Dia 19 - Polimorfismo

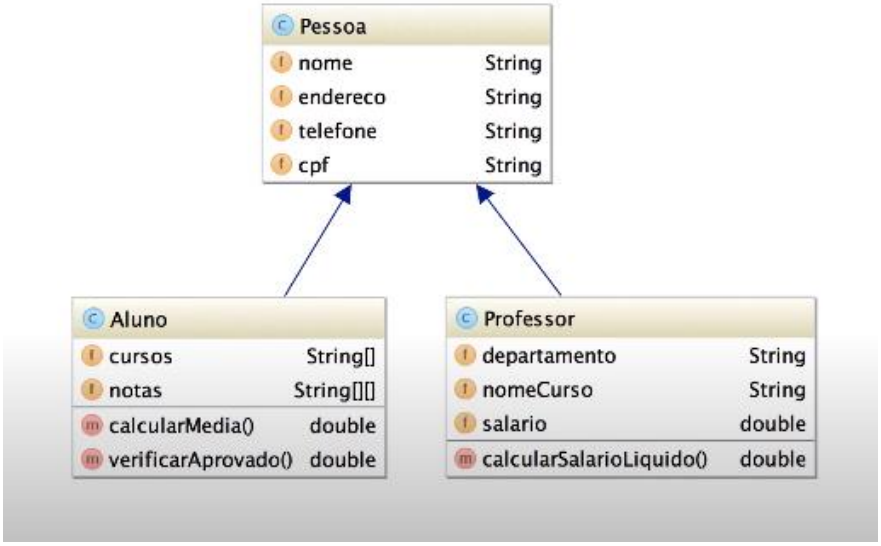
Conteúdo

Introdução:

Polimorfismo significa "muitas formas" e é um termo definido em linguagens orientadas a objetos (como Java, C # e C ++), que permite aos desenvolvedores usar os mesmos elementos de maneiras diferentes. Polimorfismo significa que os objetos podem apresentar comportamentos diferentes ao receber mensagens. No polimorfismo, temos dois tipos:Quando realizamos a mesma operação várias vezes na mesma classe, ocorre polimorfismo estático. A escolha de qual operação chamar depende da assinatura do método sobrecarregado. Quando a subclasse se sobrepõe ao método original, o polimorfismo dinâmico ocorre na herança. Agora, o método selecionado ocorre em tempo de execução e não ocorre mais em tempo de compilação. O método escolhido depende do tipo de objeto que recebe a mensagem. Nos tópicos a seguir, apresentaremos mais dos dois tipos de polimorfismo acima, bem como exemplos de como realmente implementar e usar o polimorfismo em ambientes diferentes.

Vamos Criar 3 Classes que já foi criado na aula passada que é Pessoa,Aluno e Professor(É só copiar o código da aula passada);

Dia 19 - Polimorfismo



Agora vamos trabalhar com o polimorfismo que na pratica ele subscreve a herança. Vou te explicar:

```
1. public class Teste {
2. public static void main(String[] args) {
3. Pessoa pessoa = new Pessoa();
4. Pessoa aluno = new Aluno();
5. Pessoa professor = new Professor();
6. pessoa.setEndereco("Rua 1, num 1");
7. aluno.setEndereco("Rua 2, num 2");
8. professor.setEndereco("Rua 3, num 3");
9. System.out.println(pessoa.obterEtiquetaEndereco());
10.  System.out.println(aluno.obterEtiquetaEndereco());
11.  System.out.println(professor.obterEtiquetaEndereco());
12.  }
13.  }
```

O compilador ira retornar a seguinte informação:

Rua 1 , num 1

Rua 2, num 2

Rua 3, num 3

No código 4. Ocorre o polimorfismo: Pessoa aluno = new Aluno();

Em que eu chamo os métodos da Classe Aluno: ou seja , tem o setNome(); de Pessoa e setNome(); de Aluno esse polimorfismo em tempo de execução ira chamar o Metodo Aluno:

.Caso eu queira criar uma nova regra que o obterEtiquetaEndereco ter uma String chamado

```
1. public String obterEtiquetaEndereco(){

2. String s = "Endereço do Aluno: ";
3. s += super.getEndereco();

4. return s;
5. }
```

Dia 19 - Polimorfismo

Mas o que vai acontecer? Se tem o obterEtiquetaEndereco em Aluno e obterEtiquetaEndereco em Pessoa vai chamar qual? Depende se eu instanciarPessoa pessoa = new Pessoa(); ira retornar somente a rua , enquanto o Pessoa aluno = new Aluno(); ira retornar o Endereço do Aluno: Rua x , num x; Isso se chama Sobrescrita, em que eu herdo o método da classe Pessoa e sobrescrevo na classe Aluno.

O código agora irá retornar o seguinte:

Rua 1 , Num 1

Endereço do Aluno: Rua 2 , num 2

Endereço do Aluno: Rua 3 , num 3

EXERCICIO

Dado o seguinte código(responda o exercício 1,2 e 3)

1. Public class Pessoa{
2. Private int notaFiscal;
3. Public void setNotaFiscal(notaFiscal){
4. this.notaFiscal = notaFiscal
5. }
6. Public void getNotaFical(){
7. Return notaFiscal;
8. }
9. }

==========

1. Public class Aluno extends Pessoa{
2. Private int notaFiscal;
3. Public void setNotaFiscal(notaFiscal){
4. this.notaFiscal = notaFiscal
5. }
6. Public void getNotaFical(){
7. String s =”Nossa Nota Fiscal: ”;
8. s+=super.getEndereco();
9. Return s;
10. }
11. }

==========

```
1. public class Teste {

2. public static void main(String[] args) {

3. Pessoa pessoa = new Pessoa();
4. Pessoa aluno = new Aluno();

5. pessoa.setEndereco("Rua 1, num 1");
```

Dia 19 - Polimorfismo

```
6. aluno.setEndereco("Rua 2, num 2");

7. System.out.println(pessoa.obterEtiquetaEndereco());
8. System.out.println(aluno.obterEtiquetaEndereco());
9. }
10.  }
```

1) No código 4 podemos ver que esta usando o polimorfismo e podemos ver que método getNotaFiscal() esta diferente de Aluno e Pessoa, qual o nome disso?
   a) Sobrecarga de metodo
   b) Sobrelinha de método
   c) Sobrescrita de metodo
   d) Nem uma das respostas

2) Qual o motive de instanciar Pessoa pessoa = new Pessoa(); e Pessoa aluno = new Aluno();
   a) Para sobrescrever outros métodos por exemplo caso eu queira chamar um método de pessoa obterEtiquetaEndereco e o metodo de aluno quero que tenha um método sobrescrito;
   b) Para sobrelinhar outros métodos por exemplo caso eu queira chamar um método de pessoa obterEtiquetaEndereco e o metgodo de aluno quero que tenha um método sobrelinhar;
   c) Não á motivo, porque não ira compilar por conta do tempo de execução do compillador
   d) Nem uma das respostas

3) Na linha 9,10 ir na classe Teste, oque ira compilar?
   a) Rua 1, Num 1
      Rua 2, Num 2
   b) Rua 1, Num 1
      Nossa Nota Fiscal: Rua 2,Num 2
   c) Nossa Nota Fiscal: Rua 1, Num 2
      Nossa Nota Fiscal: Rua 2,Num 2

   d) Nem uma das respostas

PROVA

1) Dado o exercício 1

```
class Top {

 public Top(String s) { System.out.print("B"); }

}

public class Bottom2 extends Top {

 public Bottom2(String s) { System.out.print("D"); }
```

Dia 19 - Polimorfismo

```
public static void main(String [] args) {
new Bottom2("C");
System.out.println(" ");
} }
```

Qual é o resultado

A. BD

B. DB

C. BDC

D. DBC

E. Compilador falha

2) Dado:

```
3. Classe pública Tenor extends Singer {
4. public static String sing () {return "fa"; }
5. public static void main (String [] args) {
6. Tenor t = novo Tenor ();
7. Cantor s = novo Tenor ();
8. System.out.println (t.sing () + "" + s.sing ());
9.}
10.}
11. class Singer {public static String sing () {return "la"; }}
```

Qual é o resultado?

A. fa fa

B. fa la

C. la la

D. A compilação falha

E. Uma exceção é lançada no tempo de execução

3) Dado:

```
class Clidder {
private final void flipper () {System.out.println ("Clidder"); }
```

Dia 19 - Polimorfismo

```
}
public class Clidlet extends Clidder {
 public final void flipper () {System.out.println ("Clidlet"); }
 public static void main (String [] args) {
 novo Clidlet (). flipper ();
}}
```

Qual é o resultado?

A. Clidlet

B. Clidder

C. Clidder

Clidlet

D. Clidlet

Clidder

E. A compilação falha