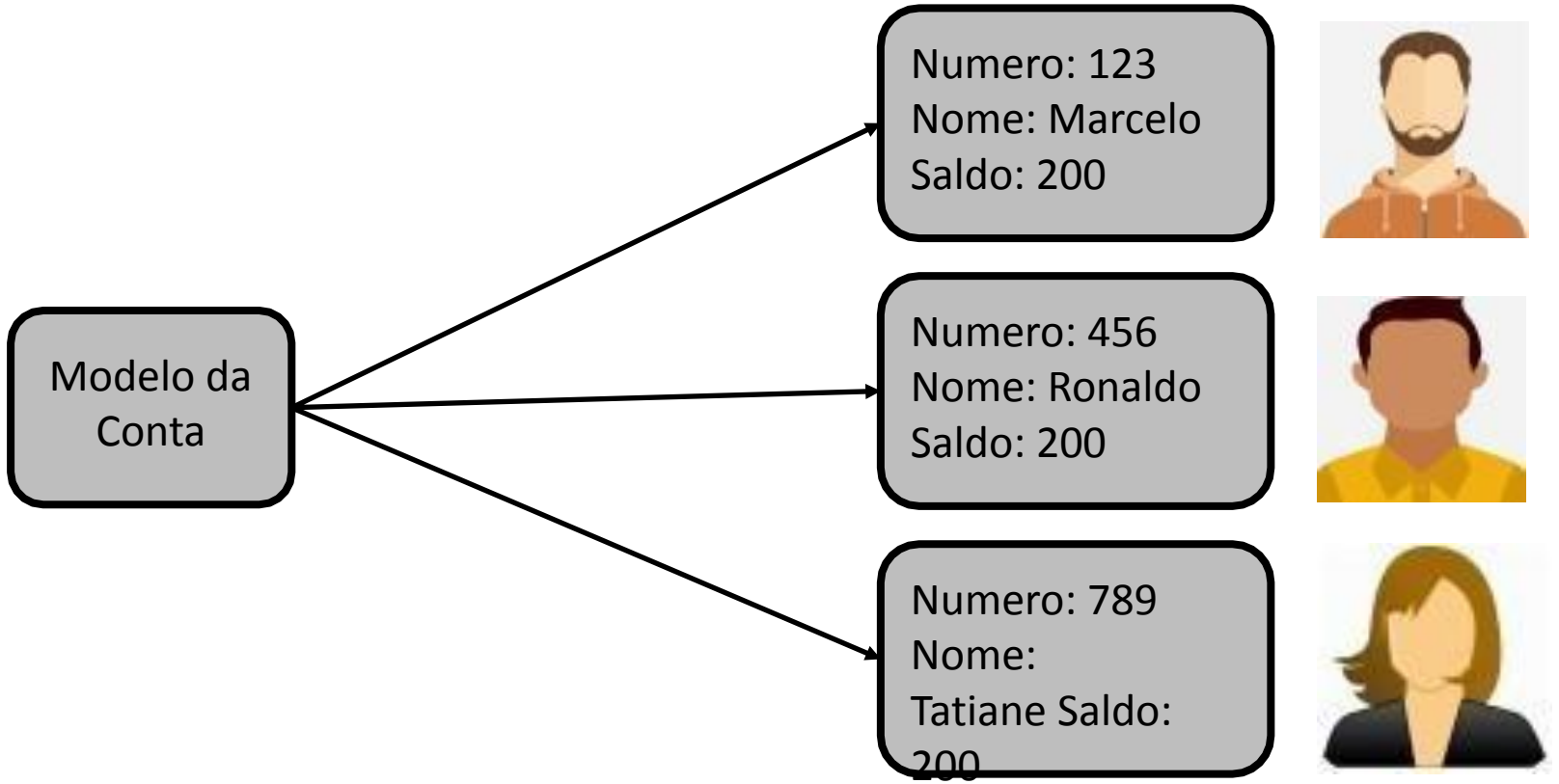
Modelando um sistema

- O que toda conta tem e é importante para nós?
  - Número da conta;
  - Nome do titular da conta;
  - Saldo.

- O que toda conta faz?
  - Saque (de uma quantidade X);
  - Depósito (de uma quantidade X);
  - Imprime o nome do titular da conta;
  - Imprime o saldo atual;
  - Transfere quantias de uma conta para outra.

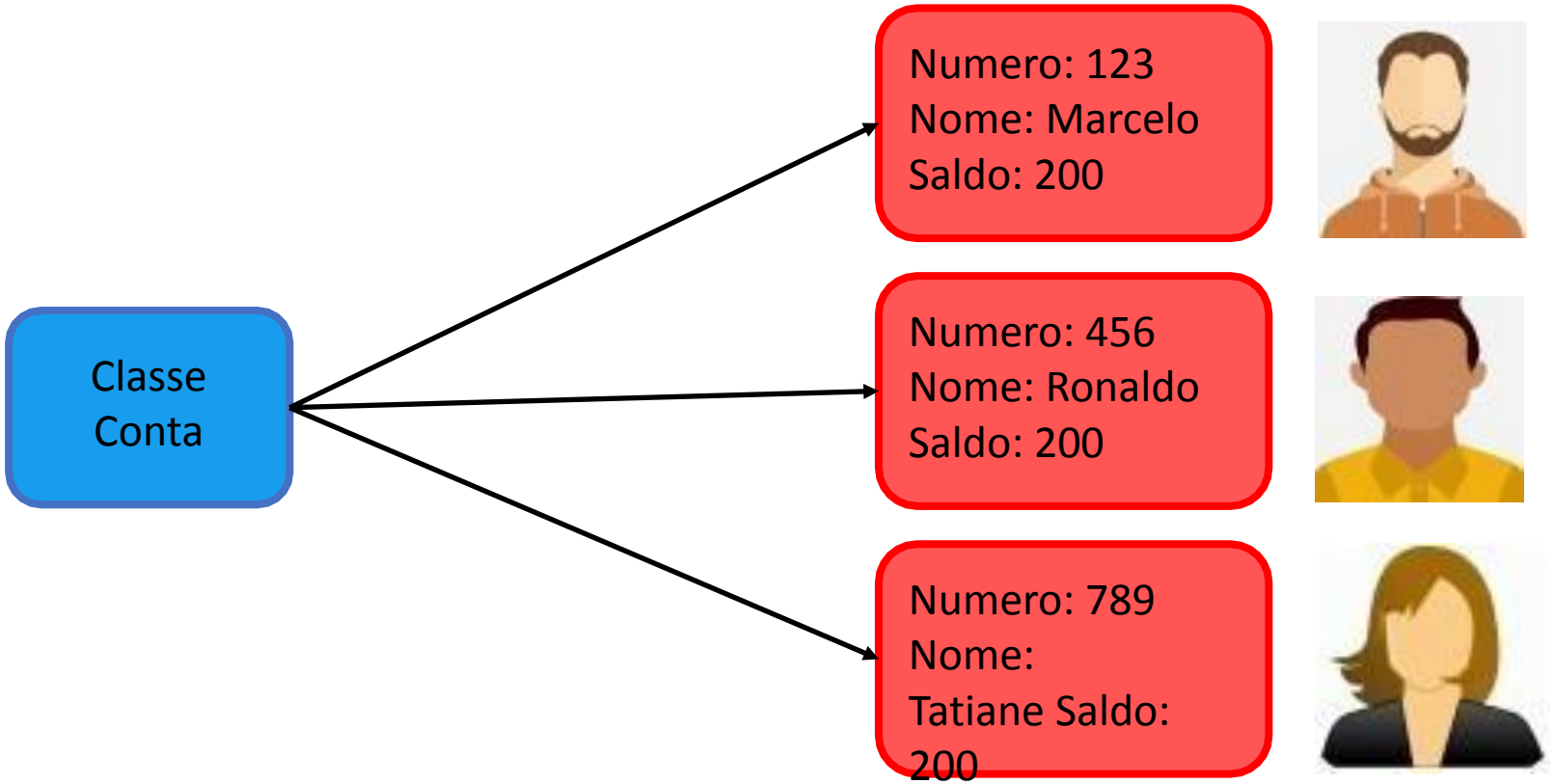
Modelando um sistema

Com esse modelo podemos abrir várias contas diferentes



Modelando um sistema

O modelo da conta, é o que chamamos de **Classe** e o que podemos construir partir desse modelo, damos o nome de **Objeto**



Implementando as classes -

Vamos começar apenas com o que uma Conta tem, e não o que ela faz. Transcrevendo o modelo anterior para um código em Java, temos a seguinte estrutura :

```
public class Conta {
   // o que uma conta tem
   int numero;
   String titular;
   double saldo;
   }
```

Construindo o

Se quisermos acessar a classe que acabamos de criar em nosso código principal, precisamos fazer :

```
public class Principal {
   public static void main(String[] args) {
      Conta minhaConta = new Conta();
      }
}
```

Construindo o objeto - Instanciação de

- Sintaxe:

Cria o objeto em memória!

```
<nomeDoModelo> = new <construtor> ([argumentos]);
```

- Exemplo:

```
professorJose = new Professor();
```

- Declarando e instanciando:

```
<nomeDoModelo> nomeDoObjeto = new <construtor> ([argumentos]);
Professor professorJose = new Professor("José Renato");
```

Acessando um

Através da variável minhaConta, podemos acessar o objeto recém criado para alterar seu titular, seu saldo, etc.

```
public class Principal {
    public static void main(String[] args) {
        Conta minhaConta = new Conta();
        minhaConta.titular = “Pedro”;
        minhaConta.saldo = 1000.0;
        minhaConta.numero = 123;
        System.out.println(“Saldo atual: ” +
        minhaConta.saldo + “Nome do titular:
        ” + minhaConta.titular);
        }
}
```

Métodos da classe -

Dentro da classe também declaramos quais as ações que ela realiza:

```
public class Conta {
   double saldo;
   // ... outros atributos da classe ...
   public void sacar(double valor) {
      double novoSaldo = this.saldo - valor;
      this.saldo = novoSaldo;
      }
}
```

Métodos da classe -

Podemos retornar ao cliente se ele pode fazer o saque ou não

```
public class Conta {
    // ... outros métodos e atributos da classe ...
    public boolean sacar(double valor) {
        if (this.saldo < valor) {

            return false;

        } else {
            double novoSaldo = this.saldo - valor;
            this.saldo = novoSaldo;

            return true;

        }
    }
}
```

Exercícios

Implemente o método transferir():

- Escreva o método transferir() que envia uma parte do saldo de uma conta X para uma conta Y;
- Requisitos: O método transferir() deve ser um método da classe Conta;
- Validação: Verifique se a transferência foi feita imprimindo os novos valores de saldo das contas X e Y na tela.