

Orientação a objetos

Polimorfismo

Polimorfismo

- A palavra polimorfismo vem do grego:
- poli = muitas, morphos = formas
- Ou seja: muitas formas
- Polimorfismo permite “programar no geral” ao invés de “programar no específico”.
- Exemplos de polimorfismo em Java são:
 - Sobrescrita de métodos
 - Sobrecarga de métodos

Polimorfismo

- Sobrescrita de métodos (override): métodos com o mesmo nome, mesmos parâmetros e mesmo tipo de retorno, alterando apenas o comportamento do método.
 - Tudo no método é igual, só muda a implementação.
- Sobrecarga de métodos (overload): métodos com o mesmo nome, porém com parâmetros diferentes e/ou tipos de retorno diferentes.
- Dinâmico: em tempo de compilação é escolhido

Polimorfismo - Sobrescrita

- Sobrescrita é o processo de fornecer uma implementação diferente para um método já definido em uma classe base.
- Isso é feito na classe derivada, que herda o método da classe base.
- A sobrescrita é realizada com a mesma assinatura de método da classe base, mas com uma implementação diferente.
- Isso permite que a classe derivada substitua o comportamento da classe base para o método em questão.

Polimorfismo - Sobrescrita

```
3 public class Animal {  
4     public void fazerSom() {  
5         System.out.println("O animal está fazendo um som.");  
6     }  
7 }
```

```
3 public class Cachorro extends Animal {  
4     @Override  
5     public void fazerSom() {  
6         System.out.println("O cachorro está latindo.");  
7     }  
8 }
```

Polimorfismo - Sobrescrita

```
3 public class SobrescritaExemplo {  
4     public static void main(String[] args) {  
5         Animal animal = new Animal();  
6         animal.fazerSom();  
7  
8         animal = new Cachorro();  
9         animal.fazerSom();  
10  
11        Cachorro cachorro = new Cachorro();  
12        cachorro.fazerSom();  
13  
14    }  
15 }
```

- Qual resultado acontece após cada chamada do método fazerSom()?

Polimorfismo - Sobrescrita

- Jogo rápido:
 - Faça implementações dos seguintes animais:
 - Passarinho, hipopótamo e peixe

Polimorfismo - Sobrescrita

- Jogo rápido [2]:
 - Crie uma classe Pessoa com um método trabalhar(). Em seguida, crie uma classe Programador que herda da classe Pessoa e sobrescreve o método trabalhar() para imprimir "Programando...".

Polimorfismo - Sobrescrita

- Jogo rápido [2]:

```
3 public class Pessoa {  
4     public void trabalhar() {  
5         System.out.println("Eu só trabalho...");  
6     }  
7 }
```

```
3 public class Programador extends Pessoa {  
4     public void trabalhar() {  
5         System.out.println("Programando...");  
6     }  
7 }
```

Captura Retangular

Polimorfismo - Sobrescrita

- Jogo rápido [2]:

```
5 public class Principal {  
6     public static void main(String[] args) {  
7         int op;  
8         Scanner sc = new Scanner(System.in);  
9         System.out.println("Digite 1 para criar uma pessoa e 2 para programador:");  
10        op = sc.nextInt();  
11        Pessoa p;  
12        if (op==1) {  
13            p = new Pessoa();  
14            p.trabalhar();  
15        }  
16        else if(op==2) {  
17            p = new Programador();  
18            p.trabalhar();  
19        }  
20    }  
21 }
```

Polimorfismo - Sobrecarga

- A sobrecarga é o processo de fornecer várias implementações para um método com o mesmo nome, mas com diferentes assinaturas de parâmetros.
- Isso permite que a classe tenha vários métodos com o mesmo nome, mas que façam coisas diferentes, dependendo dos parâmetros que são passados.
- O Java decide qual método chamar com base nos parâmetros que são passados durante a chamada do método

Polimorfismo - Sobrecarga

```
3 class Calculadora {  
4     public int somar(int x, int y) {  
5         return x + y;  
6     }  
7  
8     public int somar(int x, int y, int z) {  
9         return x + y + z;  
10    }  
11 }
```

Polimorfismo - Sobrecarga

```
3 public class SobrecargaExemplo {  
4     public static void main(String[] args) {  
5         Calculadora calculadora = new Calculadora();  
6         int resultado1 = calculadora.somar(2, 3);  
7         int resultado2 = calculadora.somar(2, 3, 4);  
8         System.out.println("Resultado1: " + resultado1);  
9         System.out.println("Resultado2: " + resultado2);  
10    }  
11 }  
12
```

Polimorfismo - Sobrecarga

- Jogo rápido: Faça a implementação para solucionar o erro do código gerado pela chamada abaixo:

```
3 public class SobrecargaExemplo {  
4     public static void main(String[] args) {  
5         Calculadora calculadora = new Calculadora();  
6         int resultado1 = calculadora.somar(2, 3);  
7         int resultado2 = calculadora.somar(2, 3, 4);  
8         System.out.println("Resultado1: " + resultado1);  
9         System.out.println("Resultado2: " + resultado2);  
10        → double resultado3 = calculadora.somar(3.5, 7);  
11        System.out.println("Resultado2: " + resultado3);  
12    }  
13 }
```

Polimorfismo - Sobrecarga

- Jogo rápido: Faça a implementação para solucionar o erro do código gerado pela chamada abaixo:
- Resolução:

```
3 class Calculadora {  
4     public int somar(int x, int y) {  
5         return x + y;  
6     }  
7  
8     public int somar(int x, int y, int z) {  
9         return x + y + z;  
10    }  
11  
12    → public double somar (double x, double y) {  
13        return x + y;  
14    }  
15 }
```

Polimorfismo - Sobrecarga

- E se a chamada for assim, em qual método da classe Calculadora ele executa?

```
3 public class SobrecargaExemplo {  
4     public static void main(String[] args) {  
5         Calculadora calculadora = new Calculadora();  
6         int resultado1 = calculadora.somar(2, 3);  
7         int resultado2 = calculadora.somar(2, 3, 4);  
8         System.out.println("Resultado1: " + resultado1);  
9         System.out.println("Resultado2: " + resultado2);  
10        → double resultado3 = calculadora.somar(3, 7);  
11        System.out.println("Resultado2: " + resultado3);  
12    }  
13 }
```


Polimorfismo - Sobrecarga

- E se a chamada for assim, em qual método da classe Calculadora ele executa?
- Por mais que a chamada aguarde um retorno double, a chamada vai para o método que envia dois inteiros pois são os parâmetros enviados.

Polimorfismo - Sobrecarga

- Jogo rápido:
 - Faça a implementação de um método `multiplica(int x, int y)`, após faça uma sobrecarga de um método para multiplicar dois números `double`.

Polimorfismo - Sobrecarga

- Jogo rápido [2]:
 - Crie uma classe Pessoa com um método dizerOla() que imprime "Olá!". Sobrecarregue o método dizerOla() para aceitar um nome por parâmetro e imprimir "Olá, <nome>!".