

# **POO – Módulo 6**

## **Polimorfismo**

**Módulo ministrado pelo**  
**Prof. Edson Emílio Scalabrin**  
**PUCPR**

**Material produzido pelo**  
**prof. Alcides Calsavara (BCC/PUCPR)**

# Conceitos

## **1. Polimorfismo**

## **2. Referência polimórfica**

## **3. Atribuição polimórfica**

- compatibilidade entre tipos
- compatibilidade entre classes
- type cast e conversão do tipo
- verificação da classe de um objeto

## **4. Chamada de método polimórfica**

- dynamic binding

## **5. Coleção de objetos polimórfica**

# Polimorfismo

“Qualidade ou estado de ser capaz de assumir diferentes formas”

# Polimorfismo

Exemplo: Qual a forma de um **animal**?

# Polimorfismo

Exemplo: Qual a forma de um **animal**?



# Polimorfismo

Exemplo: Qual a forma de um **animal**?

Uso de um termo genérico para designar uma diversidade de tipos de objetos

# Polimorfismo

Exemplo: Qual a forma de um **animal**?

Uso de um termo genérico para designar uma diversidade de tipos de objetos

Pode-se usar ***tipo*** ou ***classe*** no lugar de ***forma*** :

- Qual o **tipo** de um animal?
- Qual a **classe** de um animal?

# Polimorfismo

Exemplo: Qual a forma de um **veículo**?



# Polimorfismo

Exemplo: Qual a forma de um **veículo**?



# Polimorfismo

## Mais exemplos:

- Qual a forma de uma **planta**?
- Qual a forma de um **móvel**?
- Qual a forma de um **imóvel**?
- Qual a forma de um **arquivo**?
- Qual a forma de um **polígono**?
- Qual a forma de uma **profissão**?
- Qual a forma de um **idioma**?
- Qual a forma de uma **música**?

# Polimorfismo em Programação Orientada a Objetos

Onde se espera uma instância de certa classe pode aparecer uma instância de qualquer subclasse daquela classe.

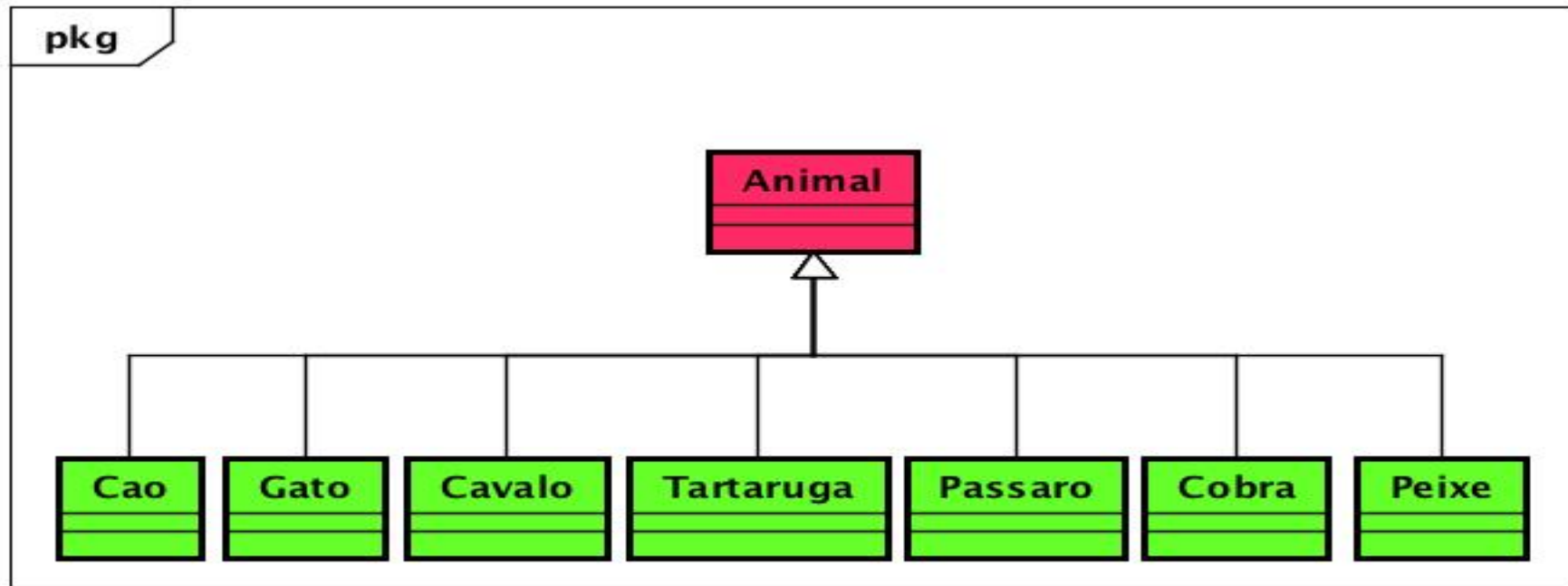
# Polimorfismo em Programação Orientada a Objetos

Onde se espera uma instância de certa classe pode aparecer uma instância de qualquer subclasse daquela classe.

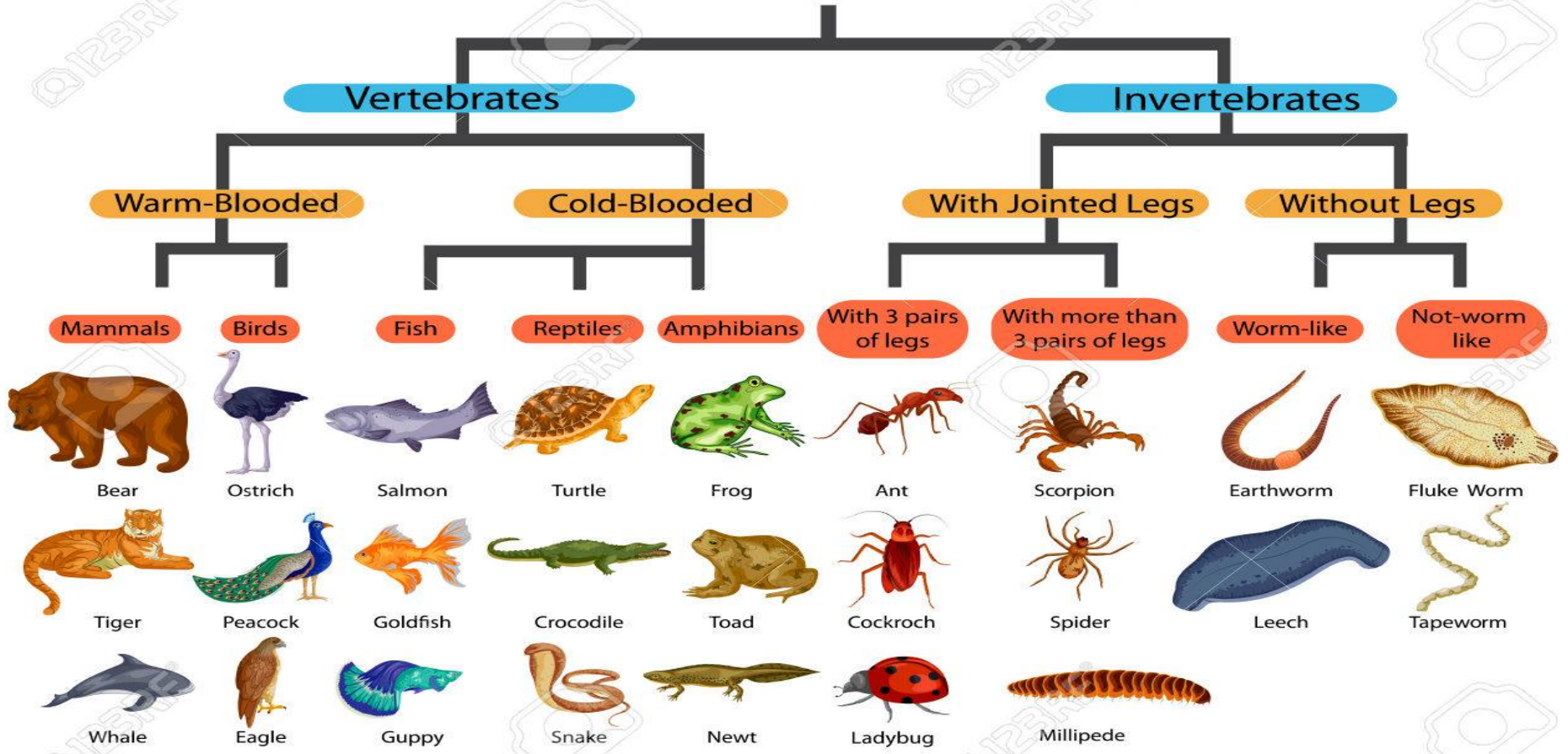
## ***Exemplo:***

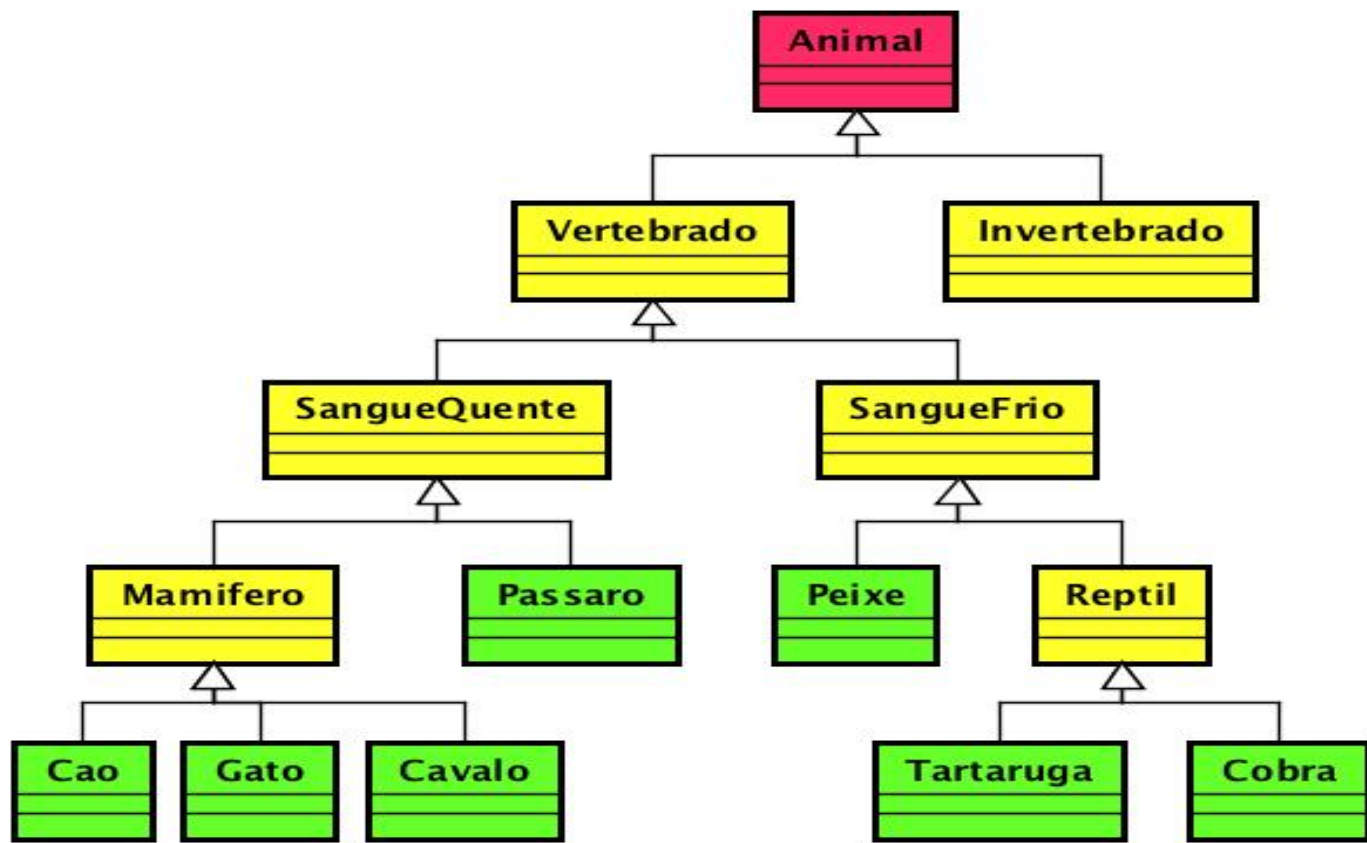
- Onde se espera uma instância de **Animal** pode aparecer uma instância de Cão, Gato, Tartaruga, Cavalo, Pássaro, Cobra, Peixe, ...
- Uma **coleção** de objetos da classe **Animal** pode conter objetos das classes Cão, Gato, Tartaruga, Cavalo, Pássaro, Cobra, Peixe, ...

# Relação entre Polimorfismo e Hierarquia de Classes



# Classification of Animals





# Polimorfismo em Programação Orientada a Objetos

Onde se espera uma instância de certa classe pode aparecer uma instância de qualquer subclasse daquela classe.

Exemplo:

- Onde se espera uma instância de **Animal** pode aparecer uma instância de Cão, Gato, Tartaruga, Cavalo, Pássaro, Cobra, Peixe, **Vertebrado**, **Invertebrado**, **SangueQuente**, **SangueFrio**, **Mamifero** ou **Reptil**.
- Onde se espera uma instância de **Mamífero** pode aparecer uma instância de Cão, Gato ou Cavalo



# Referência Polimórfica

O tipo de uma referência é uma classe.

```
Animal a;      // a é uma referência da classe Animal  
Cao c;         // c é uma referência da classe Cao  
Peixe p;       // p é uma referência da classe Peixe  
Mamifero m;    // m é uma referência da classe Mamifero
```

O tipo de uma referência polimórfica é uma classe que possui subclasses.

```
Animal a;      // a é uma referência polimórfica  
Cao c;         // c não é uma referência polimórfica  
Peixe p;       // p não é uma referência polimórfica  
Mamifero m;    // m é uma referência polimórfica
```

# Referência Polimórfica

Uma referência polimórfica pode referenciar um objeto da sua própria classe ou de qualquer subclasse direta ou indireta.

```
Mamifero m;  
    // m pode referenciar um objeto da classe Mamifero  
    // m pode referenciar um objeto da classe Cao  
    // m pode referenciar um objeto da classe Gato  
    // m pode referenciar um objeto da classe Cavalo
```

```
Animal a;  
    // a pode referenciar um objeto da classe Animal  
    // a pode referenciar um objeto da classe Vertebrado  
    // a pode referenciar um objeto da classe Invertebrado  
    // ...  
    // a pode referenciar um objeto da classe Cobra
```

# Atribuição Polimórfica

(1) Vinculação de um novo objeto a uma referência polimórfica.

```
Mamifero m;  
  
m = new Mamifero( ... );  
  
m = new Cao( ... );  
  
m = new Gato( ... );  
  
m = new Cavalo( ... );
```

# Atribuição Polimórfica

(2) Vinculação de um objeto existente a uma referência polimórfica.

```
Mamifero m
```

```
Cao c = new Cao( ... );  
m = c;
```

```
Gato g = new Gato( ... );  
m = g;
```

```
Cavalo h = new Cavalo( ... );  
m = h;
```

# Atribuição Polimórfica

(3) Vinculação de um objeto a um **parâmetro** de um método que é uma referência polimórfica.

```
class Zelador {  
    public void alimentar(Mamifero m) { ... }  
}
```

```
Cao c = new Cao( ... );  
Gato g = new Gato( ... );  
Cavalo h = new Cavalo( ... );  
Zelador z = new Zelador( ... );  
z.alimentar(c);  
z.alimentar(g);  
z.alimentar(h);
```

```
Zelador z = new Zelador( ... );  
Mamifero k = new Cao( ... );  
z.alimentar(k);  
k = new Gato( ... );  
z.alimentar(k);  
z.alimentar(new Cavalo( ... ));
```

# Atribuição Polimórfica

(4) Vinculação de um objeto ao **retorno** de um método que é uma referência polimórfica.

```
class Fazendeiro{  
    public Mamifero comprar(int k){  
        Mamifero m;  
        switch(k) {  
            case 1: m = new Cao(...); break;  
            case 2: m = new Gato(...); break;  
            default: m = new Cavalo(...);  
        }  
        return m;  
    }  
}
```

```
Fazendeiro f = new Fazendeiro(...);  
Mamifero x = f.comprar(2);  
Mamifero y = f.comprar(3);
```

# Compatibilidade entre Tipos

```
int k = 10;  
  
float f = k;  
double x = f;  
double y = k;  
f = 10;  
x = 7.5F;  
y = 10;
```

```
float f = 7.5F;  
int k = f;
```

```
float f = 7.5F;  
int k = (int) f;
```

O conjunto dos números reais inclui o conjunto dos números inteiros.

**int** é compatível com **float**  
**float** é compatível com **double**  
**int** é compatível com **double**

**float não** é compatível com **int**

operação de *type cast* (conversão de tipo)

# Compatibilidade entre Classes

```
Animal a;  
Mamifero m;  
Cao c;
```

```
a = m;  
m = c;  
a = c;
```

```
Mamifero m;  
Cao c = m;
```

```
Mamifero m;  
Cao c = (Cao)m;
```

O conjunto dos animais inclui conjunto dos mamíferos, que inclui o conjunto dos cães.

**Cao** é compatível com **Mamifero**  
**Mamifero** é compatível com **Animal**  
**Cao** é compatível com **Animal**

**Mamifero** **não** é compatível com **Cao**

operação de *type cast* (conversão de tipo)



# Verificação da Classe de um Objeto

Expressão booleana: *referência **instanceof** classe*

É verdadeiro quando o objeto referenciado é uma instância da classe.

```
class Veterinario {  
    public void examinar(Mamifero m) {  
        if (m instanceof Cao)  
        { Cao x = (Cao)m; x.latir(); }  
        else  
        if (m instanceof Gato)  
        { Gato y = (Gato)m; y.miar(); }  
        else  
        if (m instanceof Cavalo)  
        { Cavalo z = (Cavalo)m; z.relinchar(); }  
    }  
}
```

# Chamada de Método Polimórfica

Ocorre quando um método é chamada a partir de uma referência polimórfica e o método é sobrescrito nas subclasses.

```
public class Animal {  
}
```

```
public class Vertebrado extends Animal {  
}
```

```
public class SangueQuente extends Vertebrado {  
}
```

```
class Mamifero extends SangueQuente {  
    public void soar( ) {  
        System.out.println("Som de mamífero");  
    }  
}
```

```
class Veterinario {  
    public void examinar(Mamifero m) {  
        m.soar( );  
    }  
}
```

```
class Gato extends Mamifero {  
    public void miar( )  
    { System.out.println("miados"); }  
    public void soar( ) { miar(); }  
}
```

```
class Cao extends Mamifero {  
    public void latir( )  
    { System.out.println("latidos"); }  
    public void soar( ) { latir(); }  
}
```

```
class Cavalo extends Mamifero {  
    public void relinchar()  
    { System.out.println("relinchos"); }  
    public void soar( ) { relinchar(); }  
}
```

# Chamada de Método Polimórfica

Qual implementação do método **soar** será chamada?

Depende da classe do objeto vinculado ao parâmetro **m**.

```
class Veterinario {  
    public void examinar(Mamifero m) {  
        m.soar();  
    }  
}
```

```
Veterinario v = new Veterinario( );  
Cao c = new Cao( );  
v.examinar( c );  
Gato g = new Gato( );  
v.examinar( g );
```

*dynamic binding*  
*late binding*

# Coleção de Objetos Polimórfica

A coleção é composta por referências polimórficas.

```
ArrayList<Mamifero> rebanho = new ArrayList<Mamifero> ();  
  
rebanho.add(new Cao());  
rebanho.add(new Gato());  
rebanho.add(new Cavalo());  
  
for (Mamifero m: rebanho) m.soar();
```