LINGUAGEM DE PROGRAMAÇÃO I

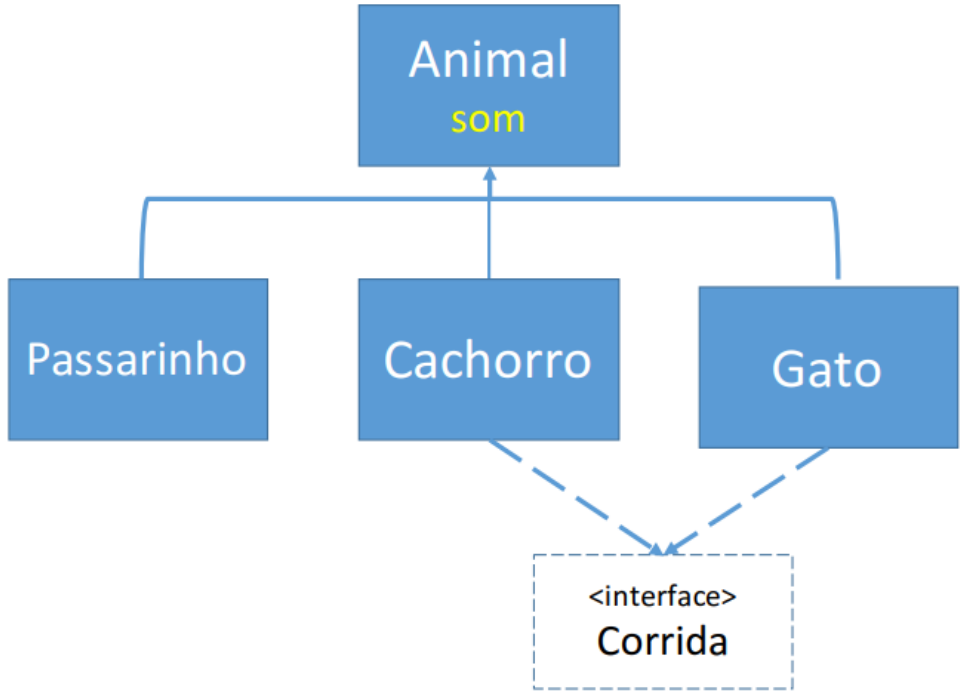
MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNIESP

O que sabemos até aqui:

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

            Cachorro  c = new Cachorro();
            Gato g = new Gato();
           Passarinho p = new Passarinho();
   }
}
```



POLIMORFISMO

"Várias Formas"

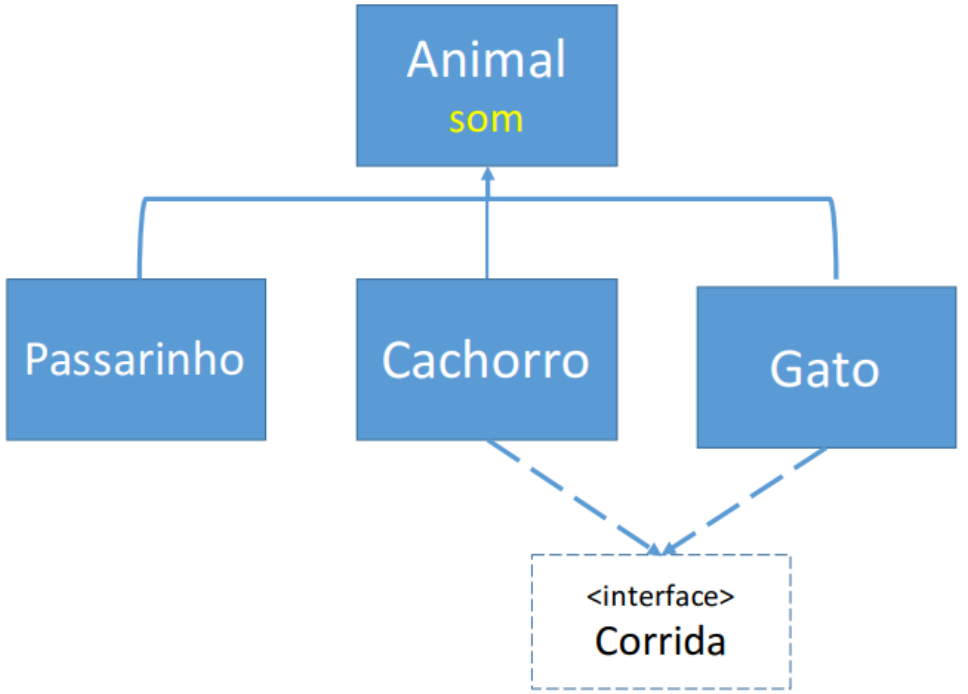
Permite que classes pertencentes a uma mesma linha de herança possuam comportamentos diferentes para o mesmo método

Um animal pode se “transformar” em passarinho, cachorro ou gato...

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

        Animal a1 = new Cachorro();
        Animal a2 = new Cachorro();
        Animal a3 = new Gato();
        Animal a4 = new Passarinho();

  }
}
```

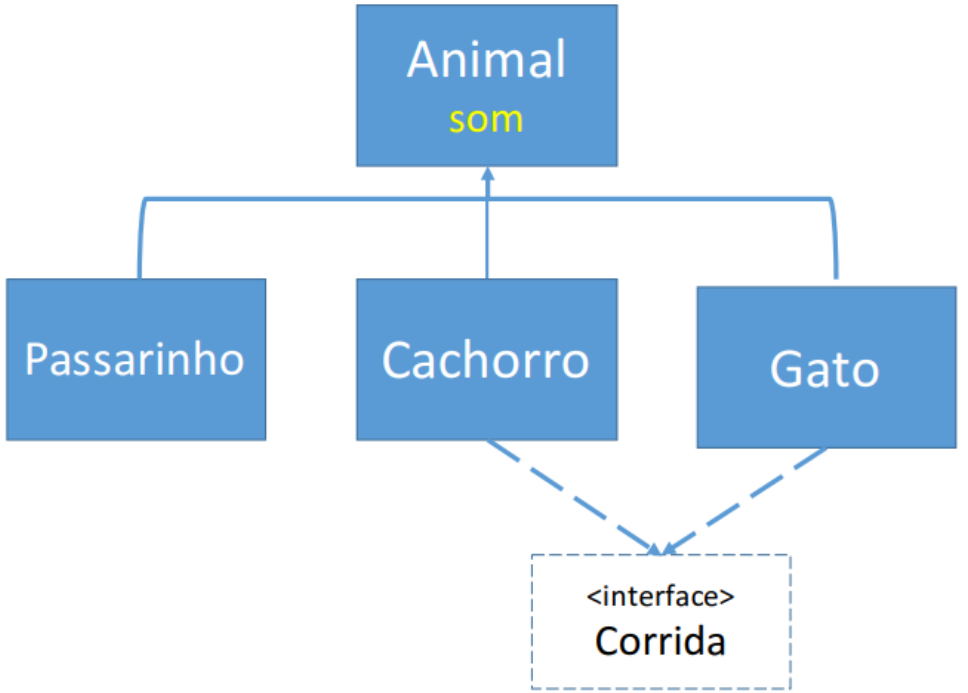


Um animal pode se “transformar” em passarinho, cachorro ou gato...

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

        Animal c1 = new Cachorro();
        Animal c2 = new Cachorro();
        Animal g1 = new Gato();
        Animal p1 = new Passarinho();

  }
}
```



Exemplo 1

```
public asbtract class Animal {
 public void comer() {
  System.out.println();
 }
}
```

```
public class Cao extends Animal {

 @Override
 public void comer() {
  System.out.println("Cão comendo!");
 }
}
```

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

    Animal c = new Cao();

    c.comer();

  }
}
```

Exemplo 2

```
public asbtract class Animal {
 public abstract void comer();
}
```

```
 public class Cao extends Animal {

  @Override
  public void comer() {
   System.out.println("Cão comendo!");
  }
 }
```

```
  //Classe com método para polimorfismo
  public class Teste {
    public void fazerComer (Animal a) {
    a.comer();
    }
  }
```

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

    //Objeto polimórfico
     Teste t = new Teste();

     t.fazerComer(new Cao());
    }
  }
```

Elabore um código para este problema:

Todos são instrumentos musicais (Classe PAI). Eles são cadastrados no sistema com nome e código. Porém, Existem a categoria de instrumentos de corda e de sopro. Os instrumentos de corda, podem precisar trocar a corda. Estes Instrumentos precisam cadastrar a quantidade de cordas.

void tocar
void afinar

void trocarCorda

void tocar
void afinar

void trocarCorda

void tocar
void afinar

RESPOSTA

```java
public interface Cordas {
    public void trocar_cordas();
}
```

```java
public abstract class Instrumento {
    public abstract void tocar();
    public abstract void afinar();

}
```

```java
public class Polimorfismo {

    public void tocar(Instrumento  i){
     i.tocar();
    }

    public void afinar(Instrumento i){
        i.afinar();
    }

    public void trocar_cordas(Cordas c){
        c.trocar_cordas();
    }
}
```

```java
public class Violao extends Instrumento implements Cordas{
    @Override
    public void tocar(){
        System.out.println("tocando violão");
    }
    @Override
    public void afinar(){
        System.out.println("afinando violão");
    }
    @Override
    public void trocar_cordas(){
        System.out.println("trocando as cordas do violão");
    }
}
```

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


        Polimorfismo p = new Polimorfismo();

        p.tocar(new Harpa());
        p.tocar(new Violao());
        p.tocar(new Flauta());

        p.afinar(new Harpa());
        p.afinar(new Violao());
        p.afinar(new Flauta());

        p.trocar_cordas(new Harpa());
        p.trocar_cordas(new Violao());

    }
}
```