

# Aula 07

## Programação Orientada a Objetos

emerson@paduan.pro.br

## Associação entre classes



emerson@paduan.pro.br

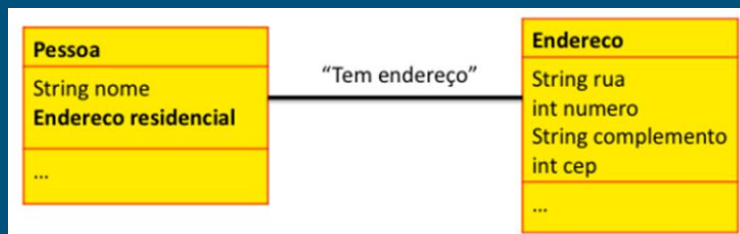
# Associação

Um sistema é composto por várias Classes.

- As classes se conectam para poderem se comunicar por troca de mensagens (chamadas de métodos)
- Quando um ou mais atributos de uma Classe é uma referência para outra Classe temos uma associação.

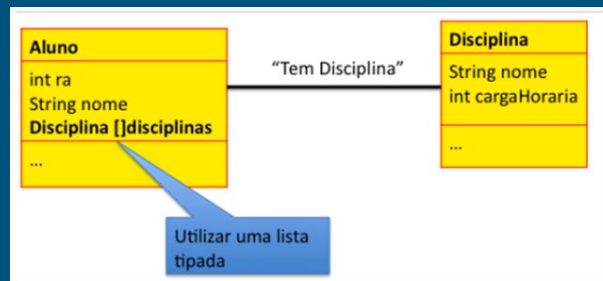
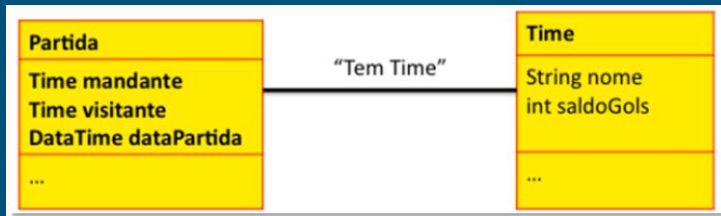
emerson@paduan.pro.br

## Exemplo



emerson@paduan.pro.br

## Exemplo 1: N



emerson@paduan.pro.br

## Em Java

```
public class Pessoa{
    //atributos
    private String nome;
    private int idade;
    private char sexo;
    private Endereco end;
    ....

    public String imprimir(){
        return "Nome: " + nome +
            "\nIdade: " + idade +
            "\nSexo: " + sexo +
            "Endereço: " + end.imprimir();
    }
}
```

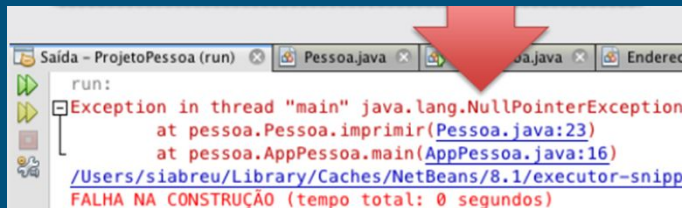
```
public class Endereco{
    //atributos
    private String logradouro;
    private String complemento;
    private int numero;
    private String cep;
    ....

    public String imprimir(){
        return "Logradouro: " + logradouro +
            "\nComplemento: " + complemento +
            "\nNúmero: " + numero +
            "CEP: " + cep;
    }
}
```

emerson@paduan.pro.br

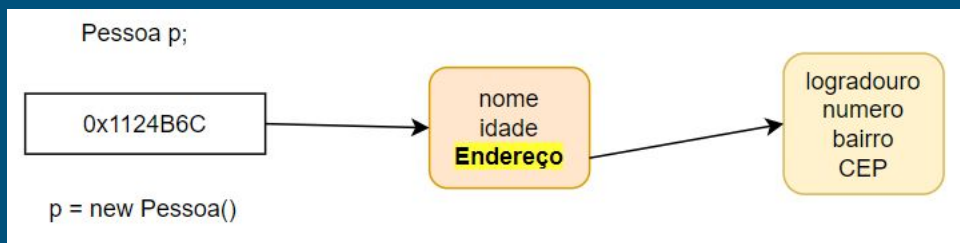
# Atenção!!!

```
public class AppPessoa {  
    public static void main(String[] args) {  
        Pessoa objPessoa = new Pessoa();  
  
        System.out.println(objPessoa.imprimir());  
    }  
}
```



emerson@paduan.pro.br

## Criar objeto no construtor



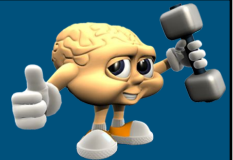
emerson@paduan.pro.br

# Resolvendo

```
public class Pessoa{  
    //atributos  
    private String nome;  
    private int idade;  
    private char sexo;  
    private Endereco end;  
  
    //construtor default  
    public Pessoa(){  
        this.end = new Endereco();  
    }  
  
    public String imprimir(){  
        return "Nome: " + nome +  
            "\nIdade: " + idade +  
            "\nSexo: " + sexo +  
            "Endereço: " + end.imprimir();  
    }  
}
```

emerson@paduan.pro.br

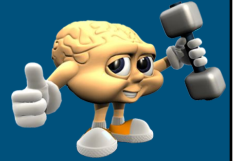
## Exercício 7-1



1. Criar um relacionamento de Associação entre as classes Animal e Proprietário:  
*Animal "Tem UM" Proprietario*
2. Criar a classe *Animal* com os seguintes atributos: nome, raça, cor, ano de nascimento e proprietário
3. Criar a Classe *Proprietario* com os seguintes atributos: nome e telefone
4. Criar uma classe main

emerson@paduan.pro.br

## Exercício 7-2



1. Criar uma classe Conta, que possua um número, e um saldo, e os métodos para obter dados da conta, realizar depósito e saque.
2. Crie as subclasses da classe Conta: ContaCorrente, ContaEspecial e ContaPoupanca.  
A ContaCorrente permite fazer saques somente se houver saldo suficiente. A ContaEspecial possui um limite que permite fazer saques se o saldo mais o limite da conta cobrir o valor pretendido de saque. A ContaPoupança faz saque se houver saldo, mas há uma taxa por operação. Além disso, a ContaCorrente deve reescrever o método deposita, com o objetivo de retirar uma taxa bancária de dez centavos de cada depósito.
3. Crie uma classe AppContas com o método main contendo um menu com opções para realizar operações nas contas.