Name: E7_David_Mugwaneza

Composite Design Pattern

Problem:
The objective of this exercise is to implement the Composite design pattern.

Steps:

1. Create an interface called **AbstractFile**. This is the highest level interface of all component. Define an empty method void **ls()** in it. It is going to be used to print out the entire content in a directory.

2. Create a class called **File** to implements **AbstractFile**. This the leaf node (or menu item in slides). The File class should have its constructor, a String instance variable called **name**.

3. You need to implement the **ls()** method by printing out the file name.

   ```
   System.out.println("File: " + name);
   ```

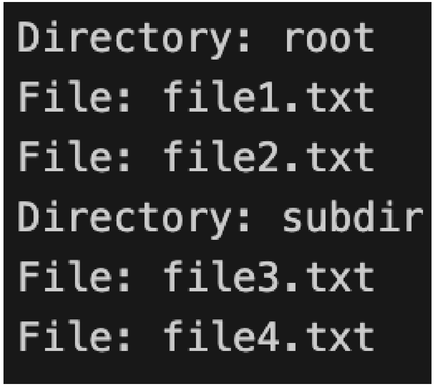
4. Create a class called **Directory** to implements **AbstractFile**. This is the composite class. It should have its constructor and a String instance variable called **name**.

5. Different from File class, we need to add an ArrayList called **includedFiles** to save all the files and sub-directories in it. Also, we need a **void add(AbstractFile file)** method to add files and directories to it. Implement this **add** method by adding the file to the ArrayList **includedFiles.**

6. The **ls()** method in **Directory** class is different from the one in **File**. In addition to printing out the directory name, you need to add a for loop to traverse every item in the ArrayList **includedFiles,** and call its **ls()** method.

7. Write a client class and a main function to demonstrate how they work. Add some files and sub-directories to the root directory, and add some files to the sub-directories. Then, print all of them by calling the **ls()** method in the root directory.

Output:



```
Directory: root
File: file1.txt
File: file2.txt
Directory: subdir
File: file3.txt
File: file4.txt
```

What to turn in: Similar as E5.