

```
1 package classes;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.IOException;
6 import java.util.ArrayList;
7 import java.util.List;
8
9 /**
10  * created by lgcaobianco on 2018-04-14
11  */
12
13 public class LeitorPontosEntrada {
14
15     private String nomeArquivo;
16     private String formato;
17     private String separadorValor;
18
19     private String getNomeArquivo() {
20         return nomeArquivo;
21     }
22
23     private String getFormato() {
24         return formato;
25     }
26
27     private String getSeparadorValor() {
28         return separadorValor;
29     }
30
31     public LeitorPontosEntrada(String nomeArquivo, String
formato) {
32         this.nomeArquivo = nomeArquivo;
33         this.formato = formato;
34         switch (formato) {
35             case ".csv":
36                 this.separadorValor = ",";
37                 break;
38             case ".txt":
39                 this.separadorValor = " ";
40                 break;
41             default:
42                 System.out.println("Formato ainda não
suportado");
43                 System.exit(1);
44                 break;
45         }
46     }
47 }
48
```

```
49
50
51     public List<Double[]> extrairPontos() {
52         List<Double[]> matrizPontos = new ArrayList<Double
53         []>();
54         String linhaLida = "";
55         BufferedReader stream = null;
56         try {
57             stream = new BufferedReader(new FileReader(
58             getNomeArquivo() + getFormato()));
59             while ((linhaLida = stream.readLine()) != null
60             ) {
61                 String[] temporario = linhaLida.split(
62                 getSeparadorValor());
63                 Double[] numerosSeparados = new Double[
64                 temporario.length];
65                 for (int i = 0; i < temporario.length; i++
66                 ) {
67                     numerosSeparados[i] = Double.
68                     parseDouble(temporario[i]);
69                 }
70                 matrizPontos.add(numerosSeparados);
71             }
72         } catch (IOException e) {
73             e.printStackTrace();
74             System.exit(1);
75         } finally {
76             if (stream != null) {
77                 try {
78                     stream.close();
79                 } catch (IOException e) {
80                     e.printStackTrace();
81                     System.exit(1);
82                 }
83             }
84         }
85     }
86     return matrizPontos;
87 }
```