



C125 – Programação Orientada a Objetos com
Java



INTRODUÇÃO A ORIENTAÇÃO A OBJETOS – PT1

Prof. Phyllipe Lima
phyllipe@inatel.br

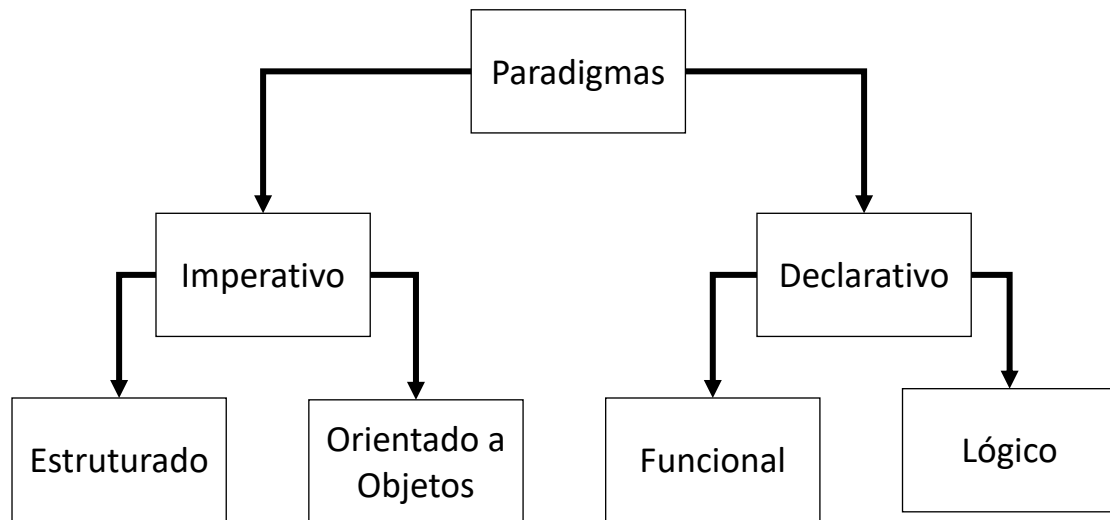
1

Agenda



- ☕ Comparativo entre programação estruturada e orientada a objetos (OO);
- ☕ Classes e Objetos;
- ☕ Uma classe em Java;
- ☕ Criando e usando um Objeto;
- ☕ Métodos;
- ☕ UML e Diagrama de Classes;
- ☕ Objetos são acessados por referências
- ☕ Classes dentro de Classes;

2



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

3

3

Imperativo



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

4

4



- ☕ Computação vista como um processo que realiza mudança de estado.
- ☕ Específica como um processamento deve ser feito
- ☕ Variáveis podem possuir diferentes valores a cada momento (mudança de estado)

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

5

5



```
int somar(int x, int y){  
    int z;  
    z = x + y;  
    return z;  
}
```

- ☕ Faça a soma de X com Y
- ☕ Me devolva o resultado

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

6

6



Estruturado



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

7

7



☕ Se baseia na construção de blocos aninhados de comandos.

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

8

8

 Utiliza três mecanismos básicos: Sequencia Seleção Iteração Exemplo de linguagem: C e Pascal Sequencia`int x;``int y;``int z;``x = x * 10;``y = x + z;``trocar(x,y);`

Seleção

```
if (x > 10){  
    printf("x maior que 10");  
}else{  
    printf("x não é maior que 10");  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

11

11

Iteração

```
for (int i = 0; i < 10; i++){  
    printf("Imprimindo o número: %d", i);  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

12

12

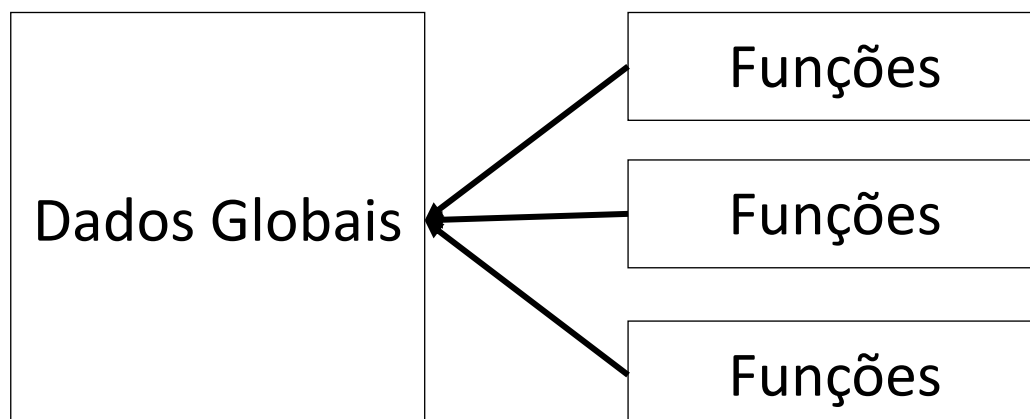
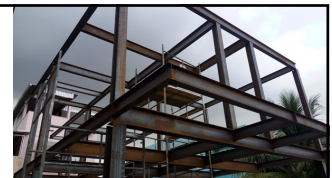
☕ E os dados?



☕ Baseado em variáveis globais

☕ Podem ser acessados de qualquer parte do código sem a necessidade de permissão

☕ E os dados?





Orientado a Objetos



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

15

15

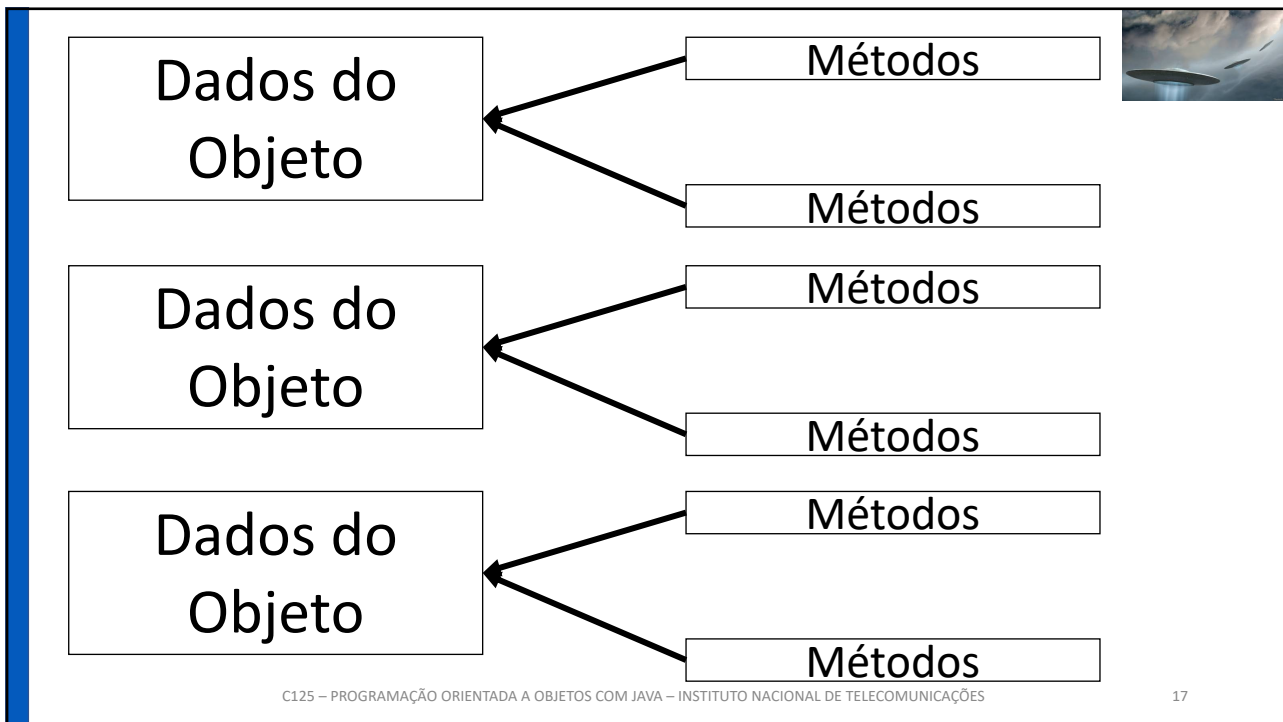


- ☕ Visto como uma evolução do paradigma estruturado;
- ☕ Apresenta melhor organização dos dados
- ☕ Exemplo: Java, C#, C++

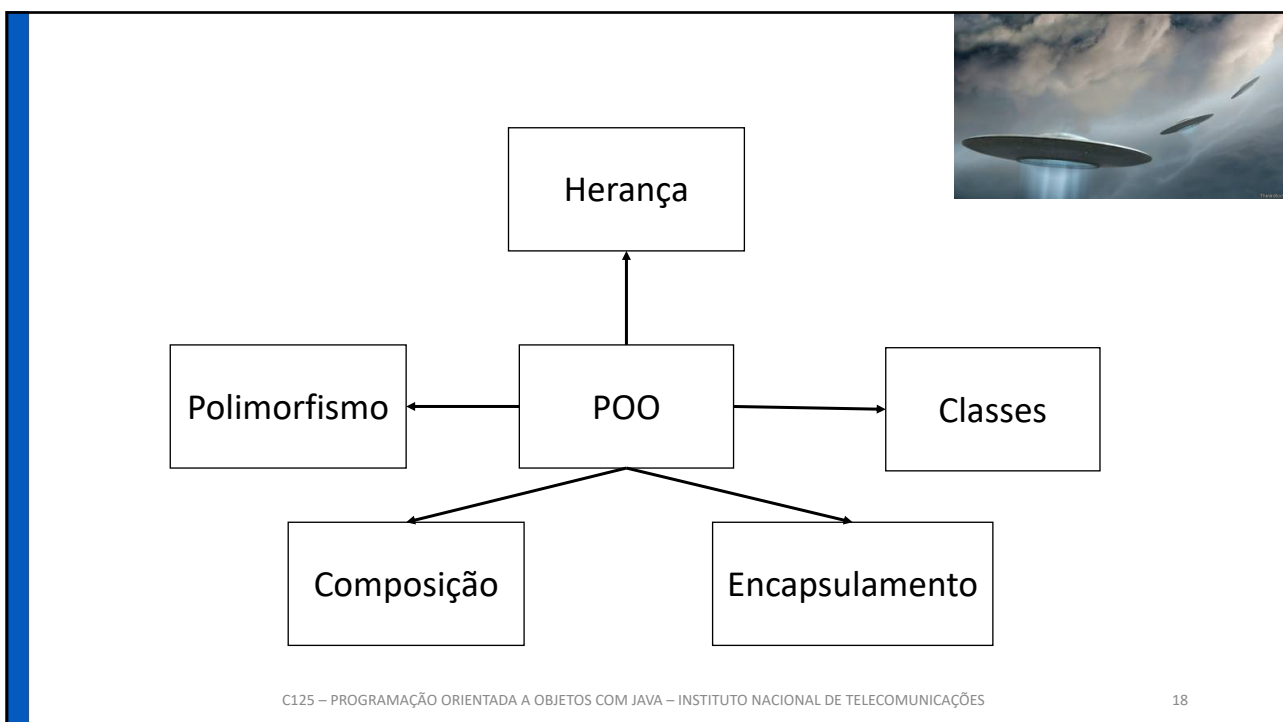
C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

16

16



17



18

Exemplo de Classe em Java

```
public class Pessoa{  
    String nome;  
    int idade;  
    public void falar(){  
        System.out.println("Olá alunos de C125");  
    }  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

19

19

Exemplo de Herança em Java

```
public class Aluno extends Pessoa{  
    double nota;  
    public void fazerProva(){  
        //fez a prova e foi aprovado 😊  
    }  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

20

20

☕ Exemplo de Composição em Java



```
public class Nave{  
    double vida;  
    Arma arma; //Nave é composta por Arma  
    public void atirar(){  
        arma.atirar(); //Quem atira é a Arma  
    }  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

21

21

☕ Exemplo de Encapsulamento



```
public class Nave{  
    private double vida;  
    public void alteraVida(){  
        //faz as verificações necessárias  
        //antes de alterar a vida  
    }  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

22

22



☕ Por que os cursos de graduação se iniciam com programação estruturada?

☕ Por que é muito mais simples iniciar um código estruturado! 😊



Classes e Objetos



- ☕ Vamos criar uma programa para um banco
- ☕ Uma abstração que precisaremos é uma CONTA!

- ☕ Precisamos responder
 - ☕ O que uma CONTA possui?
 - ☕ O que uma CONTA faz?



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

25

25

- ☕ O que uma CONTA possui? (Membros da classe)
 - ☕ Saldo;
 - ☕ Limite;
 - ☕ Número da Conta;
 - ☕ Dono da Conta;
- ☕ O que uma CONTA sabe fazer? (Métodos)
 - ☕ Sacar;
 - ☕ Depositar;
 - ☕ Transferir
 - ☕ Verificar saldo;

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

26

26

☕ Essa conta é apenas uma especificação!

☕ Ainda não podemos utilizá-la

☕ Para utilizar uma conta, precisamos construí-la e obter uma INSTÂNCIA da classe!

☕ A classe é uma especificação

☕ A instância é o objeto concreto



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

27

27

número:
cliente:
saldo:
limite:

número: 1234
cliente: Maria
saldo: 1000,00
limite: 2500,00

número: 4321
cliente: João
saldo: 700,00
limite: 1300,00

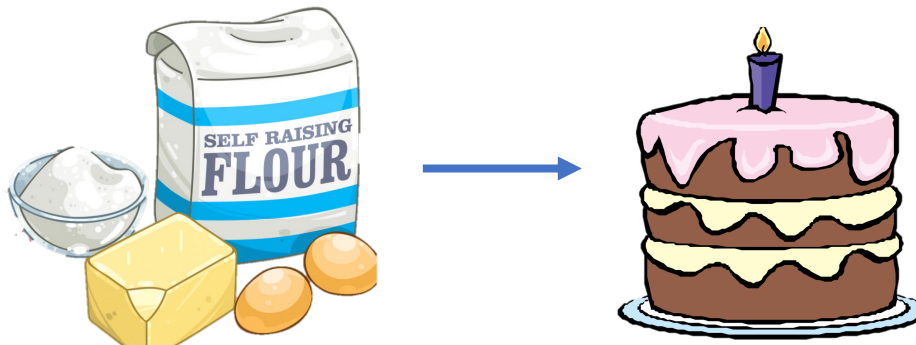


C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

28

28

- ☕ Na vida não comemos a receita!
- ☕ Precisamos “construir” o bolo primeiro



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

29

29

- ☕ Vamos criar a nossa classe Conta



```
public class Conta {
    //Membros da Classe
    int numero;
    String nomeDoDono;
    float saldo;
    float limite;

    //Metodos
}
```

- ☕ Variáveis declaradas no escopo da classe chamamos de “Membros da classe” (ou “atributos”, mas evite usar esse termo)

- ☕ Campos da classe (*field*)

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

30

30

☕ Ok! Temos nossa especificação de Conta, mas como usá-la?

☕ Precisamos de uma instância de Conta

```
public class Main {
    public static void main(String[] args) {
        //Criando uma nova instancia de Conta!
        new Conta();
    }
}
```

☕ Mas onde ela está sendo salva?

☕ Para o além!

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

31

31

☕ Precisamos atribuir a instância de Conta para alguma variável!

☕ Ela precisa ser capaz de salvar uma dado tipo "Conta".

```
public class Main {
    public static void main(String[] args) {
        //Criando uma nova instancia de Conta!
        //E atribuindo a uma variável do tipo Conta!
        Conta c = new Conta();
    }
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

32

32

☕ Atribuindo valores e acessando esses valores!



```
public class Main {  
  
    public static void main(String[] args) {  
  
        //Criando uma nova instancia de Conta!  
        //E atribuindo a uma variável do tipo Conta!  
        Conta c = new Conta();  
  
        c.nomeDoDono = "Malfurion";  
        c.saldo = 2000;  
  
        System.out.println("O dono da classe é: "  
                            + c.nomeDoDono + " e o saldo é: "  
                            + c.saldo);  
    }  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

33

33

☕ E as ações (comportamento) da classe Conta?

☕ Vamos criar métodos para isso.

☕ Como podemos sacar e depositar nessa conta?

☕ Fazemos isso na classe Conta!



```
//Metodo para depositar  
void deposita(float quantia) {  
    saldo += quantia;  
}  
//Metodo para sacar  
void saca(float quantia) {  
    float novoSaldo = saldo - quantia;  
    saldo = novoSaldo;  
}
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

34

34

☕ Invocando o método!

☕ O operador “.” tem essa função!



```
//Depositando R$ 100,00  
c.deposita(100);  
System.out.println(c.saldo);  
  
//Sacando R$ 50,00  
c.saca(50);  
System.out.println(c.saldo);
```

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

35

35

Exercício 1

☕ Exercício 1 – Zombicide!



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

36

36

Exercício 1

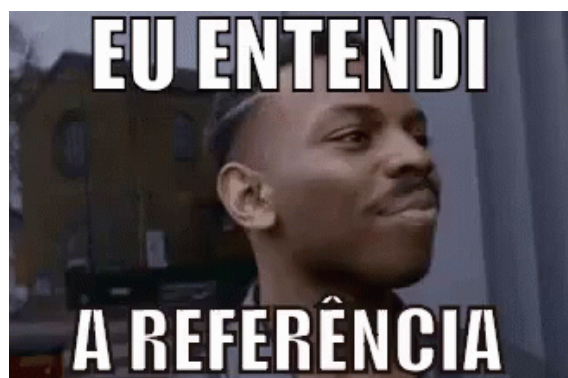


Exercício 1 – Zombicide!

Você e seus amigos estão querendo criar um software para modelar o jogo zombicide! Crie uma classe que representa um zumbi. O que um zumbi sabe sobre si? E o que ele sabe fazer?

Depois crie uma classe Principal que use o zumbi e invoque suas ações

Objetos são acessados por referência!



Objetos são acessados por referencia!

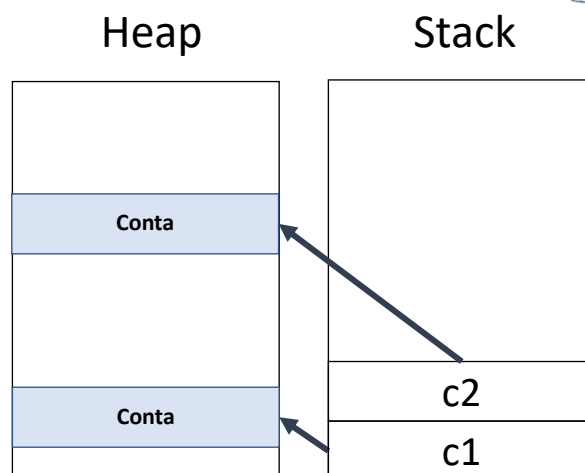
☕ Quando declaramos uma variável para guardar um objeto, estamos na verdade guardando uma REFERÊNCIA para esse objeto.

☕ Exemplo:

- ☕ `Conta c = new Conta();`
- ☕ Não é correto dizer que “c” é um objeto.
- ☕ Dizemos que “c” é uma variável que guarda uma REFERÊNCIA para uma INSTÂNCIA de Conta

Objetos são acessados por referencia!

```
Conta c1 = new Conta();
Conta c2 = new Conta();
```



Objetos são acessados por referencia!



- ☕ As variáveis `c1` e `c2` são “ponteiros” para o objeto `Conta`
- ☕ Mas não podemos fazer a aritmética de ponteiros com essas variáveis!!!!!!

```
Conta c1 = new Conta();
```

```
Conta c2 = new Conta();
```

Objetos são acessados por referencia!



- ☕ Podemos comparar variáveis de referencias?
 - ☕ Sim, mas fique atento!
 - ☕ Lembre-se que são referências!!!!



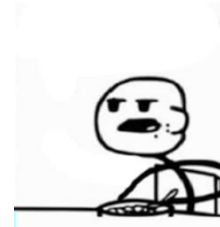
Objetos são acessados por referencia!



```
Conta c1 = new Conta();
c1.nomeDoDono = "Jaum";
c1.limite = 1000;
c1.numero = 1234;
c1.saldo = 2000;
```

```
Conta c2 = new Conta();
c2.nomeDoDono = "Jaum";
c2.limite = 1000;
c2.numero = 1234;
c2.saldo = 2000;
```

```
if(c1 == c2)
    System.out.println("São iguais!");
else
    System.out.println("Não são iguais!");
```



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

43

43

Objetos são acessados por referencia!



```
Conta c1 = new Conta();
c1.nomeDoDono = "Jaum";
c1.limite = 1000;
c1.numero = 1234;
c1.saldo = 2000;
```

```
Conta c2 = new Conta();
c2.nomeDoDono = "Jaum";
c2.limite = 1000;
c2.numero = 1234;
c2.saldo = 2000;
```

Não são iguais!

```
if(c1 == c2)
    System.out.println("São iguais!");
else
    System.out.println("Não são iguais!");
```



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

44

44

Objetos são acessados por referencia!



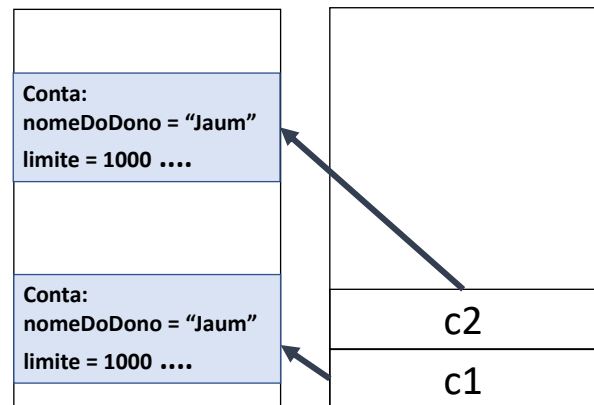
```
Conta c1 = new Conta();
c1.nomeDoDono = "Jaum";
c1.limite = 1000;
c1.numero = 1234;
c1.saldo = 2000;
```

```
Conta c2 = new Conta();
c2.nomeDoDono = "Jaum";
c2.limite = 1000;
c2.numero = 1234;
c2.saldo = 2000;
```

```
if(c1 == c2)
    System.out.println("São iguais!");
else
    System.out.println("Não são iguais!");
```

Heap

Stack



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

45

45

Objetos são acessados por referencia!



```
Conta c1 = new Conta();
c1.nomeDoDono = "Jaum";
c1.limite = 1000;
c1.numero = 1234;
c1.saldo = 2000;
```

```
Conta c2 = new Conta();
c2.nomeDoDono = "Jaum";
c2.limite = 1000;
c2.numero = 1234;
c2.saldo = 2000;
```

```
c1 = c2;
```

```
if(c1 == c2)
    System.out.println("São iguais!");
else
    System.out.println("Não são iguais!");
```

São iguais!



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

46

46

Objetos são acessados por referencia!

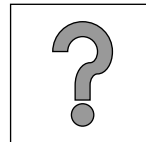


Mas e o objeto que era referenciado por c1?

```
Conta c1 = new Conta();
```

```
Conta c2 = new Conta();
```

```
c1 = c2;
```



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

47

47

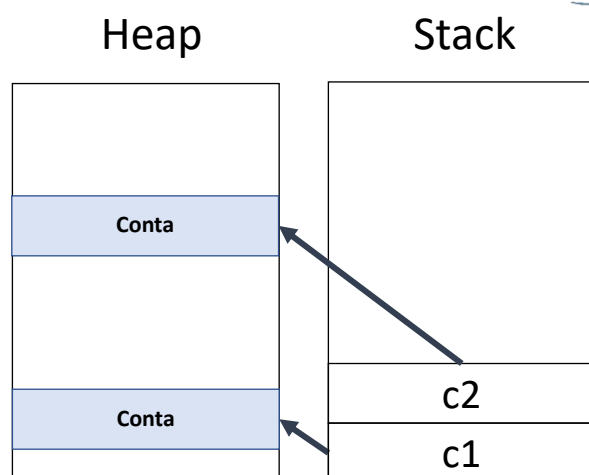
Objetos são acessados por referencia!



```
Conta c1 = new Conta();
```

```
Conta c2 = new Conta();
```

```
c1 = c2;
```



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

48

48

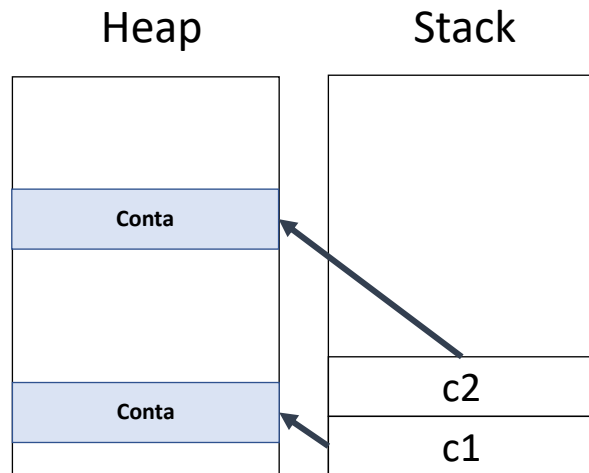
Objetos são acessados por referencia!



```
Conta c1 = new Conta();
```

```
Conta c2 = new Conta();
```

```
c1 = c2;
```



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

49

49

Objetos são acessados por referencia!



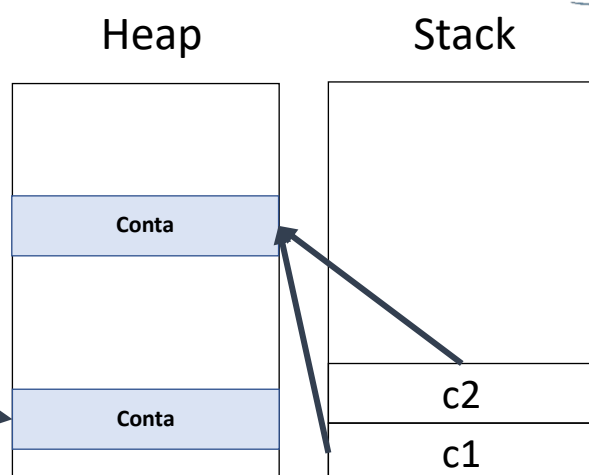
```
Conta c1 = new Conta();
```

```
Conta c2 = new Conta();
```

```
c1 = c2;
```



Garbage Collector



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

50

50



Vamos transferir dinheiro!!!!

☕ Na classe Conta, crie o método “transferir”.

☕ Ele recebe como parâmetro, outra Conta!

```
void transferir(Conta contaDestino, float quantia) {  
    saldo -= quantia;  
    contaDestino.saldo += quantia;  
}
```



Vamos transferir dinheiro!!!!

☕ Vamos testar no método “main”

```
c1.transferir(c2, 100);  
System.out.println("O saldo de c1 é: " + c1.saldo);  
System.out.println("O saldo de c2 é: " + c2.saldo);
```

O saldo de c1 é: 1900.0

O saldo de c2 é: 2100.0

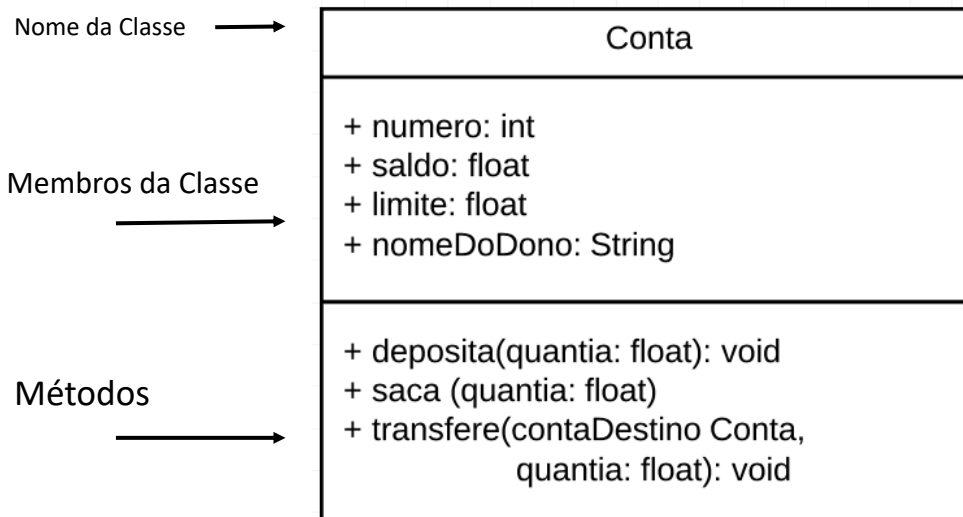
☕ Lembre-se: Objetos são passados por referência!

UML (Unified Modeling Language)!



- ☕ É uma linguagem visual para especificação, visualização e documentação de software!
- ☕ O DIAGRAMA DE CLASSES, é um dos vários tipos de diagramas oferecidos pela linguagem UML
- ☕ Considere o diagrama UML da classe Conta

UML (Unified Modeling Language)!





Exercício 2

Exercício 2 – Zombicide!



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

55

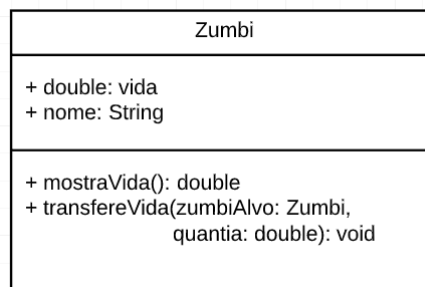
55



Exercício 2

Exercício 2 – Zombicide!

Você e seus amigos **AINDA** estão criando um software para modelar o jogo zombicide! Considere o UML para modelar o zumbi!



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

56

56

Exercício 3



Exercício 3 – Zombicide!

Você e seus amigos **CONTINUAM** criando um software para modelar o jogo zombicide! Considerando o zumbi do exercício 2

- Crie dois zumbis (cada um com seu “new”).
- Manipule a vida individualmente
- Depois faça ambas as variáveis de referencias iguais
 - Ex: `z1 = z2;`
- Manipule a vida dos zumbis (através de `z1` e `z2`).
- Verifique a vida deles



Exercício 4



Exercício 4 – Zombicide!

Sim isso mesmo, ainda estão no zumbis! Modifique o Zumbi do Exercício 2 para atender a seguinte especificação

- A ação de transferir vida faz uma verificação antes se de fato é possível ser realizada.
- Ele retorna um booleano informando se deu certo a transferência
- Modifique o Diagrama para atender o novo requisito



Resolução dos Exercícios



<https://github.com/phillima-inatel/C125>



C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES



59

59



Material Complementar



 **Capítulo 4 da apostila FJ-11**
 **Orientação a Objetos Básica**

C125 – PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA – INSTITUTO NACIONAL DE TELECOMUNICAÇÕES

60

60

Inatel

C125 – Programação Orientada a Objetos com
Java

INTRODUÇÃO A ORIENTAÇÃO A OBJETOS – PT1

Prof. Phyllipe Lima
phyllipe@inatel.br

