1. **package** com.readfile;
3. **import** java.io.BufferedReader;
4. **import** java.io.File;
5. **import** java.io.FileInputStream;
6. **import** java.io.FileReader;
7. **import** java.io.IOException;
8. **import** java.io.InputStream;
9. **import** java.io.InputStreamReader;
10. **import** java.io.RandomAccessFile;
11. **import** java.io.Reader;
13. **public** **class** ReadFromFile {
15. /\*\*
16. \* @param args
17. \*/
18. **public** **static** **void** main(String[] args) {
19. // TODO Auto-generated method stub
20. String fileName = "C:/Users/Administrator/Desktop/Noname1.txt";
21. //readFileByBytes(fileName);
22. //readFileByChars(fileName);
23. //readFileByLines(fileName);
24. readFileByRandomAccess(fileName);
25. }
27. /\*\*
28. \* 随机读取文件内容
29. \*/
30. **public** **static** **void** readFileByRandomAccess(String fileName) {
31. RandomAccessFile randomFile = **null**;
32. **try** {
33. System.out.println("随机读取一段文件内容：");
34. // 打开一个随机访问文件流，按只读方式
35. randomFile = **new** RandomAccessFile(fileName, "r");
36. // 文件长度，字节数
37. **long** fileLength = randomFile.length();
38. // 读文件的起始位置
39. **int** beginIndex = (fileLength > 4) ? 0 : 0;
40. // 将读文件的开始位置移到beginIndex位置。
41. randomFile.seek(beginIndex);
42. **byte**[] bytes = **new** **byte**[10];
43. **int** byteread = 0;
44. // 一次读10个字节，如果文件内容不足10个字节，则读剩下的字节。
45. // 将一次读取的字节数赋给byteread
46. **while** ((byteread = randomFile.read(bytes)) != -1) {
47. System.out.write(bytes, 0, byteread);
48. }
49. } **catch** (IOException e) {
50. e.printStackTrace();
51. } **finally** {
52. **if** (randomFile != **null**) {
53. **try** {
54. randomFile.close();
55. } **catch** (IOException e1) {
56. }
57. }
58. }
59. }
60. /\*\*
61. \* 以行为单位读取文件，常用于读面向行的格式化文件
62. \*/
63. **public** **static** **void** readFileByLines(String fileName) {
64. File file = **new** File(fileName);
65. BufferedReader reader = **null**;
66. **try** {
67. System.out.println("以行为单位读取文件内容，一次读一整行：");
68. reader = **new** BufferedReader(**new** FileReader(file));
69. String tempString = **null**;
70. **int** line = 1;
71. // 一次读入一行，直到读入null为文件结束
72. **while** ((tempString = reader.readLine()) != **null**) {
73. // 显示行号
74. System.out.println("line " + line + ": " + tempString);
75. line++;
76. }
77. reader.close();
78. } **catch** (IOException e) {
79. e.printStackTrace();
80. } **finally** {
81. **if** (reader != **null**) {
82. **try** {
83. reader.close();
84. } **catch** (IOException e1) {
85. }
86. }
87. }
88. }
90. /\*\*
91. \* 以字符为单位读取文件，常用于读文本，数字等类型的文件
92. \*/
93. **public** **static** **void** readFileByChars(String fileName) {
94. File file = **new** File(fileName);
95. Reader reader = **null**;
96. **try** {
97. System.out.println("以字符为单位读取文件内容，一次读一个字节：");
98. // 一次读一个字符
99. reader = **new** InputStreamReader(**new** FileInputStream(file));
100. **int** tempchar;
101. **while** ((tempchar = reader.read()) != -1) {
102. // 对于windows下，\r\n这两个字符在一起时，表示一个换行。
103. // 但如果这两个字符分开显示时，会换两次行。
104. // 因此，屏蔽掉\r，或者屏蔽\n。否则，将会多出很多空行。
105. **if** (((**char**) tempchar) != '\r') {
106. System.out.print((**char**) tempchar);
107. }
108. }
109. reader.close();
110. } **catch** (Exception e) {
111. e.printStackTrace();
112. }
113. **try** {
114. System.out.println("\n以字符为单位读取文件内容，一次读多个字节：");
115. // 一次读多个字符
116. **char**[] tempchars = **new** **char**[30];
117. **int** charread = 0;
118. reader = **new** InputStreamReader(**new** FileInputStream(fileName));
119. // 读入多个字符到字符数组中，charread为一次读取字符数
120. **while** ((charread = reader.read(tempchars)) != -1) {
121. // 同样屏蔽掉\r不显示
122. **if** ((charread == tempchars.length)
123. && (tempchars[tempchars.length - 1] != '\r')) {
124. System.out.print(tempchars);
125. } **else** {
126. **for** (**int** i = 0; i < charread; i++) {
127. **if** (tempchars[i] == '\r') {
128. **continue**;
129. } **else** {
130. System.out.print(tempchars[i]);
131. }
132. }
133. }
134. }
136. } **catch** (Exception e1) {
137. e1.printStackTrace();
138. } **finally** {
139. **if** (reader != **null**) {
140. **try** {
141. reader.close();
142. } **catch** (IOException e1) {
143. }
144. }
145. }
146. }
147. /\*\*
148. \* 以字节为单位读取文件，常用于读二进制文件，如图片、声音、影像等文件。
149. \*/
150. **public** **static** **void** readFileByBytes(String fileName) {
151. File file = **new** File(fileName);
152. InputStream in = **null**;
153. **try** {
154. System.out.println("以字节为单位读取文件内容，一次读一个字节：");
155. // 一次读一个字节
156. in = **new** FileInputStream(file);
157. **int** tempbyte;
158. **while** ((tempbyte = in.read())!=-1) {
159. System.out.println(tempbyte);
160. }
161. } **catch** (Exception e) {
162. // TODO: handle exception
163. e.printStackTrace();
164. }
166. **try** {
167. System.out.println("以字节为单位读取文件内容，一次读多个字节：");
168. // 一次读多个字节
169. **byte**[] tempbytes = **new** **byte**[100];
170. **int** byteread = 0;
171. in = **new** FileInputStream(fileName);
172. ReadFromFile.showAvailableBytes(in);
173. // 读入多个字节到字节数组中，byteread为一次读入的字节数
174. **while** ((byteread = in.read(tempbytes)) != -1) {
175. System.out.write(tempbytes, 0, byteread);//好方法，第一个参数是数组，第二个参数是开始位置，第三个参数是长度
176. }
177. } **catch** (Exception e1) {
178. e1.printStackTrace();
179. } **finally** {
180. **if** (in != **null**) {
181. **try** {
182. in.close();
183. } **catch** (IOException e1) {
184. }
185. }
186. }
187. }
189. /\*\*
190. \* 显示输入流中还剩的字节数
191. \*/
192. **private** **static** **void** showAvailableBytes(InputStream in) {
193. **try** {
194. System.out.println("当前字节输入流中的字节数为:" + in.available());
195. } **catch** (IOException e) {
196. e.printStackTrace();
197. }
198. }
200. }