

## ColorConverter.java

```

1 package project1_jk;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.util.NoSuchElementException;
6 import java.util.Scanner;
7
8
9
10 /**
11  * This class is the program performing color conversion.
12  * The program is interactive.
13  * When the program is executed the name of the input file containing the list of all the named
14  * CSS colors is provided as the program's single command line argument. The data in this file
15  * serves as a database of all the named colors.
16  * In the interactive part, the user enters a hexadecimal representation of a color. The program
17  * responds by printing the RGB description and the color name (if one exists in the list of
18  * named colors).
19  *
20  * @author Joanna Klukowska
21  *
22  */
23 public class ColorConverter {
24
25     /**
26     * The main() method of this program.
27     * @param args array of Strings provided on the command line when the program is started;
28     * the first string should be the name of the input file containing the list of named colors.
29     */
30     public static void main(String[] args) {
31
32         //verify that the command line argument exists
33         if (args.length == 0 ) {
34             System.err.println("Usage Error: the program expects file name as an argument.\n");
35             System.exit(1);
36         }
37
38         //verify that command line argument contains a name of an existing file
39         File colorFile = new File(args[0]);
40         if (!colorFile.exists()){
41             System.err.println("Error: the file "+colorFile.getAbsolutePath()+" does not exist.\n");
42             System.exit(1);
43         }
44         if (!colorFile.canRead()){
45             System.err.println("Error: the file "+colorFile.getAbsolutePath()+
46                 " cannot be opened for reading.\n");
47             System.exit(1);
48         }
49
50         //open the file for reading
51         Scanner inColors = null;
52
53
54         try {
55             inColors = new Scanner (colorFile ) ;
56         } catch (FileNotFoundException e) {
57             System.err.println("Error: the file "+colorFile.getAbsolutePath()+
58                 " cannot be opened for reading.\n");
59             System.exit(1);
60         }
61
62         //read the content of the file and save the data in a list of named colors
63         ColorList list = new ColorList();
64         String line = null;

```

## ColorConverter.java

```

65 Scanner parseLine = null;
66 String colorName = null;
67 String hexValue = null;
68 Color current = null;
69 while (inColors.hasNextLine()) {
70     try {
71         line = inColors.nextLine();
72         parseLine = new Scanner(line);
73         parseLine.useDelimiter(", ");
74         colorName = parseLine.next();
75         hexValue = parseLine.next();
76     }
77     catch (NoSuchElementException ex ) {
78         //caused by an incomplete or miss-formatted line in the input file
79         System.err.println(line);
80         continue;
81     }
82     try {
83         current = new Color (hexValue.trim(), colorName.trim());
84         list.add( current );
85     }
86     catch (IllegalArgumentException ex ) {
87         //ignore this exception and skip to the next line
88     }
89 }
90
91 //interactive mode:
92
93 Scanner userInput = new Scanner (System.in );
94 String userValue = "";
95
96
97
98 do {
99     System.out.println("Enter the color in HEX format (#RRGGBB) or \"quit\" to stop:");
100    //get value of from the user
101    userValue = userInput.nextLine();
102    if (!userValue.equalsIgnoreCase("quit")) {
103        Color c = list.getColorByHexValue( userValue );
104        if ( c == null ) {
105            try {
106                c = new Color (userValue);
107            }
108            catch (IllegalArgumentException ex ) {
109                System.out.println("Error: This is not a valid color specification.");
110                continue;
111            }
112        }
113        System.out.println(c + "\n");
114    }
115 } while (!userValue.equalsIgnoreCase("quit"));
116
117 userInput.close();
118
119 }
120
121 }
122 }
123

```