

# Java Advanced

21/02/2025

**Prof. Dr. Marcel Stefan Wagner**

**Aula 04 – Revisão de POO**

**FIAP**

# Tópicos Abordados

- 1 Introdução à Programação Orientada a Objetos
- 2 Classes
- 3 Encapsulamento
- 4 Herança
- 5 Abstração
- 6 Polimorfismo
- 7 Temas para a Próxima Aula
- 8 Referências Bibliográficas

# Introdução

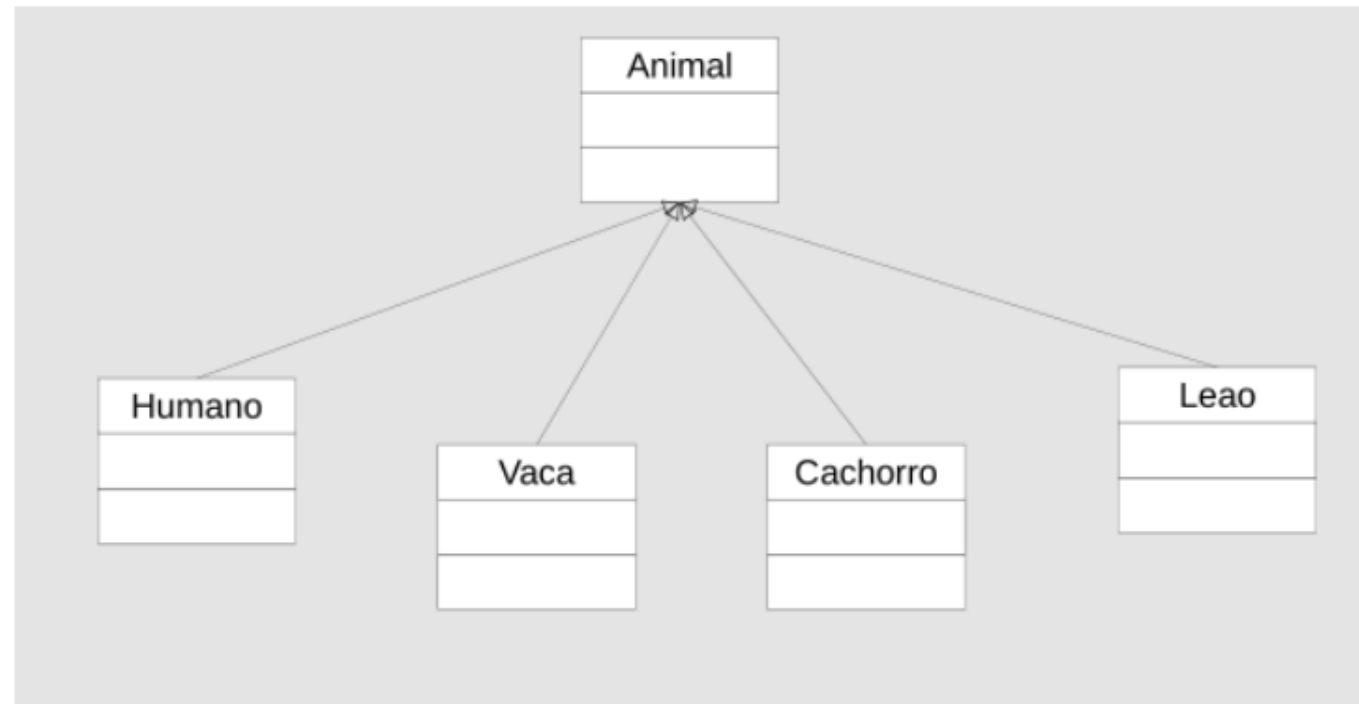


## Abstração

- Essa característica permite que grandes sistemas sejam especificados em um nível muito geral, muito antes de ocorrer a implementação dos métodos individuais.
- Classes que não podem ser instanciadas!
- Permite definir métodos sem implementação - que devem ser redefinidos nas subclasses.

# Classes Abstratas

## Abstração



# Classes Abstratas

## Abstração em Java

```
public abstract class Animal {  
    public abstract void falar()  
}  
  
public class Humano extends Animal {  
    public void falar() {  
        System.out.println("Eu posso falar - bla bla bla.");  
    }  
}  
  
public class Vaca extends Animal {  
    public void falar() {  
        System.out.println("Eu posso mugir - muuuuuu.");  
    }  
}
```

# Classes Abstratas

## Abstração em Java

```
public abstract class Animal {  
    public abstract void falar();  
}
```

Define a classe abstrata

Define o método abstrato

```
public class Humano extends Animal {  
    public void falar() {  
        System.out.println("Eu posso falar - bla bla bla.");  
    }  
}
```

```
public class Vaca extends Animal {  
    public void falar() {  
        System.out.println("Eu posso mugir - muuuuuu.");  
    }  
}
```

# Classes Abstratas

## Abstração em Java

```
public class Cachorro extends Animal {  
    public void falar() {  
        System.out.println("Eu posso latir - au au au.");  
    }  
}  
  
public class Leao extends Animal {  
    public void falar() {  
        System.out.println("Eu posso rugir - roooooaaaarr.");  
    }  
}
```



# Classes Abstratas

## Abstração em Java

```
public class Main {  
  
    public static void main (String[] args) {  
        Humano h = new Humano();  
        h.falar();  
  
        Vaca v = new Vaca();  
        v.falar();  
  
        Cachorro c = new Cachorro();  
        c.falar();  
  
        Leao l = new Leao();  
        l.falar();  
    }  
}
```

# Classes Abstratas

## Abstração em Java

- Será mostrado na tela:

Eu posso falar - bla bla bla.

Eu posso mugir - muuuuuu.

Eu posso latir - au au au.

Eu posso rugir - roooooaaaarr.

# Classes Abstratas

## Exercícios

- **Exercício 3** - Faça um programa em Java para um aplicativo de desenho, onde temos uma Forma abstrata (cor e método para calcular a área).
  - Deve-se ter o círculo, quadrado, retângulo e triângulo.

## Polimorfismo

- A palavra polimorfismo significa ter muitas formas.
  - Em palavras simples, podemos definir polimorfismo como a capacidade de uma mensagem ser exibida em mais de uma forma.
- O polimorfismo é considerado uma das características mais importantes da programação orientada a objetos.

## Polimorfismo

- Usando uma definição mais formal:
  - Polimorfismo é quando duas ou mais classes herdam da mesma classe mãe.
  - Ambas invocam métodos com nomes idênticos.
  - Porém, com comportamentos diferentes.

# Polimorfismo em Java

## Polimorfismo em Java

```
public class Ave {  
    public void introduzir() {  
        System.out.println("Existem muitas aves.");  
    }  
  
    public void voar() {  
        System.out.println("A maioria das aves podem voar, mas algumas não.");  
    }  
}  
  
public class Cegonha extends Ave {  
    public void voar() {  
        System.out.println("Cegonha pode voar.");  
    }  
}
```

# Polimorfismo em Java

## Polimorfismo em Java

```
public static void main (String[] args) {  
    Ave a = new Ave();  
    a.introduzir();  
    a.voar();  
  
    Cegonha c = new Cegonha();  
    c.introduzir();  
    c.voar();  
  
    Avestruz az = new Avestruz();  
    az.introduzir();  
    az.voar();  
  
    Andorinha and = new Andorinha();  
    and.introduzir();  
    and.voar();  
}
```

# Polimorfismo em Java

## Polimorfismo em Java

- Será mostrado na tela:

Existem muitas aves.

A maioria das aves podem voar, mas algumas não.

Cegonha pode voar.

Existem muitas aves.

Avestruz não pode voar.

Andorinha pode voar.



# Polimorfismo em Java

## Exercícios

- **Exercício 4** - Usando polimorfismo, faça um programa em java para venda de imóveis:
  - Crie a classe Imovel, que possui um endereço e um preço.
  - crie uma classe Novolmovel, que herda Imovel e possui um adicional no preço. Crie métodos de acesso e impressão deste valor adicional.
  - crie uma classe Velholmovel, que herda Imovel e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.

## Referências

George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair. **Sistemas Distribuídos: Conceitos e Projeto**. Bookman Editora, 5 edition, 2013.

Harvey M Deitel, Paul J Deitel, David R Choffnes, et al. **Sistemas Operacionais**. Pearson/Prentice Hall, 3 edition, 2005.

Maarten Van Steen and A Tanenbaum. **Sistemas Distribuídos: Princípios e Paradigmas**. Pearson/Prentice Hall, 2 edition, 2007.

Harvey M Deitel and Paul J Deitel. **Java, como programar**. Ed. Pearson/Prentice Hall, 8 edition, 2010.

.....

# Obrigado!

Agradecimento pela parceria e elaboração de materiais aos professores:  
Prof. Me. Gustavo Torres Custódio  
Prof. Thiago Yamamoto

.....

Contato: [profmarcel.wagner@fiap.com.br](mailto:profmarcel.wagner@fiap.com.br)

Cursos:

Tecnologia em Análise e Desenvolvimento de Sistemas (TDS)

Tecnologia em Defesa Cibernética (TDC)

Engenharia de Software (ES)

