CSE222 HW4

Core Functions

Directory FileSystem.changeDirectory(String path)

The method takes a `path` as an argument, which is a string representing the path of the directory to navigate to. The path is split into individual directory names using the `/` character as a delimiter.

If the first directory in the path is "root", the method prints a message indicating that it's changing to the root directory and returns the root directory.

If the path does not start with "root", the method starts from the root directory and iterates over the directory names in the path.

For each directory name, it checks if the name is not empty. If it is not, it prints a message indicating that it's searching for the directory.

It then retrieves the children of the current directory (which are `FileSystemElement` objects) and checks each child to see if it's a `Directory` and if its name matches the current directory name in the path.

If it finds a match, it sets the current directory to the found directory, prints a message indicating that it found the directory, and breaks out of the loop.

If it does not find a match after checking all the children, it prints a message indicating that the directory was not found and returns the current directory.

After iterating over all the directory names in the path, it returns the current directory, which is the directory that the path points to.

void moveElement(String name, String newPath)

The method takes two arguments: name, which is the name of the file or directory to move, and newPath, which is the path of the directory to move the file or directory to.

It first retrieves the FileSystemElement with the given name from the current directory.

If the FileSystemElement is not found (i.e., getChildByName(name) returns null), it prints an error message and returns from the method.

If the FileSystemElement is found, it gets the path of the FileSystemElement relative to the current directory and removes the FileSystemElement from the current directory.

It then changes the current directory to the directory specified by newPath using the changeDirectory(newPath) method. If the directory change is successful (i.e., changeDirectory(newPath) does not return null), it adds the FileSystemElement to the new directory and sets the parent of the FileSystemElement to the new directory.

Finally, it changes the current directory back to the original directory (i.e., the directory before the move operation). This is done to ensure that the current directory remains the same after the move operation, regardless of whether the move operation was successful or not.

FileSystemElement findElement(Directory directory, String name)

The method takes two arguments: directory, which is the directory to start the search from, and name, which is the name of the file or directory to find.

It iterates over each child of the given directory. A child can be either a file or a directory.

For each child, it checks if the name of the child matches the given name. If it does, it returns the child, indicating that it found the file or directory.

If the name of the child does not match the given name and the child is a directory, it calls itself recursively with the child directory and the given name as arguments. This allows it to search the subdirectory for the file or directory.

If the recursive call finds the file or directory in the subdirectory (i.e., it does not return null), it returns the found file or directory.

If it has checked all the children of the directory and has not found the file or directory, it returns null, indicating that the file or directory was not found in the given directory or its subdirectories.

void printDirectoryTree(Directory directory, int level)

The method takes two arguments: directory, which is the directory to print, and level, which is the current level of indentation.

It first prints the name of the current directory, indented by level number of times. If the current directory is the same as the directory being printed, it adds " (Current Directory)" to the end of the name.

It then checks if the directory has any children (i.e., files or subdirectories). If it does not, it prints a line indicating that the directory is empty, indented by level + 1 number of times.

If the directory does have children, it iterates over each child. For each child, it checks if the child is a directory. If it is, it calls itself recursively with the child directory and level + 1 as arguments, which prints the child directory and its subdirectories. If the child is not a directory (i.e., it's a file), it prints the name of the file, indented by level + 1 number of times.

This process continues until all directories and their subdirectories have been printed, creating a visual representation of the directory structure.

void printSortedDirectoryByTimestamp()

The method first creates a new ArrayList called sortedChildren that contains all the children (i.e., files and subdirectories) of the current directory.

It then sorts sortedChildren using the Collections.sort() method. The sorting is done based on the date and time each file or directory was created, which is retrieved using the getDateCreated() method of the FileSystemElement class. The Comparator used for sorting compares the creation dates of two FileSystemElement objects and returns a negative number, zero, or a positive number depending on whether the first date is less than, equal to, or greater than the second date.

It creates a SimpleDateFormat object to format the creation dates for printing.

It prints a message indicating that it's printing the contents of the directory sorted by the date created.

It then iterates over each child in sortedChildren (which are now sorted by creation date). For each child, it determines whether the child is a directory or a file by checking if the child is an instance of the Directory class. If it is, it sets type to "/", otherwise it leaves type as an empty string. It then formats the creation date of the child using the SimpleDateFormat object and prints the name, type, and formatted creation date of the child.

Recep Furkan Akın

210104004042