



LINGUAGEM DE PROGRAMAÇÃO I

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNESP

public class Animal {

Variáveis: nome, idade, genero, peso, especie, habitat



som ()

“miau”



som ()

“au au”

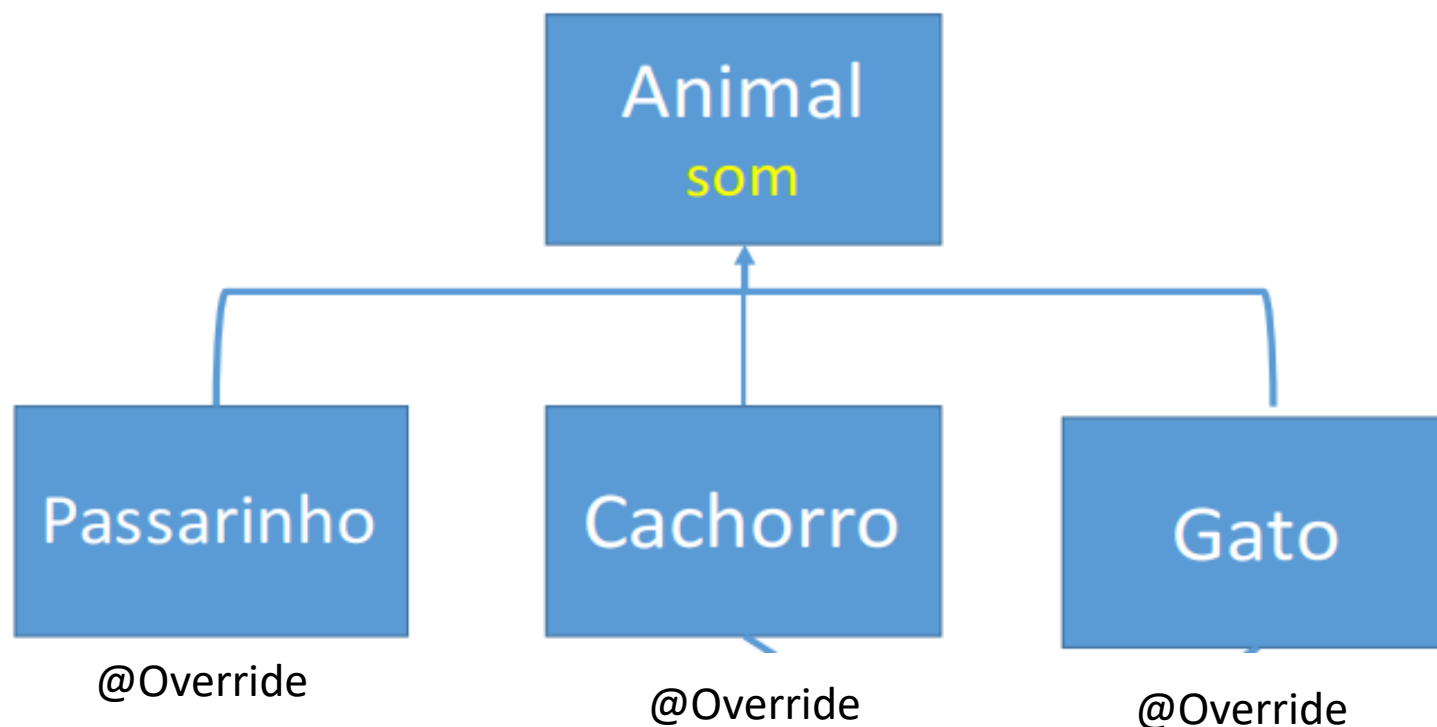


som ()

“piu”

Classes Abstratas

Uma classe abstrata é uma classe que não é instanciada diretamente, mas serve como modelo para outras classes que a estendem. Ela pode ter métodos abstratos.



// Classe abstrata

```
public abstract class Animal {  
    protected String nome;  
    protected int idade;  
  
    public Animal(String nome, int idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

//get e set

```
public abstract void emitirSom("Eu emito um som");  
}
```

```
class Cachorro extends Animal{  
    public Cachorro(String nome, int idade) {  
        super(nome, idade);  
    }  
  
    @Override  
    public void emitirSom() {  
        System.out.println("Au au");  
    }  
}
```

```
public abstract class Animal {
```

Variáveis: nome, idade, genero, peso, especie, habitat



```
@Override som ( )
```



```
@Override som ( )
```



```
@Override som ( )
```

E AGORA? →

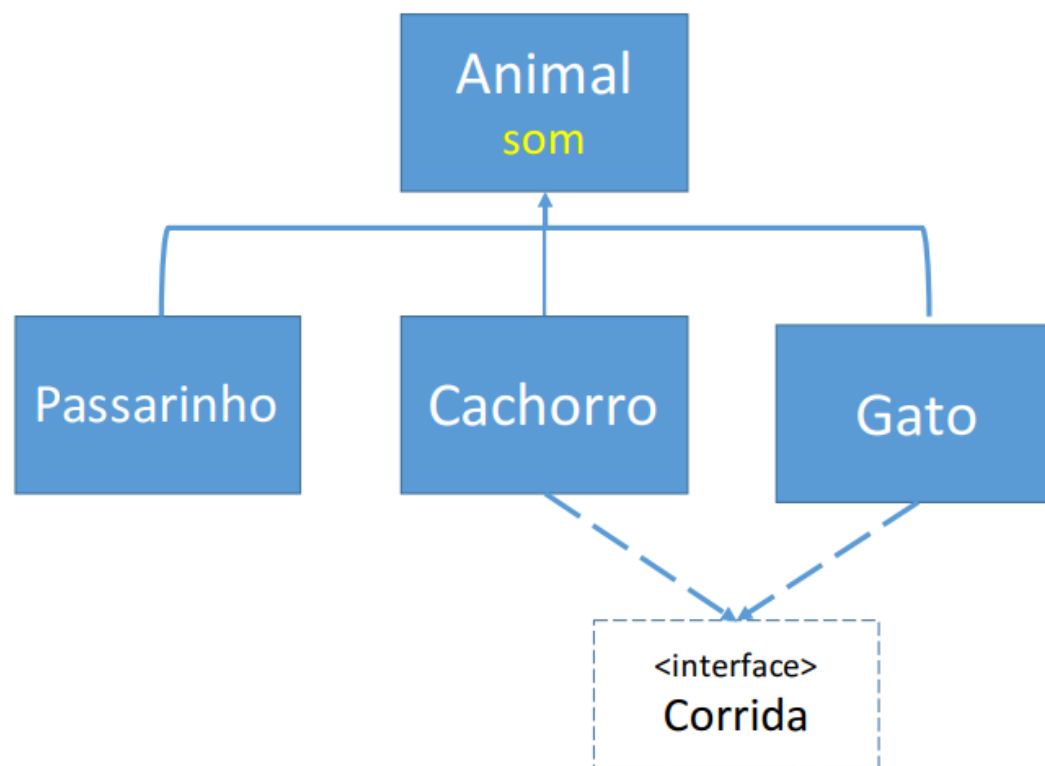
```
Correr ( )
```

```
Correr ( )
```

```
????
```

Interface

Uma interface é um tipo especial de classe abstrata que define apenas a **assinatura de métodos**, sem implementação.



// Classe abstrata

```
public abstract class Animal {  
    protected String nome;  
    protected int idade;
```

```
    public Animal(String nome, int idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }
```

//get e set

```
    public abstract void emitirSom("Eu emito um som");  
}
```

```
public interface Corrida{  
    public void correr(); //uma interface possui apenas a assinatura do método  
}
```

```
public class Cachoro extends Animal implements Corrida{  
    public Cachorro(String nome, int idade) {  
        super(nome, idade);  
    }  
}
```

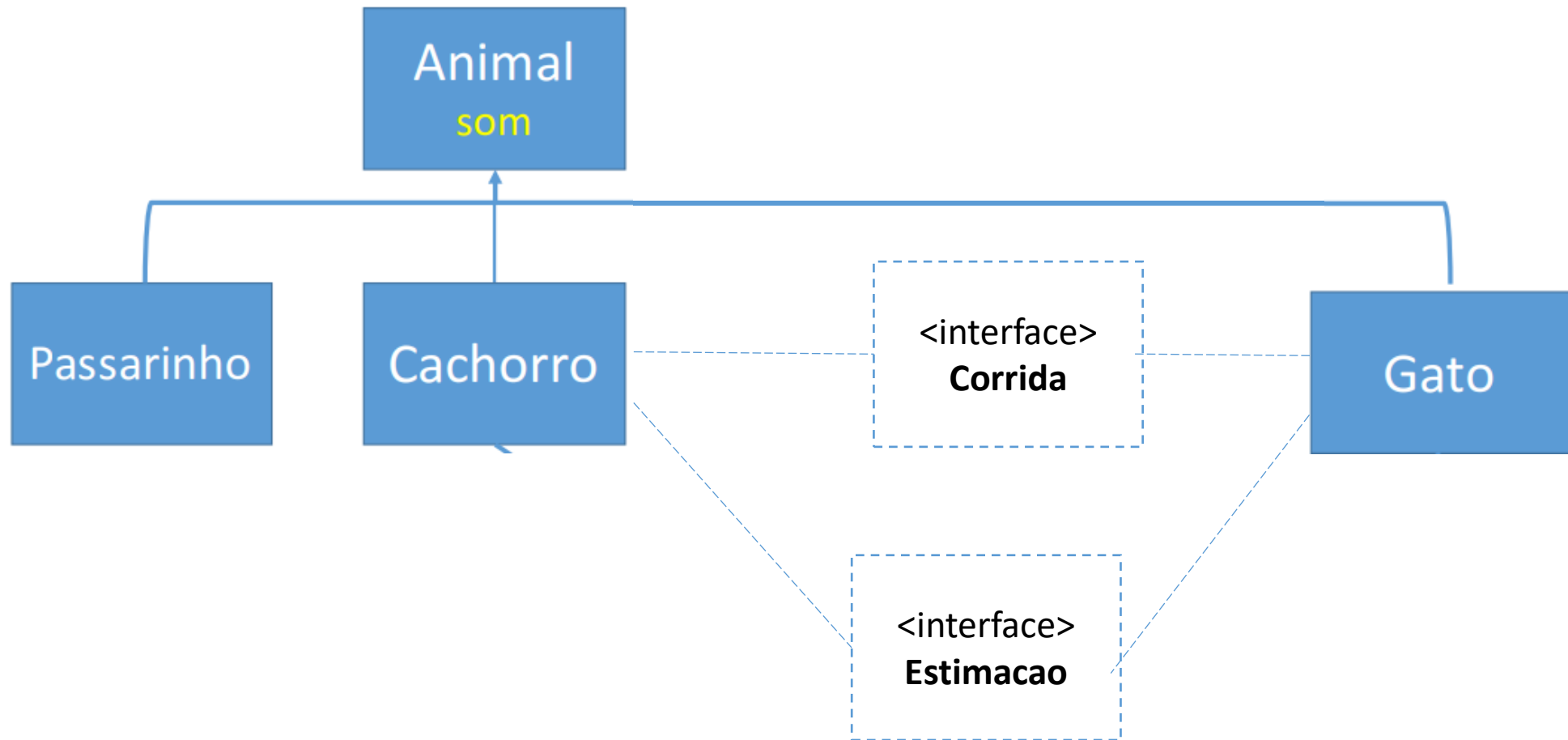
@Override

```
public void emitirSom() {  
    System.out.println("Au au");  
}
```

//@Override

```
public void correr("Estou correndo rápido") {  
}
```


ATENÇÃO!



```
public interface Corrida{  
    public void correr(); //uma interface possui apenas a assinatura do método  
}
```

```
public interface Estimacao{  
    public void brincar(); //uma interface possui apenas a assinatura do método  
}
```

```
public class Cachoro extends Animal implements Corrida, Estimacao{
```

```
    // implementação
```

```
}
```

Dúvidas?



Exercício

- 1) Elabore um código que tenha um gato, um cachorro e um passarinho. Todos possuem nome, todos emitem sons, mas apenas gato e cachorro podem correr. O passarinho pode voar.
- 2) Elabore um código que tenha uma empresa que vende produtos. Podendo ser livro, eletrônico ou roupa. Todos eles possuem nome e preço, mas quando existe promoção, livros possuem desconto de 10% e eletrônicos, apenas 5% e roupas sempre estão com preço fixo. Neste sentido, o preço final de todos os produtos dependem do desconto aplicado quando há promoção.