

Aula 07

Programação Orientada a Objetos

emerson@paduan.pro.br

Upcast

É uma conversão na qual subclasses são promovidas a superclasses.

Característica: A conversão é implícita! A promoção é realizada automaticamente!

Exemplo: Gerente "é um" Funcionario

```
Funcionario func = new Gerente(); //upcast
```

emerson@paduan.pro.br

Downcast

É a operação inversa, superclasses são convertidas em subclasses.
Característica: A conversão é explícita! Tem que indicar o TIPO!

Exemplo:

```
Funcionario func = new Gerente(); //upcast  
  
Gerente ger = (Gerente) func; //downcast
```

emerson@paduan.pro.br

Associação entre classes



emerson@paduan.pro.br

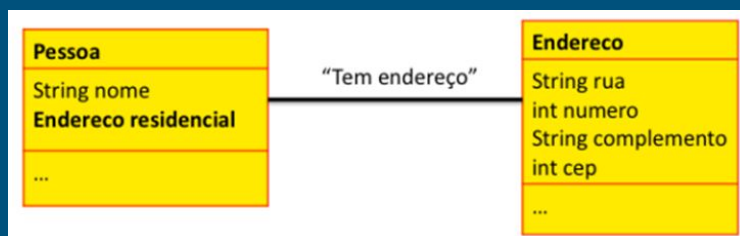
Associação

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.

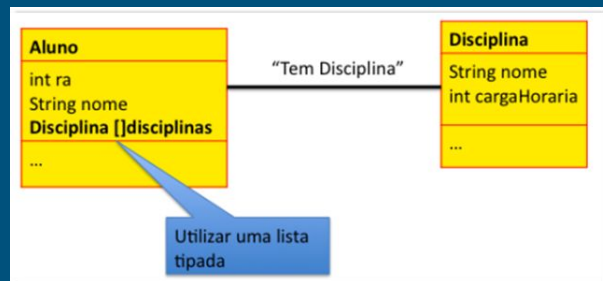
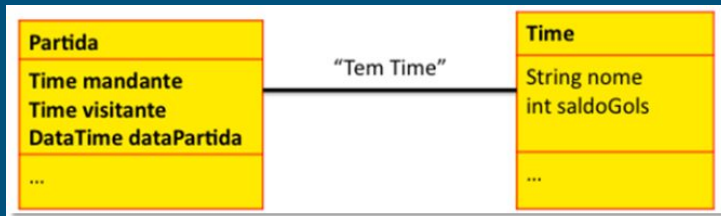
emerson@paduan.pro.br

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();
    }
}
```

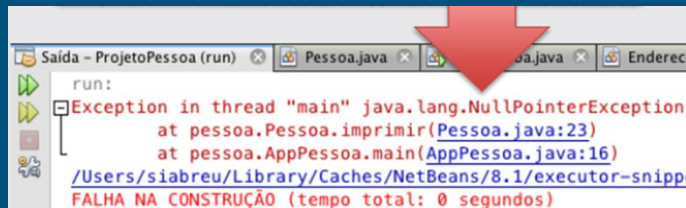
```
public class Endereco{
    //atributos
    private String logradouro;
    private String complemento;
    private int numero;
    private String cep;
    ....

    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());  
    }  
}
```



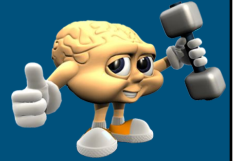
emerson@paduan.pro.br

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" *Proprietário*
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 para *Gerenciar* objetos da classe *Animal*
5. Criar a classe *AppAnimalProprietario* para:
 - Cadastrar *Animal* e *Proprietário*
 - Listar todos os animais cadastrados
 - Dada uma raça, listar o nome de todos os proprietários que tenham animal dessa raça