

## Súatório

Este material apresenta um exemplo prático da aplicação de diversos conceitos da programação orientada a objetos na linguagem Java. Em adição, diferentes aspectos próprios à linguagem Java (como tratamento de exceções, manipulação de *streams* e classe *Object*) serão brevemente introduzidos. O programa abaixo encontra-se disponível para *download* na página da disciplina. Ao final deste material de apoio é também apresentado um exemplo de entrada e saída resultante da execução deste programa.

Para aproveitar o conteúdo, o aluno deve:

- Acompanhar na aula a discussão, linha a linha, do código fonte;
- Recuperar o arquivo fonte, compila-lo, e executá-lo, compreendendo tanto o algoritmo empregado como o emprego dos conceitos da orientação a objetos.

Na sua auto avaliação sobre a matéria, o aluno deve verificar se é capaz de:

- Identificar os conceitos de orientação a objetos no programa e seus respectivos usos;
- Identificar recursos próprios de Java e dominar seu uso;
- Utilizar algum ambiente de programação em Java para desenvolver programas;
- Estender o exemplo de código apresentado com outras funcionalidades.

1.	<code>import java.io.*;</code>
2.	
3.	<code>abstract class Lista {</code>
4.	<code>    protected class Elem {</code>
5.	<code>        private Object ob;</code>
6.	<code>        private Elem next;</code>
7.	
8.	<code>        public Elem( Object ob, Elem next ) {</code>
9.	<code>            this.ob = ob;</code>
10.	<code>            this.next = next;</code>
11.	<code>        }</code>
12.	<code>        public Object getElem() {</code>
13.	<code>            return ob;</code>
14.	<code>        }</code>
15.	<code>        public Elem getNext() {</code>
16.	<code>            return next;</code>
17.	<code>        }</code>
18.	<code>        public void setNext( Elem proximo ) {</code>
19.	<code>            next = proximo;</code>
20.	<code>        }</code>
21.	<code>    } //Elem</code>
22.	<code>    protected Elem prim;</code>
23.	<code>    protected int nbelem;</code>
24.	
25.	<code>    public Lista() {</code>
26.	<code>        nbelem = 0;</code>
27.	<code>        prim = null;</code>
28.	<code>    }</code>
29.	<code>    public int length() {</code>
30.	<code>        return nbelem;</code>
31.	<code>    }</code>
32.	<code>    abstract void insere( Object ob );</code>
33.	<code>    abstract Object retira();</code>
34.	<code>}</code>
35.	
36.	<code>class Fila extends Lista {</code>
37.	<code>    public Fila() {</code>
38.	<code>        super();</code>
39.	<code>    }</code>
40.	<code>    public void insere( Object ob ) {</code>
41.	<code>        nbelem++;</code>
42.	<code>        Elem aux = new Elem( ob, prim );</code>
43.	<code>        prim = aux;</code>
44.	<code>    }</code>
45.	<code>    public Object retira() {</code>
46.	<code>        Object aux = null;</code>
47.	
48.	<code>        if( nbelem == 1 ) {</code>
49.	<code>            nbelem = 0;</code>
50.	<code>            aux = prim.getElem();</code>
51.	<code>            prim = null;</code>

```
52.     }
53.     else if( nbelem > 1 ) {
54.         Elem penultimo = prim,
55.             ultimo = prim.getNext();
56.         for( ; ultimo.getNext() != null ; ultimo = ultimo.getNext() )
57.             penultimo = ultimo;
58.         aux = ultimo.getElem();
59.         penultimo.setNext(null);
60.         nbelem--;
61.     }
62.     return aux;
63. }
64. }
65.
66. class Pilha extends Lista {
67.     public Pilha() {
68.         super();
69.     }
70.     public void insere(Object ob) {
71.         nbelem++;
72.         Elem aux = prim;
73.         prim = new Elem( ob, aux );
74.     }
75.     public Object retira() {
76.         Object aux = null;
77.
78.         if( nbelem > 0 ) {
79.             aux = prim.getElem();
80.             nbelem--;
81.             prim = prim.getNext();
82.         }
83.         return aux;
84.     }
85.     public void push( Object ob ) {
86.         this.insere(ob);
87.     }
88.     public Object pop() {
89.         return this.retira();
90.     }
91. }
92.
93. class ManipulaLista {
94.     private Lista l;
95.
96.     public ManipulaLista() {
97.         l = new Fila();
98.     }
99.     public ManipulaLista( Lista l ) {
100.        this.l = l;
101.    }
102.    public void encheLista() {
103.        String linha;
104.        InputStreamReader teclado = new InputStreamReader( System.in );
105.        BufferedReader console = new BufferedReader(teclado);
106.
107.        for( int i = 0 ; i < 10 ; i++ ) {
108.            try {
109.                System.out.print("Entre texto " + (i+1) + " de 10: " );
110.                linha = console.readLine();
111.            }
112.            catch ( Exception e ) {
113.                System.out.println("Nao conseguiu ler...");
114.                return;
115.            }
116.            //l.insere(new String(linha));
117.            l.insere( linha );
118.        }
119.    }
120.    public void esvaziaLista() {
121.        for( int i = l.length() ; i > 0 ; i-- )
```

122.	System.out.println( "Retirado: " + l.retira() );
123.	}
124.	}
125.	
126.	public class UsaLista {
127.	public static void main (String args[]) {
128.	ManipulaLista ml;
129.	
130.	System.out.println("Manipulando uma Fila");
131.	Fila f = new Fila();
132.	ml = new ManipulaLista( f );
133.	ml.encheLista();
134.	ml.esvaziaLista();
135.	
136.	System.out.println("Manipulando uma Pilha");
137.	Pilha p = new Pilha();
138.	ml= new ManipulaLista( p );
139.	ml.encheLista();
140.	ml.esvaziaLista();
141.	}
142.	}

### Exemplo de uso: compilação e execução. Ambiente GNU-Linux, JDK 1.4.2.10

```
gersonc@anahy ~/fnt/Java $ javac UsaLista.java
gersonc@anahy ~/fnt/Java $ java UsaLista
Manipulando uma Fila
Entre texto 1 de 10: um
Entre texto 2 de 10: dois
Entre texto 3 de 10: tres
Entre texto 4 de 10: quatro
Entre texto 5 de 10: cinco
Entre texto 6 de 10: seis
Entre texto 7 de 10: sete
Entre texto 8 de 10: oito
Entre texto 9 de 10: nove
Entre texto 10 de 10: dez
Retirado: um
Retirado: dois
Retirado: tres
Retirado: quatro
Retirado: cinco
Retirado: seis
Retirado: sete
Retirado: oito
Retirado: nove
Retirado: dez
Manipulando uma Pilha
Entre texto 1 de 10: um
Entre texto 2 de 10: dois
Entre texto 3 de 10: tres
Entre texto 4 de 10: quatro
Entre texto 5 de 10: cinco
Entre texto 6 de 10: seis
Entre texto 7 de 10: sete
Entre texto 8 de 10: oito
Entre texto 9 de 10: nove
Entre texto 10 de 10: dez
Retirado: dez
Retirado: nove
Retirado: oito
Retirado: sete
Retirado: seis
Retirado: cinco
Retirado: quatro
Retirado: tres
Retirado: dois
Retirado: um
gersonc@anahy ~/fnt/Java $
```