



SENAI

SÃO PAULO

Prof. Vedilson Prado



AULA 04

PILARES POO - HERANÇA E POLIMORFISMO



<p> Prof. Veditson </p>

Programação Orientada a Objetos

Polimorfismo

Herança

Abstração

Encapsulamento





Abstração

Encapsulamento



Herança



Polimorfismo

HERANÇA

Conceito: A herança é o pilar da POO que permite que uma classe **filha** herde atributos e comportamentos de outra classe **pai**. Isso evita repetição de código e cria uma hierarquia lógica.

👉 Proteger os dados internos da classe, permitindo acesso controlado.



HERANÇA



HERANÇA



Barbie

- nome: String
- idade: int
- profissao: String
- + trabalhar(String trabalho): void

```
class Barbie {  
    String nome;  
    String profissao;  
    int idade;  
  
    void trabalhar() {  
        System.out.println(nome + " esta trabalhando como"  
+  
        profissao + "!");  
    }  
}
```



```
public class Main {  
    public static void main(String[] args) {  
  
        Barbie barbie = new Barbie(); // instanciando  
        barbie.nome = "Prof Barbie";  
        barbie.profissao = "Professora";  
        barbie.idade = 25;  
  
        barbie.trabalhar();  
    }  
}
```



HERANÇA



HERANÇA

Barbie

- nome: String
- idade: int
- profissao: String

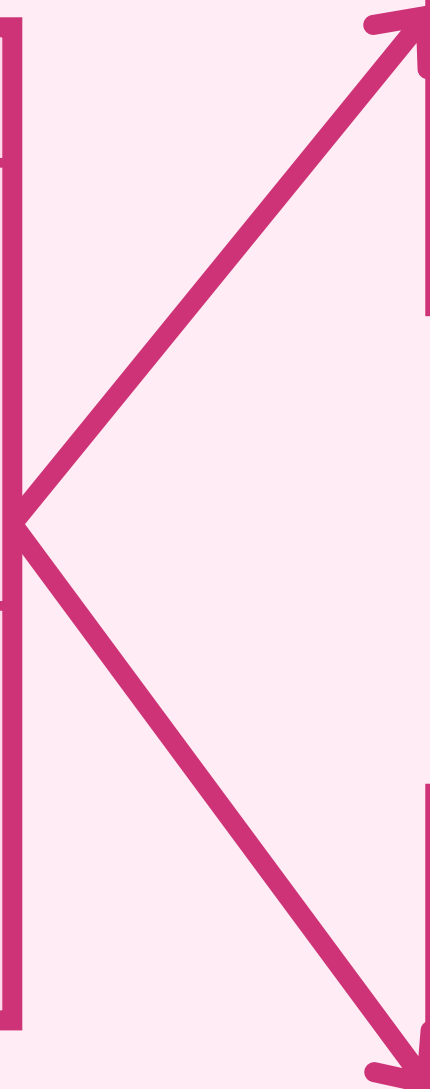
+ trabalhar(String trabalho): void

BarbieProfessora

+ ensinar(String materia): void

BarbieAstronauta

+ caminhadaLunar(): void



HERANÇA

```
class BarbieProfessora extends Barbie{  
    private String profissao = "Professora";  
  
    void ensinar(String materia) {  
        System.out.println(nome + " esta ensinando" +  
            materia + "!");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        BarbieProfessora barbie1 = new BarbieProfessora ();  
        barbie1.nome = "Prof Barbie";  
        barbie1.idade= 25;  
  
        barbie1.ensinar();  
    }  
}
```



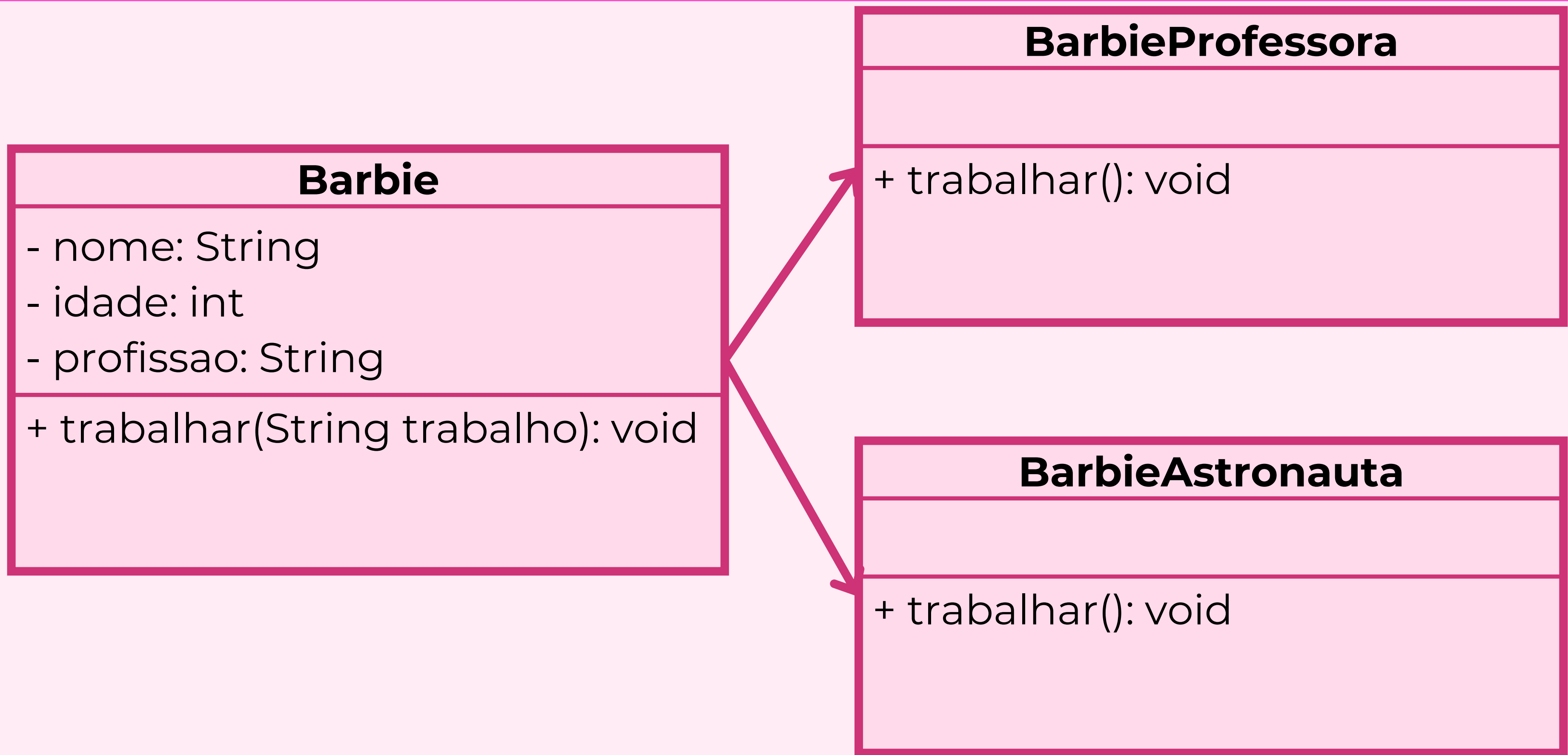

POLIMORFISMO

Conceito: O polimorfismo significa "muitas formas". Ele permite que um mesmo método ou ação tenha comportamentos diferentes dependendo do objeto que o utiliza.

Isso acontece quando uma subclasse sobrescreve (override) um método herdado, mudando o comportamento padrão. Assim, temos a mesma chamada, mas com resultados diferentes dependendo do objeto que a executa.

👉 Um mesmo método se comporta de formas diferentes em cada classe.

POLIMORFISMO



POLIMORFISMO

```
class BarbieAstronauta extends Barbie{  
    private String profissao = "Astronauta";  
  
    void trabalhar() {  
        System.out.println(nome + " explorando o espaço!");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        BarbieAstronauta barbie2 = new BarbieAstronauta ();  
        barbie2.nome = "Capitã Barbie";  
        barbie2.idade= 35;  
  
        barbie1.trabalhar();  
        barbie2.trabalhar();  
    }  
}
```


A nighttime photograph of a city street, likely in São Paulo, featuring tall buildings and light trails from moving vehicles. The image is dark, with the primary light sources being the city lights and the long-exposure light trails.

SENAI

DEPARTAMENTO REGIONAL
DE SÃO PAULO

www.sp.senai.br