

APIs Java: Arquivos

POO29004 – Engenharia de Telecomunicações

Prof. Emerson Ribeiro de Mello

<http://docente.ifsc.edu.br/mello/poo>

10 DE JUNHO DE 2020



**INSTITUTO
FEDERAL**
Santa Catarina

Câmpus
São José

Arquivo texto: Java I/O API

Escrevendo em um arquivo texto

```
1 public static void main(String args[]){
2     File arquivo;
3     try{
4         arquivo = new File("arquivo.txt");
5         FileWriter fwArquivo = null;
6
7         if (arquivo.exists() == true) {// Abre para adicionar dados
8             fwArquivo = new FileWriter(arquivo, true);
9         }else{ // se não existir, então cria o arquivo
10             fwArquivo = new FileWriter(arquivo);
11         }
12         BufferedWriter bw = new BufferedWriter(fwArquivo);
13         bw.write("Olá mundo\n");
14
15         // fechando arquivo
16         bw.close();
17         fwArquivo.close();
18     }catch(Exception e){...}
19 }
```



Lendo dados de um arquivo texto

```
20 public static void main(String args[]){  
21     File arquivo;  
22     try{  
23         arquivo = new File("arquivo.txt");  
24         Scanner leitor = new Scanner(arquivo);  
25  
26         //varrendo o arquivo linha por linha até chegar no final do mesmo  
27         while (leitor.hasNextLine()) {  
28             System.out.println(leitor.nextLine());  
29         }  
30  
31         leitor.close();  
32     }catch(Exception e){...}  
33 }
```



Lendo dados de um arquivo texto

```
34 public static void main(String args[]){
35     File arquivo;
36     try{
37         arquivo = new File("arquivo.txt");
38         Scanner leitor = new Scanner(arquivo);
39
40         //varrendo o arquivo linha por linha até chegar no final do mesmo
41         while (leitor.hasNextLine()) {
42             System.out.println(leitor.nextLine());
43         }
44
45         leitor.close();
46     }catch(Exception e){...}
47 }
```

■ Código completo no *link* <https://gist.github.com/emersonmello>



- Trata-se uma instrução `try` onde se declara um ou mais recursos
 - **Recurso** – objeto que precisa ser fechado antes do encerramento do programa. Exemplo: quando se trabalha com arquivos em disco
 - Qualquer objeto que implementar `java.lang.AutoCloseable` pode ser usada como um recurso
- Garante que cada recurso será fechado após a execução do `try`
- É possível ter blocos `catch` e `finally` como qualquer instrução `try`



Try-with-resources – Exemplo: fechando recursos

```
48 try{
49     fwArquivo = new FileWriter(new File("arquivo.txt"));
50     BufferedWriter bw = new BufferedWriter(fwArquivo);
51     .....
52 }catch(Exception e){
53     ...
54 }finally{
55     bw.close();
56     fwArquivo.close();
57 }
```

```
58 try(fwArquivo = new FileWriter(new File("arquivo.txt"));
59     BufferedWriter bw = new BufferedWriter(fwArquivo)) {
60     .....
61 }catch(Exception e){
62     ...
63 }
```



Objetos para arquivos


```
64 import java.io.Serializable;
65
66 public class Pessoa implements Serializable{
67     private String nome;
68     private String cpf;
69
70     public Pessoa(String nome, String cpf) {
71         this.nome = nome;
72         this.cpf = cpf;
73     }
74     public void imprimir(){
75         System.out.println("Nome: " + nome + "\n" + "CPF: " + cpf);
76     }
```

- A interface `java.io.Serializable` permite que objetos sejam serializados, ou seja, permite que esses objetos sejam gravados em arquivos, transmitidos pela rede, etc.



Classe Principal.java

```
77 public class Principal {
78     private Pessoa[] agenda = new Pessoa[2];
79
80     public void salvarEmDisco(){
81         //veja o codigo na proxima lamina
82     }
83     public void lerDoDisco(){
84         //veja o codigo daqui 2 laminas
85     }
86     public static void main(String[] args) {
87         Principal p = new Principal();
88
89         p.agenda[0] = new Pessoa("joao", "123");
90         p.agenda[1] = new Pessoa("maria", "456");
91
92         p.salvarEmDisco();
93         p.lerDoDisco();
94     }
95 } //fim da classe
```



Trecho de código para gravar objeto em arquivo

```
96 public void salvarEmDisco(){
97     File arquivo = new File("agenda.dat");
98     try {
99
100         FileOutputStream fout = new FileOutputStream(arquivo);
101         ObjectOutputStream oos = new ObjectOutputStream(fout);
102
103         // gravando o vetor 'agenda' no arquivo chamado 'agenda.dat'
104         oos.writeObject(this.agenda);
105
106         oos.flush(); // limpando dados em buffer
107         oos.close(); // fechando fluxo de saída
108         fout.close(); // fechando arquivo
109
110     } catch (Exception ex) {
111         System.err.println("erro: " + ex.toString());
112     }
113 }
```



Trecho de código para ler objeto de um arquivo

```
114 public void lerDoDisco(){
115     File arquivo = new File("agenda.dat");
116     try {
117         FileInputStream fin = new FileInputStream(arquivo);
118         ObjectInputStream oin = new ObjectInputStream(fin);
119
120         // Lendo os objetos de um arquivo
121         this.agenda = (Pessoa[]) oin.readObject();
122
123         oin.close(); //fechando fluxo de entrada
124         fin.close(); //fechando arquivo
125
126         // Uma forma de diferente do for para percorrer vetores
127         for (Pessoa p : this.agenda) {
128             p.imprimir();
129         }
130     } catch (Exception ex) {
131         System.err.println("erro: " + ex.toString());
132     }
133 }
```



Exercícios

- Melhore a Lista de Exercícios 05 de forma que agenda telefônica possa ser persistida em disco



- API JAVA 7: NIO.2

<https://docs.oracle.com/javase/tutorial/essential/io>

