



LINGUAGEM DE PROGRAMAÇÃO I

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNESP

CONSTRUTORES

Um construtor é um método utilizado por uma classe para criar objetos dela.

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

```
        Carro c = new Carro ();
```

```
    }  
}
```

Como era antes?

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

```
        Carro c = new Carro ("Chevrolet", "Ônix", 2022);
```

```
    }  
}
```

Como podemos fazer agora?

```
public class Principal{  
    public static void main(String[] args) {  
  
        Carro c = new Carro ();  
  
    }  
}
```

Se um construtor não for definido pelo desenvolvedor, por padrão, ele é automaticamente incluído quando a classe é compilada.

```
public class Carro {  
    private String marca;  
    private String modelo;  
    private int ano;
```

```
    public Carro() {  
  
    }  
}
```

```
    // Métodos getters e setters  
}
```

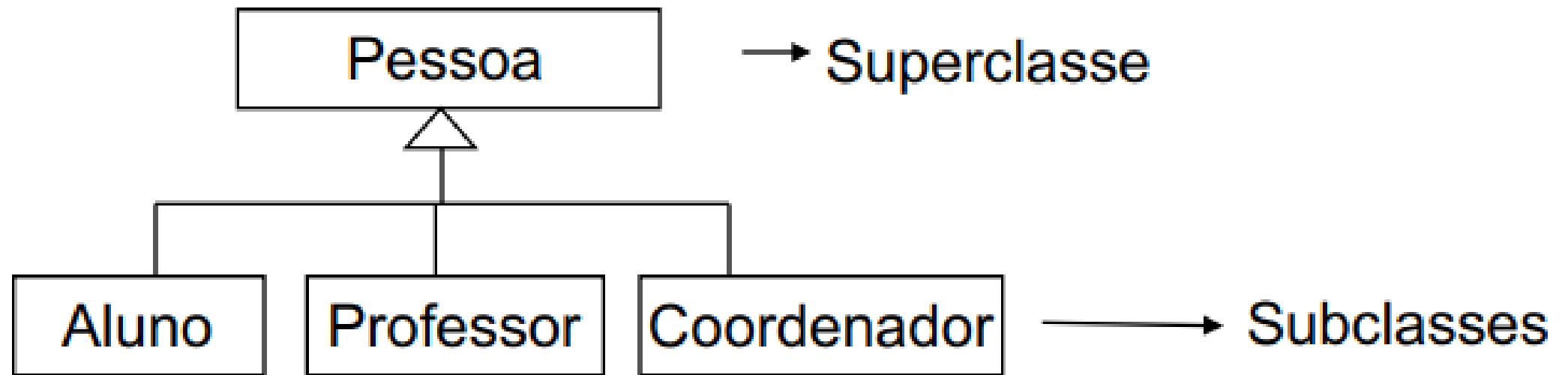
```
public class Principal{  
    public static void main(String[] args) {  
  
        Carro c = new Carro ("Chevrolet", "Ônix", 2022);  
        //Predefinindo valores  
    }  
}
```

```
public class Carro {  
    private String marca;  
    private String modelo;  
    private int ano;
```

```
    public Carro(String marca, String modelo, int ano) {  
        this.marca = marca;  
        this.modelo = modelo;  
        this.ano = ano;  
    }  
    //construtor
```

```
    // Métodos getters e setters  
}
```

OBSERVAÇÃO!



Exemplo 1

```
public class Pessoa {  
    protected String nome;
```

```
    public Pessoa(String nome) {  
        this.nome = nome;  
    }
```

```
}
```

```
public class Aluno extends Pessoa {  
    private int matricula;
```

```
    public Aluno(String nome, int mat) {  
        super(nome);  
        this.matricula = mat;  
    }
```

Exemplo 2

```
public class Pessoa{  
    protected String nome;
```

```
    public Pessoa(String nome){  
        super(); ←  
        this.nome=nome;  
    }
```

```
    // demais métodos devem vir aqui  
}
```

```
public class Aluno extends Pessoa{  
    private int mat;
```

```
    public Aluno(String nome, int mat){  
        super(nome);  
        this.mat=mat;  
    }
```

```
    // demais métodos devem vir aqui  
}
```


E a Main?

Exemplo 1

```
public class Main {  
    public static void main(String[] args) {  
  
        Aluno aluno = new Aluno("João", 12345);  
  
        System.out.println("Nome: " + a.getNome());  
        System.out.println("Matricula: " + a.getIdade());  
  
    }  
}
```

Exemplo 2

```
//import
public class Main {
    public static void main(String[] args) {
        // Scanner
        Aluno aluno = new Aluno("João", 12345);

        System.out.println("Digite o nome do aluno: ");
        String nomeAluno = sc.nextLine();

        System.out.println("Digite a matrícula: ");
        int matriculaAluno = sc.nextInt();

        Aluno aluno = new Aluno(nomeAluno, matriculaAluno);
    }
}
```

DÚVIDAS?

Exercício

1) Implemente um programa em java que possua um professor com nome, idade e salário. Além disso, o programa deve ter implementado um aluno com nome, idade e matrícula . As informações devem ser feitas de forma fixa na instanciação do objeto e depois mostre todas informações na tela

2) Elabore um programa que resolva o seguinte problema:

```
Produto produto1 = new Produto("Camisa", 49.99, "Camisa polo azul");  
Produto produto2 = new Produto("Calça", 99.99, "Calça jeans preta");
```

Nota de Aula

Nota de Aula

Elabore um código com as classes que completem a classe principal:

```
public class Main {  
    public static void main(String[] args) {  
  
        Carro carro = new Carro("Toyota", "Corolla");  
        carro.mostrarDetalhes();  
  
    }  
}
```