CPSC 3333 Programming Assignment 5

Overview

It is important that you name the required classes and methods exactly the same (case included) as specified. Testing will fail if you do not.

This week we will write a simulation of the directory structure.

Deliverables

You must submit the following file:

Directory.java

Specifications

For this assignment you can use any library provided by java (e.g. ArrayList, LinkedList, etc.) I will give you the freedom to select what you think works best. You may also have as many private variables and private methods as you see fit. You are only allowed to have the public methods specified below. You are not allowed to have any public variables.

- Directory.java
  This will simulate a directory structure in an OS. For this assignment you will only need to create the directory structure. There will be no files in these directories. You will be responsible for a "tree" based directory structure. Therefore, each directory tree should start with a directory called "root". After that, any number of directories can be added to *root*. *Root* may have any number of children, and those children can also have any number of children, and so on (there is no limit to the depth of this tree).

  For this assignment you are not required to traverse the tree or find any specific directory in the tree. However, each child should have any number of children.

  The constructor will take the one parameter. This parameter is the name of the directory. E.g. The first Directory should be "root".

  This class must have the following public method:

  Directory addChild(String)

  This will add a child directory to the current directory.

  Example code could look like the following:

  Directory root = new Directory("root")

  Directory user = root.addChild("user")

  user.addChild("user1")

  user.addChild("user2")

  Directory apps = root.addChild("apps")

  Directory chrome = apps.addChild("chrome")

  Directory word = apps.addChild("word")

  At this point the directory root would have two children (user and apps). Root/user would have two children (user1 and user2). Root/apps would also have two children (chrome and word).

CPSC 3333 Programming Assignment 5