

# Polimorfismo

## Exemplo de *Software*

Neste exemplo de *software* será desenvolvida a estrutura apresentada na Figura 1.

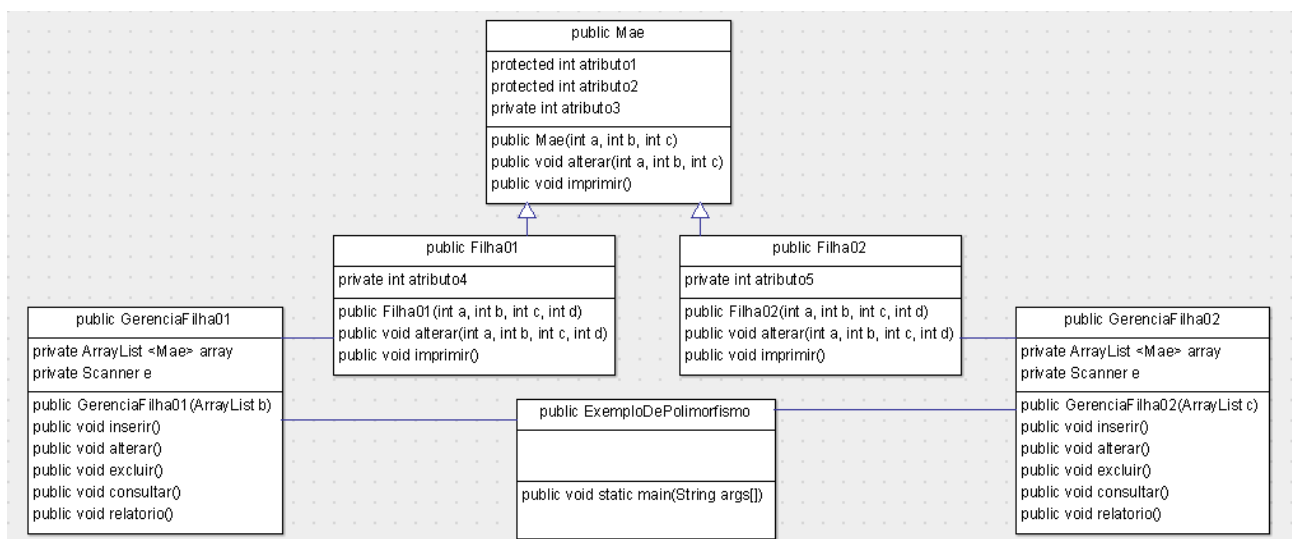


Figura 1: Estrutura do Software.

Neste *software*, na classe “ExemploDePolimorfismo” que possui o método *main* (principal), será criado um *ArrayList* do tipo “Mae” que será referenciado pelas classes “GerenciaFilha01” e “GerenciaFilha02”. E este objeto *ArrayList* criado no método *main*, que é do tipo “Mae”, referenciará tanto objetos do tipo “Filha01” quanto objetos do tipo “Filha02”, concretizando assim um exemplo de polimorfismo.

Sem mais delongas, vamos ao código nas páginas a seguir:

# Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
1 package exemplodepolimorfismo;
2
3 /**
4  *
5  * @author Edwar Saliba Júnior
6  */
7
8 public class Mae {
9     protected int atributo1;
10    protected int atributo2;
11    private int atributo3;
12
13    public Mae (int a, int b, int c){
14        this.atributo1 = a;
15        this.atributo2 = b;
16        this.atributo3 = c;
17    }
18
19    public void alterar(int a, int b, int c) {
20        this.atributo1 = a;
21        this.atributo2 = b;
22        this.atributo3 = c;
23    }
24
25    public void imprimir(){
26        System.out.println("Valor de Atributo1: " + atributo1);
27        System.out.println("Valor de Atributo2: " + atributo2);
28        System.out.println("Valor de Atributo3: " + atributo3);
29    }
30 }
```

Figura 2: Classe "Mae".

# Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
1 package exemplodepolimorfismo;
2
3 /**
4  *
5  * @author Edwar Saliba Júnior
6  */
7 public class Filha01 extends Mae {
8
9     private int atributo4;
10
11     public Filha01(int a, int b, int c, int d){
12         super(a,b,c);
13         this.atributo4 = d;
14     }
15
16     public void alterar(int a, int b, int c, int d){
17         super.alterar(a,b,c);
18         atributo4 = d;
19     }
20
21     @Override
22     public void imprimir(){
23         super.imprimir();
24         System.out.println("Valor de Atributo4: " + atributo4);
25     }
26 }
```

Figura 3: Classe "Filha01".

## Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
1 package exemplodepolimorfismo;
2
3 /**
4  *
5  * @author Edwar Saliba Júnior
6  */
7 public class Filha02 extends Mae {
8
9     private int atributo5;
10
11     public Filha02(int a, int b, int c, int d){
12         super(a,b,c);
13         this.atributo5 = d;
14     }
15
16     public void alterar(int a, int b, int c, int d){
17         super.alterar(b,c,d);
18         atributo5 = a;
19     }
20
21     @Override
22     public void imprimir(){
23         super.imprimir();
24         System.out.println("Valor de Atributo5: " + atributo5);
25     }
26 }
```

Figura 4: Classe "Filha02".

# Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
1 package exemplodepolimorfismo;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.Scanner;
6
7 /**
8  *
9  * @author Edwar Saliba Júnior
10 */
11 public class GerenciaFilha01 {
12     private ArrayList<Mae> array;
13     private Scanner e;
14
15     public GerenciaFilha01(ArrayList b) {
16         this.array = b;
17         e = new Scanner(System.in);
18     }
19
20     public void inserir(){
21         int a, b, c, d;
22
23         System.out.println("==[Digite os Valores]==");
24         System.out.println("Atributo1: ");
25         a = e.nextInt();
26         System.out.println("Atributo2: ");
27         b = e.nextInt();
28         System.out.println("Atributo3: ");
29         c = e.nextInt();
30         System.out.println("Atributo4: ");
31         d = e.nextInt();
32
33         array.add(new Filha01(a, b, c, d));
34     }
35
36     public void alterar(){
37         int a, b, c, d, pos;
38
39         System.out.println("Qual posição do array deseja alterar? ");
40         pos = e.nextInt();
41
42         /* O teste a seguir garante que a posição solicitada existe no
43            ArrayList. */
44         if(pos >= 0 && pos < array.size()){
45             /* Cria uma referência para o objeto na posição "pos" do array.
46                Lembre-se: mm tem apenas uma referência para o objeto que
47                está na posição "pos" do ArrayList, ou seja, mm não tirou
48                o objeto do ArrayList, apenas está o referenciando. Desta
49                forma, toda alteração feita em mm também será refletida no
50                objeto que está na posição "pos" do ArrayList, pois, ambos
51                são o mesmo objeto. Ou melhor, referenciam o mesmo objeto. */
52             Mae mm = (Mae) array.get(pos);
53             /* Testa se o objeto referenciado é diferente de null e se o
54                mesmo é uma instância da classe Filha01. */
55             if(mm != null && mm instanceof Filha01){
56                 /* Faz a conversão do objeto que está na posição "pos" do
57                    ArrayList, do tipo "Mae" para o tipo "Filha01". */
58                 Filha01 ff = (Filha01) array.get(pos);
```

Figura 5: Classe "GerenciaFilha01" - Parte 01.

## Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
59
60         System.out.println("--[Dados Cadastrados]==");
61         ff.imprimir();
62
63         System.out.println("--[Digite os Novos Dados]==");
64         System.out.println("Atributo1: ");
65         a = e.nextInt();
66         System.out.println("Atributo2: ");
67         b = e.nextInt();
68         System.out.println("Atributo3: ");
69         c = e.nextInt();
70         System.out.println("Atributo4: ");
71         d = e.nextInt();
72
73         ff.alterar(a, b, c, d);
74     }
75     else{
76         System.out.println("Não existe instância de \"Filha01\" na " +
77             "posição: " + pos);
78     }
79 }
80 else{
81     System.out.println("Não existe a posição: " + pos + ".");
82 }
83 }
84
85 public void excluir(){
86     int pos, opcao;
87
88     System.out.println("Qual posição do array deseja excluir? ");
89     pos = e.nextInt();
90
91     if(pos >= 0 && pos < array.size()){
92         Mae mm = (Mae) array.get(pos);
93         if(mm instanceof Filha01){
94             Filha01 ff = (Filha01) array.get(pos);
95             System.out.println("--[Dados Cadastrados]==");
96             ff.imprimir();
97
98             do{
99                 System.out.println("Confirma exclusão? (1 - Sim / 2 - Não): ");
100                 opcao = e.nextInt();
101
102                 if(opcao == 1)
103                     array.remove(ff); // ou array.remove(pos);
104                 else
105                     if(opcao == 2)
106                         System.out.println("Exclusão não efetuada!");
107                 }while((opcao != 2) && (opcao != 1));
108             }
109             else{
110                 System.out.println("Não existe instância de \"Filha01\" na " +
111                     "posição: " + pos);
112             }
113         }
114         else{
115             System.out.println("Não existe a posição: " + pos + ".");
116         }
117     }
118 }
```

Figura 6: Classe "GerenciaFilha01" - Parte 02.

## Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
117     }
118
119     public void consultar() {
120         int pos;
121
122         System.out.println("Qual posição do array deseja consultar? ");
123         pos = e.nextInt();
124
125         if(pos >= 0 && pos < array.size()){
126             Mae mm = (Mae) array.get(pos);
127             if(mm instanceof Filha01){
128                 Filha01 ff = (Filha01) array.get(pos);
129                 System.out.println("--[Dados Cadastrados]==");
130                 ff.imprimir();
131             }
132             else{
133                 System.out.println("Não existe instância de \"Filha01\" na " +
134                     "posição: " + pos);
135             }
136         }
137         else{
138             System.out.println("Não existe a posição: " + pos + ".");
139         }
140     }
141
142     public void relatorio() {
143         Filha01 ff;
144         Iterator it;
145         Mae mm;
146
147         System.out.println("\n--[Relatório]==\n");
148
149         // Usando o iterator para percorrer o ArrayList.
150         it = array.iterator();
151         while(it.hasNext()){
152             mm = (Mae)it.next();
153             if(mm instanceof Filha01){
154                 ff = (Filha01) mm;
155                 ff.imprimir();
156                 System.out.println("\n-----");
157             }
158         }
159     }
160 }
```

Figura 7: Classe "GerenciaFilha01" - Parte 03.

# Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
1 package exemplodepolimorfismo;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 /**
7  *
8  * @author Edwar Saliba Júnior
9  */
10 public class GerenciaFilha02 {
11     private ArrayList<Mae> array;
12     private Scanner e;
13
14     public GerenciaFilha02(ArrayList c) {
15         this.array = c;
16         e = new Scanner(System.in);
17     }
18
19     public void inserir(){
20         int a, b, c, d;
21
22         System.out.println("==[Digite os Valores]==");
23         System.out.println("Atributo1: ");
24         a = e.nextInt();
25         System.out.println("Atributo2: ");
26         b = e.nextInt();
27         System.out.println("Atributo3: ");
28         c = e.nextInt();
29         System.out.println("Atributo5: ");
30         d = e.nextInt();
31
32         array.add(new Filha02(a, b, c, d));
33     }
34
35     public void alterar(){
36         int a, b, c, d, pos;
37
38         System.out.println("Qual posição do array deseja alterar? ");
39         pos = e.nextInt();
40
41         /* O teste a seguir garante que a posição solicitada existe no
42            ArrayList. */
43         if(pos >= 0 && pos < array.size()){
44             /* Cria uma referência para o objeto na posição "pos" do array.
45                Lembre-se: mm tem apenas uma referência para o objeto que
46                está na posição "pos" do ArrayList, ou seja, mm não tirou
47                o objeto do ArrayList, apenas está o referenciando. Desta
48                forma, toda alteração feita em mm também será aplicada ao
49                objeto que está na posição "pos" do ArrayList, pois, ambos
50                são o mesmo objeto. Ou melhor, referenciam o mesmo objeto. */
51             Mae mm = (Mae) array.get(pos);
52             /* Testa se o objeto referenciado é uma instância da classe
53                "Filha02". */
54             if(mm instanceof Filha02){
55                 /* Faz a conversão do objeto "mm" do tipo "Mae" para o tipo
56                    "Filha02". */
57                 Filha02 ff = (Filha02) mm;
58             }
59         }
60     }
61 }
```

Figura 8: Classe "GerenciaFilha02" - Parte 01.



## Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
59         System.out.println("--[Dados Cadastrados]==");
60         ff.imprimir();
61
62         System.out.println("--[Digite os Novos Dados]==");
63         System.out.println("Atributo1: ");
64         a = e.nextInt();
65         System.out.println("Atributo2: ");
66         b = e.nextInt();
67         System.out.println("Atributo3: ");
68         c = e.nextInt();
69         System.out.println("Atributo5: ");
70         d = e.nextInt();
71
72         ff.alterar(a, b, c, d);
73     }
74     else{
75         System.out.println("Não existe instância de \"Filha02\" na " +
76             "posição: " + pos);
77     }
78 }
79 else{
80     System.out.println("Não existe a posição: " + pos + ".");
81 }
82 }
83
84 public void excluir(){
85     int pos, opcao;
86
87     System.out.println("Qual posição do array deseja excluir? ");
88     pos = e.nextInt();
89
90     if(pos >= 0 && pos < array.size()){
91         Mae mm = (Mae) array.get(pos);
92         if(mm instanceof Filha02){
93             Filha02 ff = (Filha02) array.get(pos);
94             System.out.println("--[Dados Cadastrados]==");
95             ff.imprimir();
96
97             do{
98                 System.out.println("Confirma exclusão? (1 - Sim / 2 - Não): ");
99                 opcao = e.nextInt();
100
101                 if(opcao == 1)
102                     array.remove(ff); // ou array.remove(pos);
103                 else
104                     if(opcao == 2)
105                         System.out.println("Exclusão não efetuada!");
106             }while((opcao != 2) && (opcao != 1));
107         }
108     else{
109         System.out.println("Não existe instância de \"Filha02\" na " +
110             "posição: " + pos);
111     }
112 }
113 else{
114     System.out.println("Não existe a posição: " + pos + ".");
115 }
116 }
```

Figura 9: Classe "GerenciaFilha02" - Parte 02.

## Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
117
118 public void consultar(){
119     int pos;
120
121     System.out.println("Qual posição do array deseja consultar? ");
122     pos = e.nextInt();
123
124     if(pos >= 0 && pos < array.size()){
125         Mae mm = (Mae) array.get(pos);
126         if(mm instanceof Filha02){
127             Filha02 ff = (Filha02) array.get(pos);
128             System.out.println("--[Dados Cadastrados]==");
129             ff.imprimir();
130         }
131         else{
132             System.out.println("Não existe instância de \"Filha02\" na " +
133                 "posição: " + pos);
134         }
135     }
136     else{
137         System.out.println("Não existe a posição: " + pos + ".");
138     }
139 }
140
141 public void relatorio(){
142     Filha02 ff;
143     System.out.println("\n--[Relatório]==\n");
144
145     // Usando o "for avançado" para percorrer o ArrayList.
146     for(Mae mm : array){
147         if(mm instanceof Filha02){
148             ff = (Filha02) mm;
149             ff.imprimir();
150             System.out.println("\n-----");
151         }
152     }
153 }
154 }
```

Figura 10: Classe "GerenciaFilha02" - Parte 03.

# Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
1 package exemplodepolimorfismo;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 /**
7  *
8  * @author Edwar Saliba Júnior
9  */
10 public class ExemploDePolimorfismo {
11
12     public static void main(String[] args) {
13         int opcaoMP, opcaoCar, opcaoCam;
14         Scanner e = new Scanner(System.in);
15
16         /* Polimorfismo. */
17         ArrayList<Mae> arrayPolimorfico = new ArrayList();
18
19         GerenciaFilha01 gerFilha01 = new GerenciaFilha01(arrayPolimorfico);
20         GerenciaFilha02 gerFilha02 = new GerenciaFilha02(arrayPolimorfico);
21
22         do{
23             System.out.println("\n\n");
24             System.out.println("-----[Menu Principal]-----");
25             System.out.println("Escolha um Opção");
26             System.out.println("1 - Filha01");
27             System.out.println("2 - Filha02");
28             System.out.println("3 - Sair");
29             System.out.println("\nOpção: ");
30             opcaoMP = e.nextInt();
31             e.skip("\n");
32
33             switch(opcaoMP) {
34                 case 1:
35                     do{
36                         System.out.println("\n---[Menu de Filha01]---");
37                         System.out.println("Escolha um Opção");
38                         System.out.println("1 - Inclusão");
39                         System.out.println("2 - Alteração");
40                         System.out.println("3 - Exclusão");
41                         System.out.println("4 - Consulta");
42                         System.out.println("5 - Relatório");
43                         System.out.println("6 - Voltar ao Menu Principal");
44                         System.out.println("\nOpção: ");
45                         opcaoCar = e.nextInt();
46                         e.skip("\n");
47
48                         switch(opcaoCar) {
49                             case 1:
50                                 gerFilha01.inserir();
51                                 break;
52                             case 2:
53                                 gerFilha01.alterar();
54                                 break;
55                             case 3:
56                                 gerFilha01.excluir();
57                                 break;
58                             case 4:
```

Figura 11: Classe "ExemploDePolimorfismo" - Parte01

## Polimorfismo em Java

Prof. Edwar Saliba Júnior

```
59         gerFilha01.consultar();
60         break;
61     case 5:
62         gerFilha01.relatorio();
63     }
64     }while(opcaoCar != 6);
65     break;
66 case 2:
67     do{
68         System.out.println("\n---=[Menu de Filha02]---");
69         System.out.println("Escolha um Opção");
70         System.out.println("1 - Inclusão");
71         System.out.println("2 - Alteração");
72         System.out.println("3 - Exclusão");
73         System.out.println("4 - Consulta");
74         System.out.println("5 - Relatório");
75         System.out.println("6 - Voltar ao Menu Principal");
76         System.out.println("\nOpção: ");
77         opcaoCam = e.nextInt();
78         e.skip("\n");
79
80         switch(opcaoCam){
81             case 1:
82                 gerFilha02.inserir();
83                 break;
84             case 2:
85                 gerFilha02.alterar();
86                 break;
87             case 3:
88                 gerFilha02.excluir();
89                 break;
90             case 4:
91                 gerFilha02.consultar();
92                 break;
93             case 5:
94                 gerFilha02.relatorio();
95             }
96         }while(opcaoCam != 6);
97     }
98
99     }while(opcaoMP != 3);
100 }
101 }
```

Figura 12: Classe "ExemploDePolimorfismo" - Parte 02.