Sejam bem-vindos!

PROGRAMAÇÃO ORIENTADA A OBJETOS

UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Diogenes Carvalho Matias

Formação:

- **Graduação: Sistemas de Informação;**
- **Especialista em: Engenharia e Arquitetura de Software;**
- **MBA EXECUTIVO EM BUSINESS INTELLIGENCE (em andamento);**
- **Mestrado Acadêmico em Engenharia de Computação (UPE em andamento);**

Maiores informações :

Linkedin

UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Vamos analisar o seguinte cenário

Um banco, é bem fácil perceber que uma entidade extremamente importante para o nosso sistema é a conta e o que toda conta tem e é importante para nós é?

- número da conta
- nome do titular da conta
- saldo

Programação Orientada a Objetos

Vamos analisar o seguinte cenário

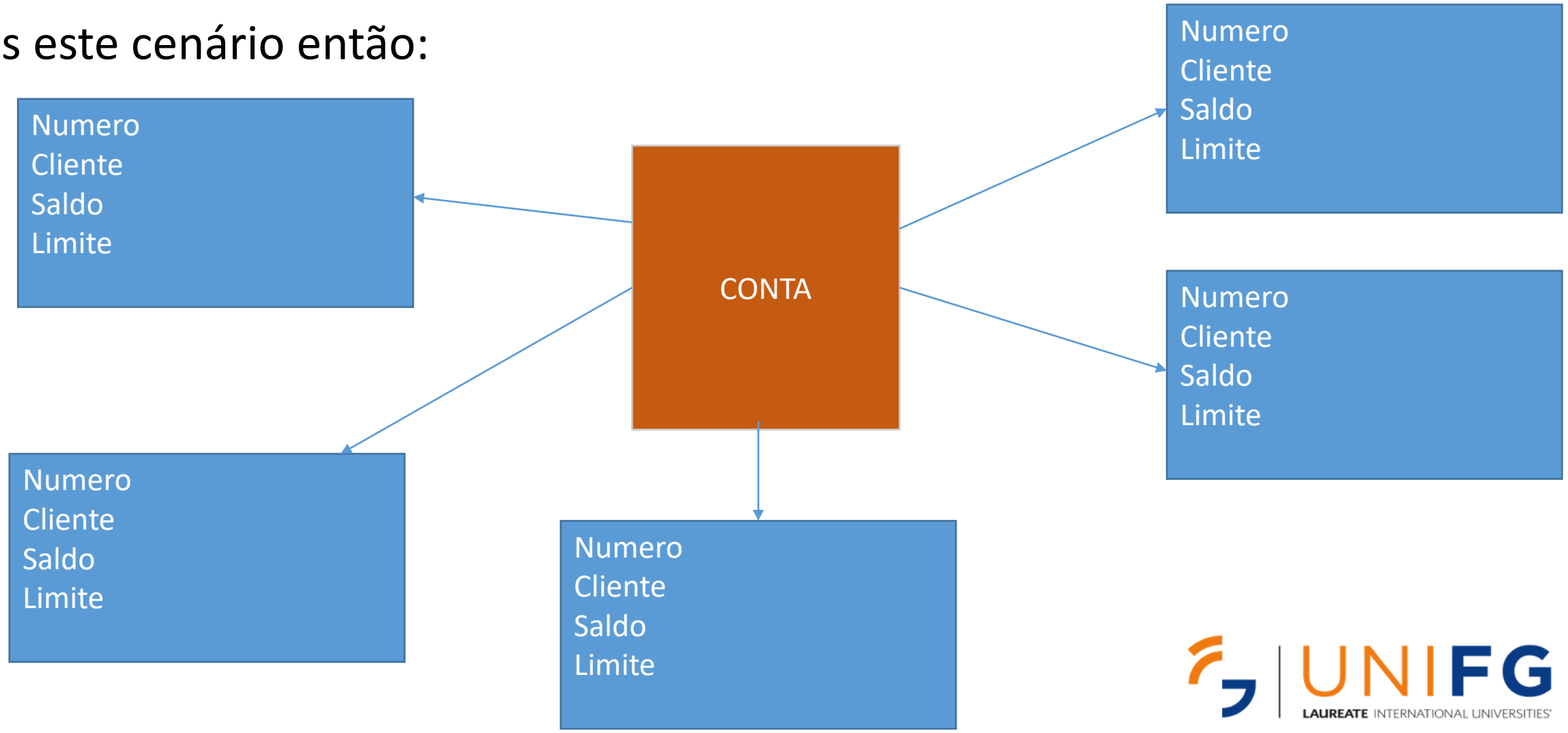
O que gostaríamos de "pedir à conta“ quando nós solicitamos em um sistema de banco?

- Saca uma quantidade X
- Deposita uma quantidade X
- Imprime o nome do titular da conta
- Ver o saldo atual
- Transfere um dinheiro de X para uma outra conta Y
- Ver o tipo de conta

UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Temos este cenário então:



UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Vamos analisar o seguinte cenário

O que gostaríamos de "pedir à conta“ quando nós solicitamos em um sistema de banco?

- Saca uma quantidade X
- Deposita uma quantidade X
- Imprime o nome do titular da conta
- Ver o saldo atual
- Transfere um dinheiro de X para uma outra conta Y
- Ver o tipo de conta

Programação Orientada a Objetos

Vamos analisar o seguinte cenário

O que gostaríamos de "pedir à conta“ quando nós solicitamos em um sistema de banco?

- Saca uma quantidade X : Queremos criar um método que saca uma determinada quantidade;

Programação Orientada a Objetos

Vamos analisar o seguinte cenário

O que gostaríamos de "pedir à conta“ quando nós solicitamos em um sistema de banco?

- Deposita uma quantidade X : Temos o método para depositar alguma quantia, para poder manipular a conta.

Programação Orientada a Objetos

OBJETOS SÃO ACESSADOS POR REFERÊNCIAS

Quando declaramos uma variável para associar a um objeto, na verdade, essa variável não guarda o objeto, e sim uma maneira de acessá-lo, chamada de **referência**.

```
public static void main(String[] args) {

        Conta conta1;
        conta1 = new Conta();
        Conta conta2;
        conta2 = new Conta();

}
```

Programação Orientada a Objetos

OBJETOS SÃO ACESSADOS POR REFERÊNCIAS

O que temos então é “Tenho uma referência **conta1** a um objeto do tipo **Conta**”.

```
public static void main(String[] args) {

        Conta conta1;
        conta1 = new Conta();
        Conta conta2;
        conta2 = new Conta();

}
```

Programação Orientada a Objetos

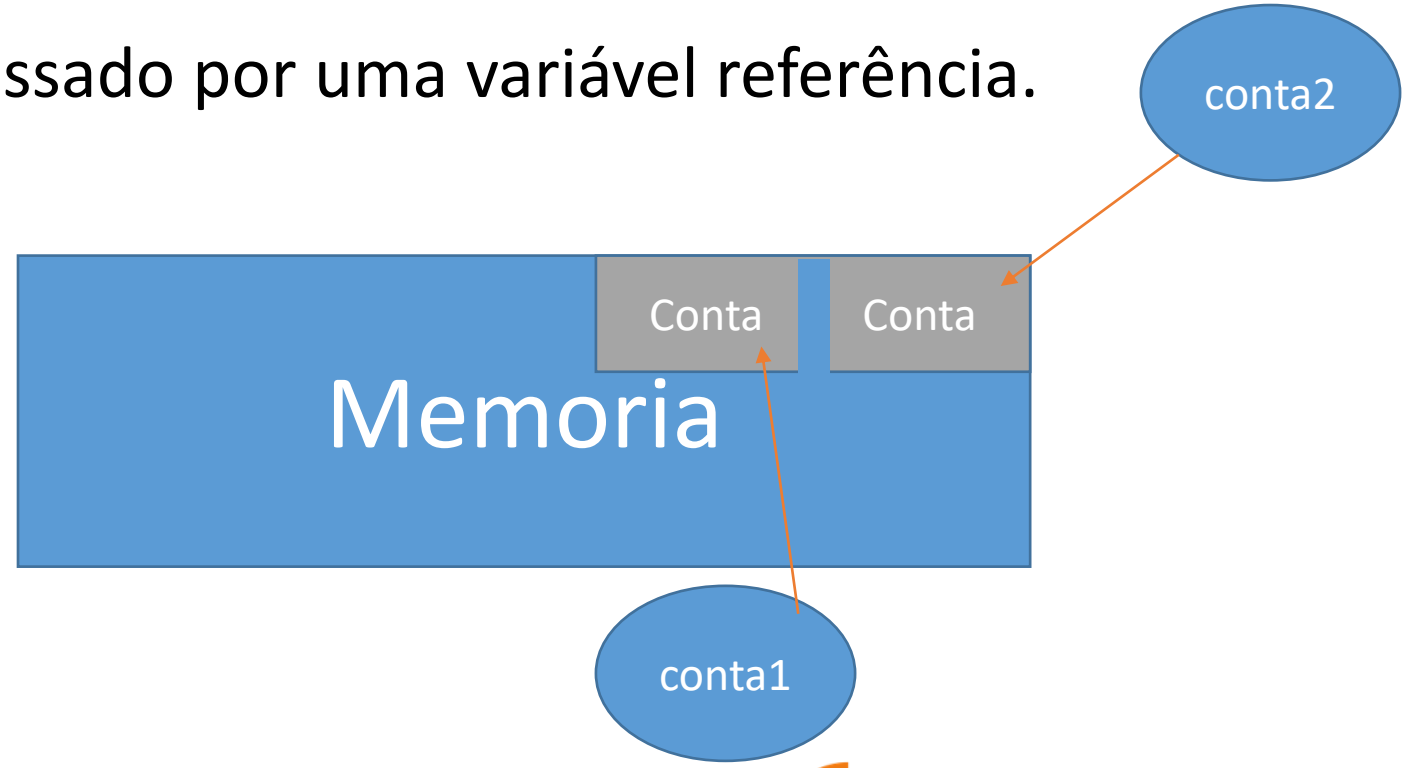
OBJETOS SÃO ACESSADOS POR REFERÊNCIAS

Ou seja todo objeto Java é acessado por uma variável referência.

```
public static void main(String[] args) {

        Conta conta1;
        conta1 = new Conta();
        Conta conta2;
        conta2 = new Conta();

}
```



UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

OBJETOS SÃO ACESSADOS POR REFERÊNCIAS

Ou seja todo objeto Java é acessado por uma variável referência.

```
public static void main(String[] args) {

Conta conta1 = new Conta();
        conta1.deposita(100);
        Conta conta2 = conta1;
        conta2.deposita(200);
System.out.println(conta1.saldo);
System.out.println(conta2.saldo);
}
```

UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

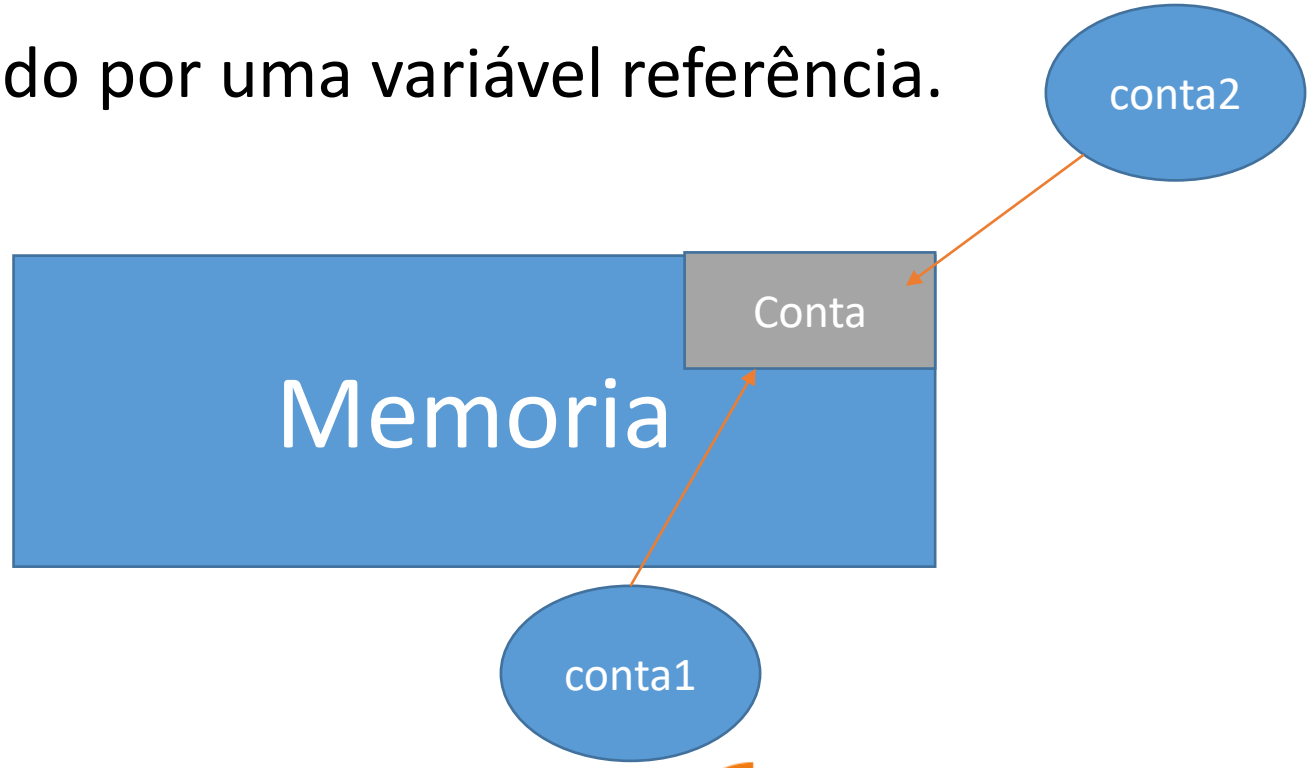
Programação Orientada a Objetos

OBJETOS SÃO ACESSADOS POR REFERÊNCIAS

Ou seja todo objeto Java é acessado por uma variável referência.

```
public static void main(String[] args) {

Conta conta1 = new Conta();
        conta1.deposita(100);
        Conta conta2 = conta1;
        conta2.deposita(200);
System.out.println(conta1.saldo);
System.out.println(conta2.saldo);
}
```



UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Vamos analisar o seguinte cenário

O que gostaríamos de "pedir à conta“ quando nós solicitamos em um sistema de banco?

- Transfere um dinheiro de X para uma outra conta Y : poderíamos verificar se a conta possui a quantidade a ser transferida disponível.

Programação Orientada a Objetos

Analisando nossa classe Conta

Vamos aumentar nossa classe Conta e adicionar **nome**, **sobrenome** e **cpf** do titular da conta, então vamos criar a nossa classe ***cliente*** que será uma composição da nossa classe ***conta***.

```
public static void main(String[] args) {
        Conta minhaConta = new Conta();
        Cliente conta = new Cliente();
        minhaConta.titular = conta;
}
```

UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Analisando nossa classe Conta

Podemos realmente navegar sobre toda essa estrutura de informação, sempre usando o “**ponto**”:

```
public static void main(String[] args) {
        Conta minhaConta = new Conta();
        minhaConta.titular.nome = "Diogenes";
}
```

Programação Orientada a Objetos

Exercício para ser feliz ☺

1-Modele uma conta. A ideia aqui é apenas modelar, isto é, só identifique que informações são importantes. Desenhe no papel tudo o que uma Conta tem e tudo o que ela faz. Ela deve ter o nome do titular (String), o número (int), a agência (String), o saldo (double) e uma data de abertura (String).

Além disso, ela deve fazer as seguintes ações: saca, para retirar um valor do saldo; deposita, para adicionar um valor ao saldo; calculaRendimento, para devolver o rendimento mensal dessa conta.

Programação Orientada a Objetos

Exercício para ser feliz ☺

2-Transforme o nosso modelo em uma classe Java. Teste-a, usando uma outra classe que tenha o main . Você deve criar a classe da conta com o nome Conta , mas pode nomear como quiser a classe de teste, contudo, ela deve possuir o método main . A classe Conta deve conter pelo menos os seguintes métodos:

- Saca que recebe um valor como parâmetro e retira esse valor do saldo da conta
- Deposita que recebe um valor como parâmetro e adiciona esse valor ao saldo da conta
- calculaRendimento que não recebe parâmetro algum e devolve o valor do saldo multiplicado por 0.1

UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Exercício para ser feliz ☺

2-Transforme o nosso modelo em uma classe Java. Teste-a, usando uma outra classe que tenha o main . Você deve criar a classe da conta com o nome Conta , mas pode nomear como quiser a classe de teste, contudo, ela deve possuir o método main . A classe Conta deve conter pelo menos os seguintes métodos:

- Saca que recebe um valor como parâmetro e retira esse valor do saldo da conta
- Deposita que recebe um valor como parâmetro e adiciona esse valor ao saldo da conta
- calculaRendimento que não recebe parâmetro algum e devolve o valor do saldo multiplicado por 0.1

UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES