

Aula 3

Classes -

- As classes são projetos de um objeto (É preciso utilizar abstração):
 - Atributos(Propriedades);
 - Métodos(Comportamentos);



= Classe



= Objeto

Classes -

- Uma classe geralmente representa um substantivo;
 - Uma pessoa, um lugar ou algo que seja “abstrato”.
- Características das classes:
 - Toda classe possui um nome;
 - Possuem visibilidade, exemplo: public, private, protected;
 - Possuem membros como: Características e Ações;
 - Para criar uma classe basta declarar a visibilidade + digitar a palavra reservada class + NomeDaClasse + abrir e fechar chaves { }.

Classes -

- Os atributos são as propriedades de um objeto;
- Conhecidos como variáveis ou campos;
- Definem o estado de um objeto;
- Esses valores podem sofrer alterações.

Classes -

- São ações ou procedimentos;
- Podem interagir e se comunicar com outros objetos;
- A execução dessas ações se dá através de mensagens;
- Solicita ao objeto que seja executada uma rotina;
- Boa prática: sempre usar verbos para os nomes dos métodos.

Modelando um sistema bancário

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:

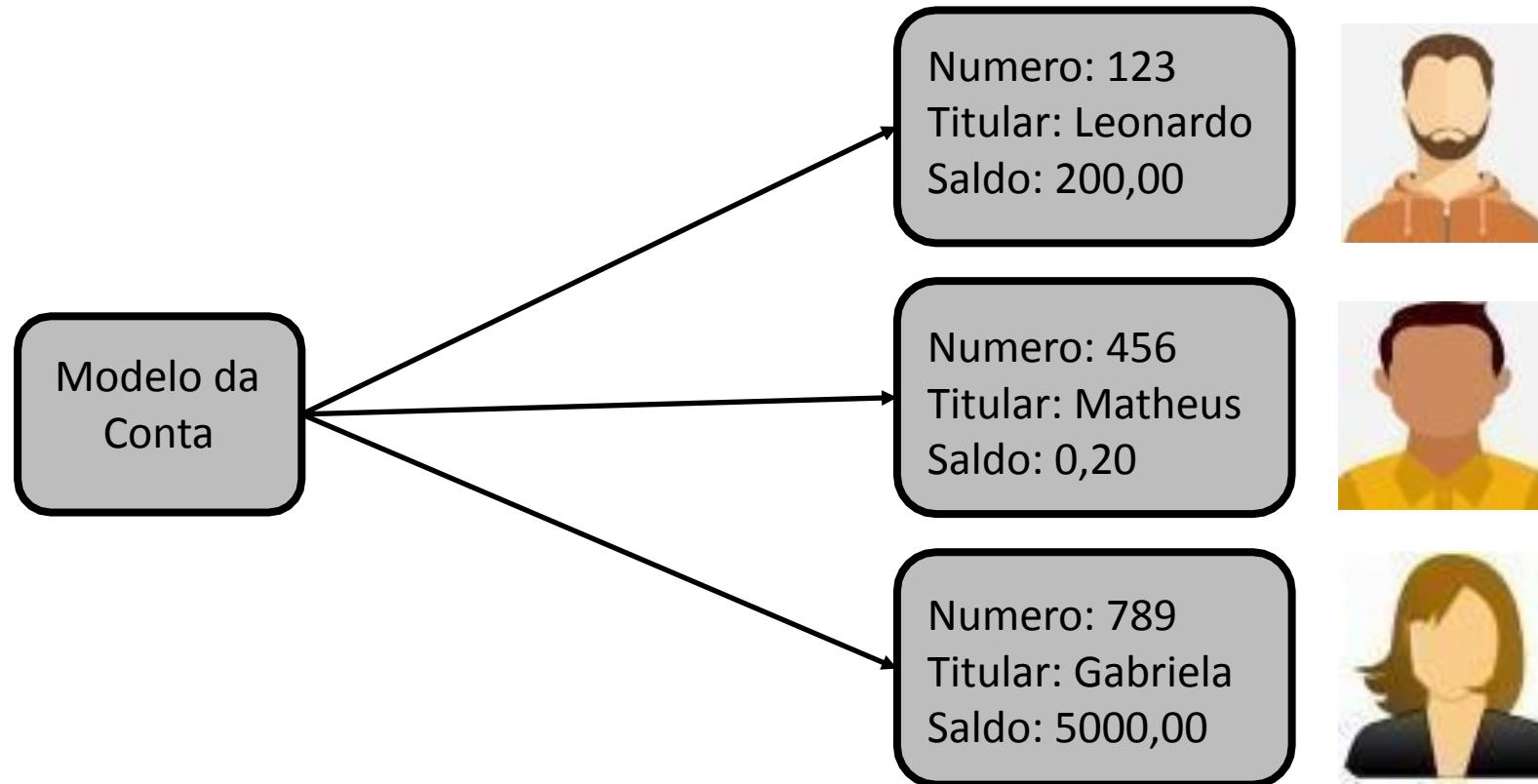
- O que toda conta tem e é importante para nós?
- O que toda conta faz que podemos modelar inicialmente?
- Quais são as operações possíveis envolvendo uma conta bancária?

Modelando um sistema bancário

- 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.

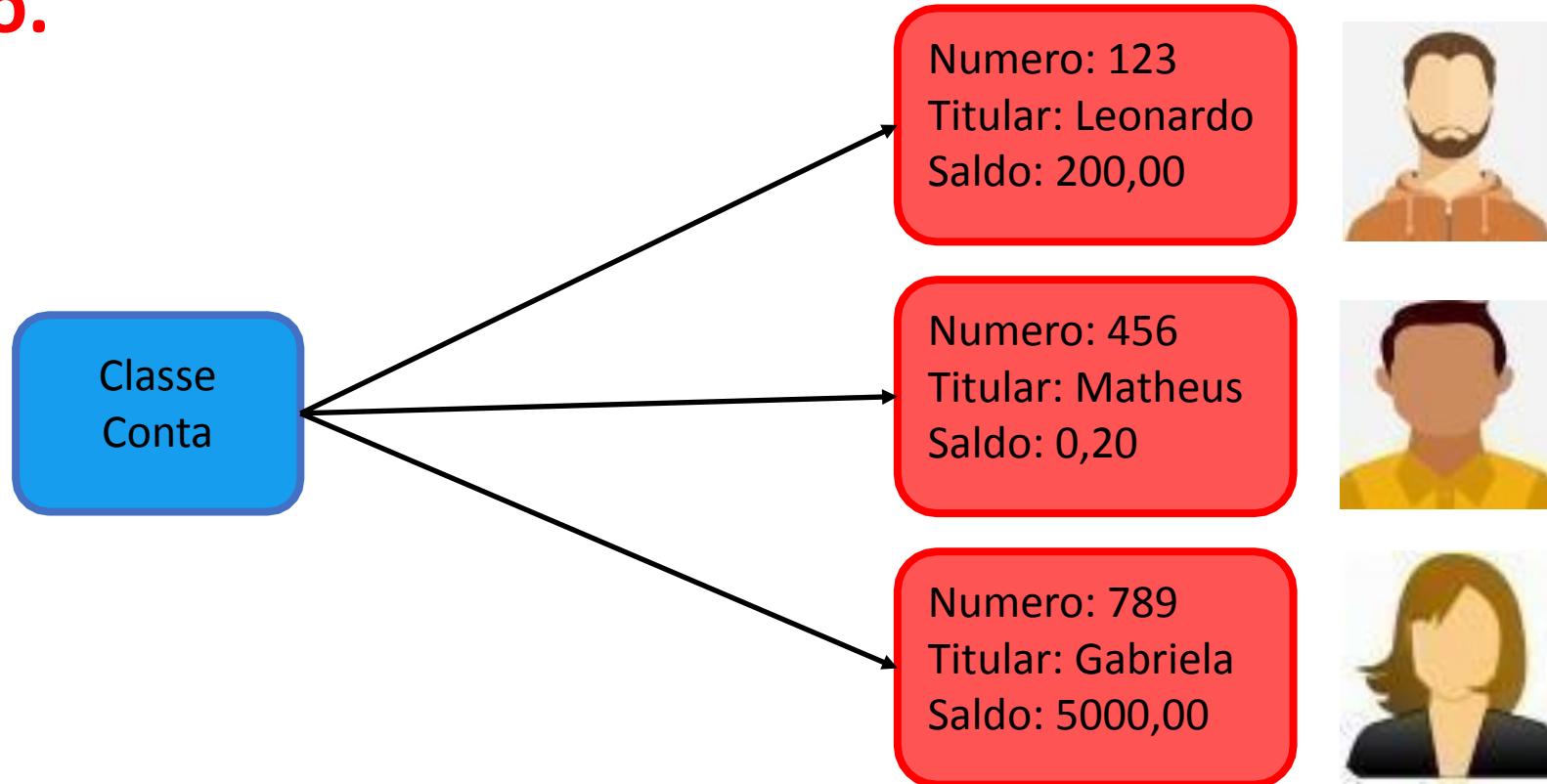
Modelando um sistema bancário

Com esse modelo podemos abrir várias contas diferentes:



Modelando um sistema bancário

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 - Atributos

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 {  
    //atributos  
    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!

Modelo <nomeDoObjeto> = new <construtor> ([argumentos]);

Classe <apelido> = new <Classe>();

- Exemplo:

Professor professor1;

professor1 = new Professor();

OU

Professor professor1 = new Professor();

- Declarando e instanciando:

Professor professorJose = new Professor(147, "Rafaela", 600.00);

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;  
  
        System.out.println("Saldo atual: " + minhaConta.saldo);  
    }  
  
}
```

Modificadores de Acesso

Visibilidade	public	protected	default	private
A partir da mesma classe	✓	✓	✓	✓
Qualquer classe no mesmo pacote	✓	✓	✓	✗
Qualquer classe filha no mesmo pacote	✓	✓	✓	✗
Qualquer classe filha em pacote diferente	✓	✓	✗	✗
Qualquer classe em pacote diferente	✓	✗	✗	✗

Variáveis compartilhadas de classe

Podemos também passar a variável totalDeContas para dentro da classe e incrementá-la no construtor:

```
public class Conta {  
    private int totalDeContas;  
    //...  
    public Conta(){  
        this.totalDeContas += 1;  
    }  
}
```

Variáveis compartilhadas de classe

Para solucionar de forma definitiva o problema, precisamos adicionar o atributo **static**, que torna a variável `totalDeContas` compartilhada entre todos os objetos da classe:

```
public class Conta {  
    private static int totalDeContas;  
    //...  
    public Conta(){  
        this.totalDeContas += 1;  
    }  
}
```