CSE 222

Data Structures and Algorithms

4th Assignment

Report

Author

Mert Emir ŞEKER

200104004085

Date

22.04.2024

Table Of Contents

Makefile:

```makefile
JC = javac
JFLAGS = -classpath .
JD = javadoc
JDFLAGS = -protected -splitindex -use -author -version -d ./javadoc
RM = rm
JR = java

CLASSES = \
        Directory.java \
        File.java \
        Main.java \
        FileSystem.java \
        FileSystemElement.java


all : Main.class

run :
    $(JR) Main

classes : $(CLASSES:.java=.class)

%.class : %.java
    $(JC) $(JFLAGS) $<

doc:
    $(JD) $(JDFLAGS) *.java

clean:
    $(RM) *.class

cleandoc:
    $(RM) -r ./javadoc
```

Makefile Commands:

All: Compiles the java files.

Run: Runs the program.

Clean: Removes compiled .class files. This command removes temporary files created during compilation.

Classes: Converts all .java files into .class files, ensuring compilation of each Java class.

Doc: Generates JavaDoc documentation. This command documents all Java files in the project and saves the results to the javadoc folder.

Cleandoc: Removes generated JavaDoc documentation. This command removes the documents created during documentation generation.

Code Explanation:

1) **Change directory:** The file system's change_directory function makes directory changes easier. It first determines whether the path supplied is the root directory ("/"). If it is, it updates the current directory to the root and returns true. Otherwise, it returns false. In the absence of that, it removes the leading slash from the path for consistent treatment and initializes a variable that represents the root directory. It then calls change_directory_recursive, a recursive helper method, passing in the root directory, an array of path segments that were created by splitting the route, and an initial index of 0.

   Conversely, change_directory_recursive modifies the current directory by recursively exploring the directory structure in accordance with the supplied path segments. After verifying that every component has been handled, it modifies the current directory and returns true. Alternatively, it manages distinct scenarios for every path element, namely ignoring empty or "." segments, ascending to the parent directory for ".." segments, and descending into subdirectories for directory headings. Lastly, it uses the current path portion to call itself recursively with updated parameters.

2) **List Directory Contents:** The contents of the current directory are printed to the console using the list_directory_contents method. The contents are printed recursively with the proper indentation after a notice identifying the directory being listed is displayed. This is done by calling the list_contents_recursive function.

   In contrast, list_contents_recursive recursively displays the directory's contents together with the given indentation. It prints the names of each child element in the directory one after the other, before the specified indentation string. In the event that a child is a directory, its type is indicated by appending a "/" to its name; if it is a file, its name is printed. By uncommenting a line inside the method, the contents of subdirectories can be listed recursively, making the procedure really recursive if desired.

3) **Create file/directory:** The file system's create_file_or_directory function makes it easier to create files or directories. Initially, it divides the given path into segments and ascertains if it is a recursive case (many segments) or a base case (single segment). To create the file or directory directly in the current directory, the base case invokes another overloaded method called create_file_or_directory. In the instance that is recursive, it extracts the subsequent segment of the path, verifies if the relevant directory is there, and generates it if required. With the remaining path, it then recurses into the next directory.
   Conversely, the overloaded create_file_or_directory method creates a file or directory in the provided directory with the given name. If an element with the same name already exists, it checks to see if it outputs an error notice and creates the new file or directory in accordance. The method accepts erroneous type specifications effectively and distinguishes between file ("f") and directory ("d") types.

4) **Delete file/directory:** A child element (file or directory) with the given name is removed from the directory using the remove_child method. It begins by determining whether the child is present. If it is, it removes the child from the directory's children list and recursively clears its contents for directories. The clear_all_children function iterates over a directory's children, clearing their contents and deleting them from the list in order to recursively delete all of them.

5) **Move file/directory:** By recursively searching through the directory structure, the find_directory_by_path method finds a directory at the given path within the file system. If the directory is found, it returns it; else, null. A file or directory can be moved from the current directory to the designated target path using the move_file_or_directory method. It finds the element that has to be moved and the target directory, moves the element, looks for conflicts, and sends back a message with the outcome.

6) **Search file/directory:** Beginning in the root directory, the search_file_or_directory method looks up a file or directory by name. With the root directory as the starting point and the name to search for, it calls the search_recursive method. It returns a message giving the path of the element if it is discovered; if not, it returns a message saying that the file or directory could not be located. Within the given directory, the search_recursive method recursively looks for a file or directory. It returns the directory itself after first verifying that the current directory and the name match. If not, iterating over each child element, it determines whether or not their names correspond to the supplied name. It makes recursive calls to itself on whatever directory it encounters. It returns the element if it can be located; if not, null is returned. The supplied file system element's whole path is created by the get_path method, starting with the parent directory and ending at the root. It creates the complete path string by reversing the traversal order as it iterates over the element's parent references, attaching the name of each directory to a StringBuilder.

7) **Print Directory Tree:** The print_directory_contents method prints the contents of the current directory, properly indented, after printing the path to the directory from the root. Using the build_path function, it recursively builds the path from the current directory's parent, and then prints it in reverse order to show from the root to the current directory. By supplying the directory and the determined indentation level to the print_current_directory_contents function, it proceeds to print the contents of the current directory. By appending the names of each directory to a list, the build_path method recursively creates the path from the root directory to the designated directory. The path items are printed with indentation using the print_path function.The print_current_directory_contents method, which prints files without a slash and marks directories with an asterisk (*) and a slash (/) after their names, is the last method to print the contents of the current directory. Recursively handling directories, it adds more indentation for a hierarchical representation.

8) **Sort contents by date created:** The current directory's and its subdirectories' contents are sorted using the sort_contents_by_date method according to the date of creation. It uses the Collections.sort method with a comparator based on the creation date to sort the current directory's immediate children. Next, it uses the sort_contents_recursive method to recursively sort each directory's contents.Based on creation date, the contents of the provided directory and its subdirectories are sorted recursively using the sort_contents_recursive method. It calls itself recursively for every directory found, iterates through each child directory, sorts directories if encountered, and so on.The current directory's and its subdirectories' sorted contents, together with their respective creation dates, are printed using the print_sorted_contents method. By executing sort_contents_by_date, it first verifies that the contents are sorted before printing the name and creation date of each element.The path from the root directory to the designated directory is created using the get_path_from_root function. By concatenating the names of each directory from the supplied directory up to the root directory, it recursively constructs the path.

Example Outputs:

1) Change Directory:

2) List directory contents:


```
seker@DESKTOP-1EUB7L7: /mnt/c/Users/merts/OneDrive/Belgeler/homeworkjava/hw4/src
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Belgeler/homeworkjava/hw4/src$ make all
javac -classpath . Main.java
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Belgeler/homeworkjava/hw4/src$ make run
java Main
Welcome to the File & Folder Directory Tree System !!!

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 1

Current directory: /home/user/Documents
Enter new directory path: /home/user/Documents/Projects
Directory changed to: /home/user/Documents/Projects

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 2

Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
Report.docx
```

3) Create file/directory:

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 3

Current directory: /home/user/Documents/Projects
Create file or directory (f/d): d
Enter name for new directory: Project3
Directory created: Project3/

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 3

Current directory: /home/user/Documents/Projects
Create file or directory (f/d): f
Enter name for new file: MeetingNotes.txt
File created: MeetingNotes.txt

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 2

Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
Report.docx
* Project3/
MeetingNotes.txt
```

4) Delete file/directory:

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 4
Current directory: /home/user/Documents/Projects
Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
Report.docx
* Project3/
MeetingNotes.txt
* Project4/
Main.java
Enter the name of the file/directory to delete: Main.java
Deletion successful.

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 4
Current directory: /home/user/Documents/Projects
Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
Report.docx
* Project3/
MeetingNotes.txt
* Project4/
Enter the name of the file/directory to delete: Project4
Deletion successful.

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 2

Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
Report.docx
* Project3/
MeetingNotes.txt
```

5) Move file/directory:

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 5
Current directory: /home/user/Documents/Projects
Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
Report.docx
* Project3/
MeetingNotes.txt
Enter the name of the file/directory to move: Report.docx
Enter new directory path: /home/user/Documents/Reports
File moved: Report.docx to /home/user/Documents/Reports

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 2

Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
* Project3/
MeetingNotes.txt

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 1

Current directory: /home/user/Documents/Projects
Enter new directory path: /home/user/Documents/Reports
Directory changed to: /home/user/Documents/Reports

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 2

Listing contents of Reports:
Report.docx
```

6) Search file/directory:

```
seker@DESKTOP-1EUB7L7:/mnt/c/Users/merts/OneDrive/Belgeler/homeworkjava/hw4/src$ make run
java Main
Welcome to the File & Folder Directory Tree System !!!

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 1

Current directory: /home/user/Documents
Enter new directory path: /home/user/Documents/Projects
Directory changed to: /home/user/Documents/Projects

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 2

Listing contents of Projects:
* Project1/
* Project2/
Notes.txt
Report.docx

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 6
Search query: Notes.txt
Searching from root...
Found: Found: root/home/user/Documents/Projects/Notes.txt
```

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 1

Current directory: /home/user/Documents/Projects
Enter new directory path: /home/user/Documents
Directory changed to: /home/user/Documents

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 6
Search query: Project2
Searching from root...
Found: Found: root/home/user/Documents/Projects/Project2
```

7) Print directory tree:

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 1

Current directory: /home/user/Documents
Enter new directory path: /home/user/Documents/Projects
Directory changed to: /home/user/Documents/Projects

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 7
Path to current directory from root:
* root/
  * home/
    * user/
      * Documents/
        * Projects/ (Current Directory)
          * Project1/
          * Project2/
          Notes.txt
          Report.docx
```

8) Sort contents by date created:

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 3

Current directory: /home/user/Documents/Projects
Create file or directory (f/d): d
Enter name for new directory: Project3
Directory created: Project3/

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 3

Current directory: /home/user/Documents/Projects
Create file or directory (f/d): f
Enter name for new file: MeetingNotes.txt
File created: MeetingNotes.txt

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 3

Current directory: /home/user/Documents/Projects
Create file or directory (f/d): d
Enter name for new directory: Project4
Directory created: Project4/

===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option: 8
Sorted contents of root/home/user/Documents/Projects:
* Project1/ (2024-04-21 18:01:57.637)
* Project2/ (2024-04-21 18:01:57.637)
Notes.txt (2024-04-21 18:01:57.637)
Report.docx (2024-04-21 18:01:57.637)
* Project3/ (2024-04-21 18:05:53.594)
MeetingNotes.txt (2024-04-21 18:06:05.481)
* Project4/ (2024-04-21 18:06:12.067)
```