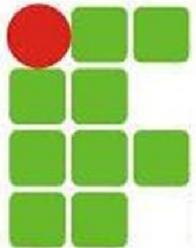
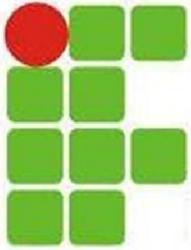

Programação Orientada a Objetos

Professor: Diego Oliveira



**Conteúdo 11:
Polimorfismo**

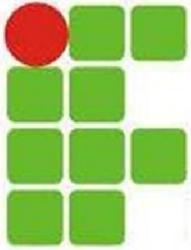




Polimorfismo

- O polimorfismo é mais um princípio fundamental da orientação a objetos
- Significa, ao pé da letra, “várias formas”
- Não confundir com sobrecarga (*overload*)
- Ele permite que classes pertencentes a uma mesma linha de herança possuam comportamentos diferentes para o mesmo método

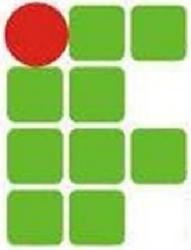




Polimorfismo

- Exemplo simples: em um jogo de xadrez temos várias peças, cada peça se movimenta, porém cada tipo de peça se movimenta de uma maneira diferente (o peão vai pra frente, o cavalo anda em L, o bisbo em diagonal e assim por diante)
- Desta maneira, todos (peão, cavalo e bispo) são peças, porém cada um se movimenta de uma maneira diferente!

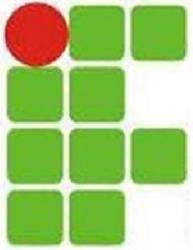




Polimorfismo

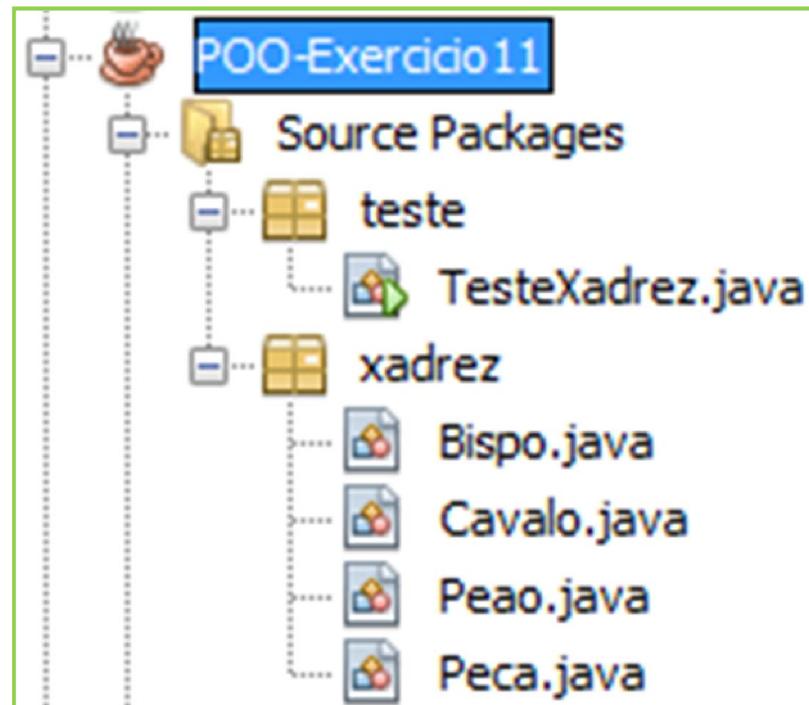
- O polimorfismo em Java se manifesta apenas na chamada de métodos
- Então ao passar uma mensagem para um objeto peça, dizendo para ele se mover, o Java identificará que tipo de peça é e fará o movimento de acordo com o tipo
- Essa decisão é tomada em tempo de execução (*runtime*) pelo Java

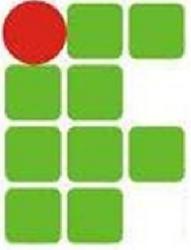




Polimorfismo

- Exemplo do Xadrez:





Polimorfismo

- Exemplo do Xadrez:

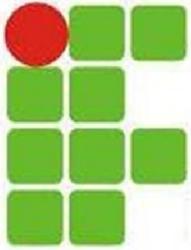
```
1 package xadrez;  
2  
3 public abstract class Peca {  
4     public abstract void mover();  
5 }
```

```
3 public class Peao extends Peca{  
4     @Override  
5     public void mover() {  
6         System.out.println("ANDAR PRA FRENTE");  
7     }  
8 }
```

```
3 public class Cavalo extends Peca{  
4     @Override  
5     public void mover() {  
6         System.out.println("ANDAR EM 'L'");  
7     }  
8 }
```

```
3 public class Bispo extends Peca{  
4     @Override  
5     public void mover() {  
6         System.out.println("ANDAR NA DIAGONAL");  
7     }  
8 }
```





Polimorfismo

- Exemplo do Xadrez:

```
8 public class TesteXadrez {
9     public static void main(String[] args) {
10         Peca peao = new Peao();
11         Peca cavalo = new Cavalo();
12         Peca bispo = new Bispo();
13
14         peao.mover();
15         cavalo.mover();
16         bispo.mover();
17     }
18 }
```

Apache Tomcat 8.0.3.0

POO-Exercicio11 (run)

run:

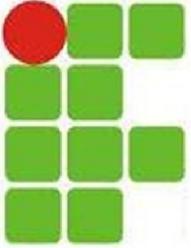
ANDAR PRA FRENTE

ANDAR EM 'L'

ANDAR NA DIAGONAL

BUILD SUCCESSFUL (total time: 0 seconds)





Polimorfismo

- Observe que os três objetos (peao, cavalo e bispo) são do tipo Peca
- Isso pode ser confirmado através do uso do *instanceof* do Java:

```
System.out.println(peao instanceof Peca);  
System.out.println(cavalo instanceof Peca);  
System.out.println(bispo instanceof Peca);
```

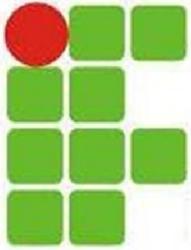
Apache Tomcat 8.0.3.0 POO-Exercicio11 (run)

ANDAR NA DIAGONAL

```
true  
true  
true
```

BUILD SUCCESSFUL (total time: 0 seconds)



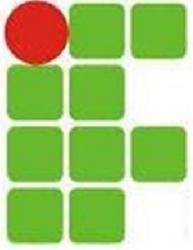


Exercício

- Crie uma estrutura de herança para demonstrar o polimorfismo utilizando classes de animais
- Crie o método comunicar() e movimentar() na classe Animal
- Nas classes descendentes de animais, sobrescreva os métodos citados de acordo com os tipos de animais



- Confirme que todos são tipos de Animal com o uso de *instanceof* do Java



Perguntas?

