Home / My courses / Previous Term (Term 3, 2015-2016) / CS 1103 - T3 2015-2016 / 10 March - 16 March /
Solutions for Exercises Unit 6

Solutions for Exercises Unit 6

This is a non-grade exercise which should be posted to your learning journal.

The sample program *DirectoryList.java* (in the code directory), given as an example in section 11.2.2 of the textbook will print a list of files in a directory specified by the user. But some of the files in that directory might themselves be directories. And the subdirectories can themselves contain directories. And so on. Write a modified version of DirectoryList that will list all the files in a directory and all its subdirectories, to any level of nesting. You will need a **recursive** subroutine to do the listing. The subroutine should have a parameter of type File. You will need the constructor from the *File* class that has the form

```
public File( File dir, String fileName )
// Constructs the File object representing a file
// named fileName in the directory specified by dir.
```

Discussion

One possible recursive algorithm for printing all the files in a directory and its subdirectories is:

```
Print the name of the directory
for each file in the directory:
   if the file is a directory:
      Print its contents recursively
   else
      Print the name of the file.
```

If the directory is given as an object of type *File*, then this can be coded easily. We need some instance methods from the *File* class. (This subroutine is not quite what I ended up using in my solution.)

```
static void listDirectoryContents( File dir ) {
    System.out.println("Directory \"" + dir.getName() + "\"");
    String[] files;  // The names of the files in the directory.
    files = dir.list();
    for (int i = 0; i < files.length; i++) {
        File f;  // One of the files in the directory.
        f = new File(dir, files[i]);
        if ( f.isDirectory() ) {
               // Call listContents() recursively to
               // list the contents of the directory, f.
            listContents(f);
        }
        else {
              // For a regular file, just print the name, files[i].
            System.out.println(files[i]);
        }
    }
} // end listDirectoryContents()
```

Every time this routine finds a directory, it lists not just the name of the directory but also, recursively, the names of everything that it contains. The only problem with this is that it doesn't indicate which items are in which directory. For example, consider the output:

```
Directory "games"
blackbox
Directory "CardGames"
cribbage
euchre
tetris
```

There is no way to tell where the list of items in "CardGames" ends. Possibly, for example, "euchre" is in the "CardGames" directory. But possibly, "cribbage" is the only file in "CardGames" and "euchre" is actually part of the listing for "games". It would be nice to use indentation to show the nesting, like this:

```
Directory "games"
   blackbox
   Directory "CardGames"
      cribbage
      euchre
   tetris
```

In this listing, you can tell that "euchre" is in fact in "CardGames" while "tetris" is in "games". To implement this, we just have to recognize that the indentation can be different in each call to the listDirectoryContents() method. Since it can be different, it should be a parameter. When we call listDirectoryContents() recursively, we should increase the indentation. You can see how this is done in the actual solution, given below.

The Solution

```
import java.io.File;
import java.util.Scanner;

/**
 * This program lists the files in a directory specified by
 * the user.  If one of those files is itself a directory, the
 * program recursively lists the contents of that subdirectory.
 * The user is asked to type in the name of the directory that is
 * to be listed.  If the name entered by the user is not a directory,
 * a message is printed and the program ends.
 */
public class RecursiveDirectoryList {


   public static void main(String[] args) {

      String directoryName;  // Directory name entered by the user.
      File directory;        // File object referring to the directory.
      Scanner scanner;       // For reading a line of input from the user.

      scanner = new Scanner(System.in);  // scanner reads from
                                         // standard input.

      System.out.print("Enter a directory name: ");
      directoryName = scanner.nextLine().trim();
      directory = new File(directoryName);

      if (directory.isDirectory() == false) {
         if (directory.exists() == false)
            System.out.println("There is no such directory!");
         else
            System.out.println("That file is not a directory.");
      }
      else {
            // List the contents of the directory recursively, with
            // no indentation at the top level.
         listDirectoryContents(directory,"");
      }

   } // end main()


   /**
    * A recursive subroutine that lists the contents of a directory,
    *   including the contents of its subdirectories to any level of nesting.
    * @param dir the directory whose contents are to be listed.
    *   It is assumed that dir is in fact a directory.
    * @param indent a string of blanks that is prepended to each item in
    *   the directory listing.  This string grows in length
    *   with each increase in the level of directory nesting.
    */
   private static void listDirectoryContents(File dir, String indent) {
      String[] files;  // List of names of files in the directory.
      System.out.println(indent + "Directory \"" + dir.getName() + "\":");
      indent += "   ";  // Increase the indentation for listing the contents.
      files = dir.list();
      for (int i = 0; i < files.length; i++) {
            // If the file is a  directory, list its contents
            // recursively.  Otherwise, just print its name.
         File f = new File(dir, files[i]);
         if (f.isDirectory())
            listDirectoryContents(f, indent);
         else
```

```
        System.out.println(indent + files[i]);
    }
  } // end listContents()


} // end class RecursiveDirectoryList
```

Last modified: Thursday, 30 July 2015, 1:22 AM

Navigation

Home

- My home
  - Site pages
  - My profile
  - Current course
    - CS 1103 - T3 2015-2016
      - Participants
      - Badges
      - 28 January - 3 February
      - 4 February - 10 February
      - 11 February - 17 February
      - 18 February - 24 February
      - 25 February - 2 March
      - 3 March - 9 March
      - 10 March - 16 March
        - Learning Guide Unit 7
        - Discussion Forum Unit 7
          - Assignment Unit 7
        - Learning Journal Unit 7
        - Self-Quiz Unit 7
        - Code Unit 7
        - Lab 11 Unit 7
        - Solutions for Assignment Unit 6
        - **Solutions for Exercises Unit 6**
      - 17 March - 23 March
      - 24 March - 30 March

Administration

University of the People is a 501(c)(3) not for profit organization. Contributions are tax deductible to the extent permitted by law.