T.C.

GEBZE TEKNİK ÜNİVERSİTESİ

CSE 222 – DATA STRUCTURES AND ALGORITHMS

HW4 - File Tree System REPORT

Mehmet Can OLÇAY

22.03.2024

1.FileSystemElement Class

2.Directory Class

3.File Class

4.FileSystem Class

5.Main Class

1.FileSystemElement Class :

The FileSystemElement class serves as a foundational component in the design of a file system structure. It encapsulates essential attributes and behaviors common to both files and directories within the file system. This report provides an overview of the FileSystemElement class and its significance in the broader context of file system implementation.

Class Overview:

The FileSystemElement class is an abstract class, designed to be extended by concrete subclasses representing specific types of file system elements such as files and directories. It defines several key attributes and methods that are shared by all file system elements.

Attributes:

Name: Each file system element has a name, which uniquely identifies it within its parent directory.

Parent: A reference to the parent directory of the file system element, allowing for navigation and hierarchical organization.

Date Created: Represents the timestamp when the file system element was created, providing metadata about its creation time.

Constructor:

The class constructor initializes the attributes of the file system element, namely the name, parent directory, and creation timestamp. It ensures that each instance of a file system element is properly initialized upon creation.

Accessor Methods:

getName(): Returns the name of the file system element.

getDateCreated(): Retrieves the timestamp indicating the creation date of the file system element.

getParent(): Returns a reference to the parent directory of the file system element.

Mutator Method:

setParent(): Allows for the modification of the parent directory of the file system element, enabling dynamic restructuring of the file system hierarchy.

Abstract Method:

The print() method is declared as abstract, indicating that concrete subclasses must provide their implementation. This method is responsible for printing information about the file system element, typically used for displaying the structure of the file system.

2.Directory Class:

The Directory class is a fundamental component of the file system structure. It represents a directory within the file system, which can contain files and subdirectories. This report provides an overview of the Directory class and its functionalities.

Class Structure:

The Directory class extends the FileSystemElement class, inheriting properties such as name, parent directory, and creation timestamp. Additionally, it maintains a list of children elements using a linked list data structure.

Functionality:

Constructor: The class constructor initializes a directory with a name and parent directory. It also initializes the list of children elements.

Adding Elements: The addElement method allows adding files or subdirectories to the directory's list of children.

Removing Elements: The removeElement method allows removing files or subdirectories from the directory's list of children.

Accessing Children: The getChildren method returns the list of children elements, providing access to files and subdirectories contained within the directory.

Printing Directory Structure: The print method overrides the abstract method defined in the FileSystemElement class. It prints the name of the directory preceded by a prefix and marked with an asterisk (*) to denote it as a directory.

3.File Class:

The File class represents a specific type of file system element within a file system structure. It extends the abstract FileSystemElement class and provides functionality specific to files. This report provides an overview of the File class, its purpose, and functionality.

Class Overview:

The File class is designed to encapsulate attributes and behaviors related to individual files within the file system. It inherits common functionalities from the FileSystemElement class while providing specialized functionality for files.

Constructor:

The class constructor initializes a file with a given name and parent directory. It ensures proper initialization of the file's attributes upon creation, including the name and parent directory, by invoking the constructor of the superclass FileSystemElement.

Print Method Override:

The print method overrides the abstract print method defined in the superclass FileSystemElement. It is responsible for printing information about the file, typically its name, with a specified prefix. This method facilitates the display of file information within the file system structure.

Functionality:

Name: Each file has a name that uniquely identifies it within its parent directory.

Parent Directory: A reference to the parent directory of the file, enabling navigation and organization within the file system hierarchy.

Printing: The class provides functionality to print the name of the file with a specified prefix, allowing for the visualization of the file structure.

4.FileSystem Class:

Introduction:

The FileSystem class serves as the core component responsible for managing the file system structure, including directories and files. It provides functionalities for creating, deleting, moving, and navigating through the file system hierarchy. This report presents an overview of the FileSystem class, its functionalities, and design considerations.

Class Overview:

The FileSystem class is designed to encapsulate the functionalities required for managing directories and files within a file system. It maintains a reference to the root directory, from which all operations within the file system are initiated. The class leverages the concept of composition and recursion to perform various file system operations efficiently.

Constructor:

The class constructor initializes the file system by creating the root directory with the name "root" and setting its parent to null. This ensures the establishment of a foundational directory structure upon the instantiation of the FileSystem object.

File System Operations:

1.Creating Directories and Files: The class provides methods to create directories and files within the file system. It ensures uniqueness of names by performing a search to prevent duplication of directory or file names.It creates directory and file according to given name and also name is unique attribute for this system.If there is a file exist with the same name , it says “there is a file exist …” .

```
D:\DirectoryTreeSystem\src>java Main
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 4
Enter directory name to create:
Directory1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 3
Enter file name to create:
File1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 9
Directory Tree:
*root/(Current Directory)
    *Directory1/
    File1
```

Figure 1.1

```
Directory Tree:
*root/(Current Directory)
    *Directory1/
    File1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 3
Enter file name to create:
Directory1
Same name exist in this system...
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option:
```

Figure 1.2

2.Moving Elements: The moveElement method facilitates the movement of files or directories within the file system hierarchy. It updates the parent reference of the element being moved and rearranges the directory structure accordingly.

```
    *Directory1/
        insidefile2
        *Directory2/(Current Directory)
            insidefile3
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 7
Enter file/directory name to change:
insidefile3
Enter directory path:
/root/Directory1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 9
Directory Tree:
*root/
    *Directory1/
        insidefile2
        *Directory2/(Current Directory)
        insidefile3
```

Figure 2.1

3.Change Directory: The changeDirectory method enables navigation through the file system structure by interpreting input paths as either absolute or relative and traversing directories accordingly.There is a option to reach parent directory writing ".." and also only with directory name , we can reach directories and also with full path there is a option to reach files.

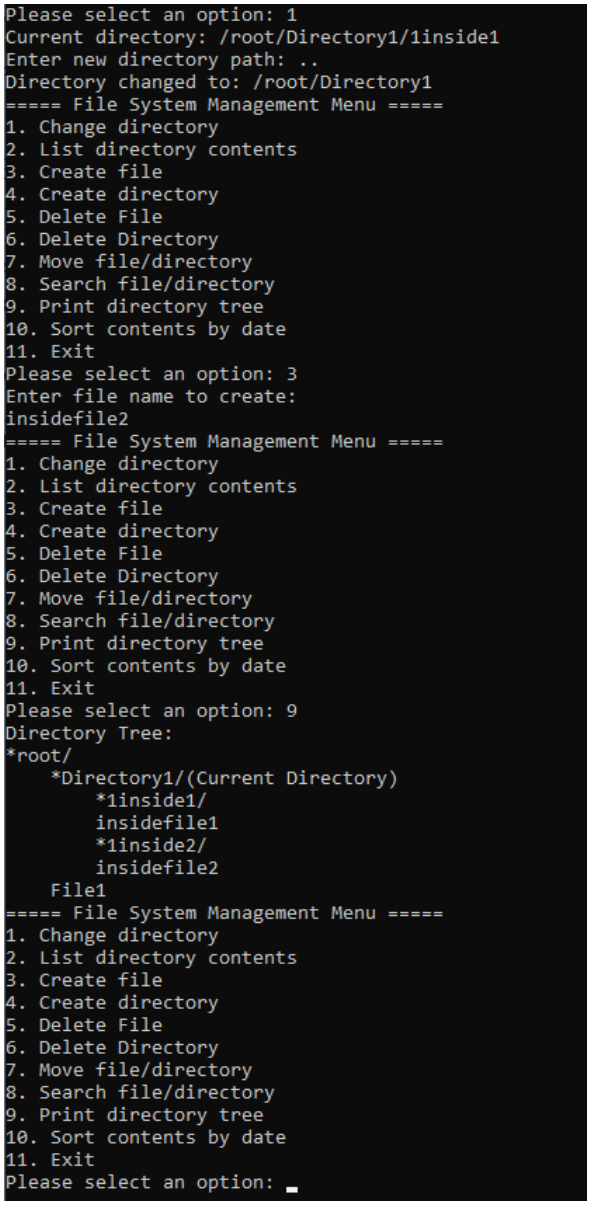

Figure 3.1

4.Listing Contents: The listContents method allows listing the contents of a directory, providing visibility into the files and subdirectories it contains.

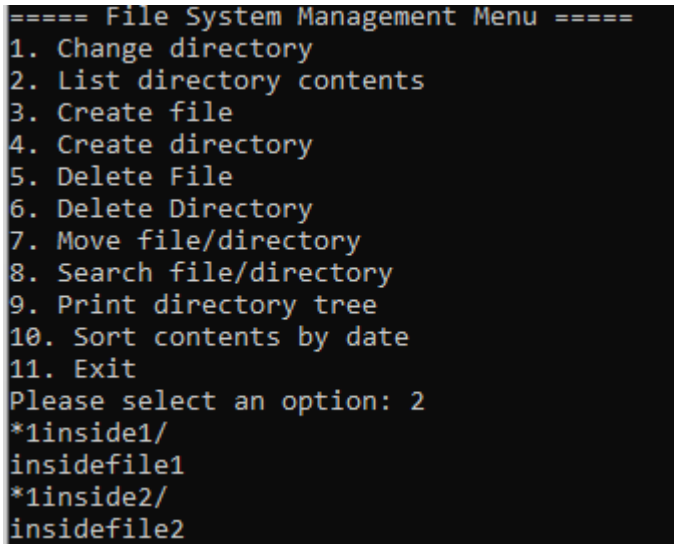

Figure 4.1

5.Deleting Elements: The class provides functionalities to delete files and directories from the file system. It recursively deletes directories and their contents to ensure a comprehensive cleanup operation.

```
Please select an option: 9
Directory Tree:
*root/
    *Directory1/(Current Directory)
        insidefile1
        insidefile2
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 5
Enter file name to delete:
insidefile1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 9
Directory Tree:
*root/
    *Directory1/(Current Directory)
        insidefile2
```

Figure 5.1

```
Please select an option: 9
Directory Tree:
*root/
    *Directory1/(Current Directory)
        *1inside1/
            fs1
            *directory4/
        insidefile1
        *1inside2/
        insidefile2
    File1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 6
Enter directory name to delete:
1inside1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 9
Directory Tree:
*root/
    *Directory1/(Current Directory)
        insidefile1
        *1inside2/
        insidefile2
    File1
```

Figure 5.2

6.Sorting Directory Contents: The sortDirectoryByDate method sorts the contents of a directory based on the creation date of the file system elements, providing an organized view of the directory's contents.The left one indicate the after move operation there is a file which is created before is exist.After sorting operation right picture happens.

```
Please select an option: 7
Enter file/directory name to change:
insidefile4
Enter directory path:
/root/Directory1
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 9
Directory Tree:
*root/
    *Directory1/
        insidefile2
        *Directory2/(Current Directory)
        insidefile3
        insidefile5
        insidefile4
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
```

Figure 6.1

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option: 10
Sorted contents of /root/Directory1 by date created:
2024-04-22 23:13:22.0 insidefile2
2024-04-22 23:17:16.0 *Directory2/
2024-04-22 23:18:33.0 insidefile3
2024-04-22 23:21:02.0 insidefile4
2024-04-22 23:21:55.0 insidefile5
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file
4. Create directory
5. Delete File
6. Delete Directory
7. Move file/directory
8. Search file/directory
9. Print directory tree
10. Sort contents by date
11. Exit
Please select an option:
```

Figure 6.2

7.Searching Elements: The class provides methods to search for specific files or directories within the file system, facilitating efficient retrieval of file system elements by name.

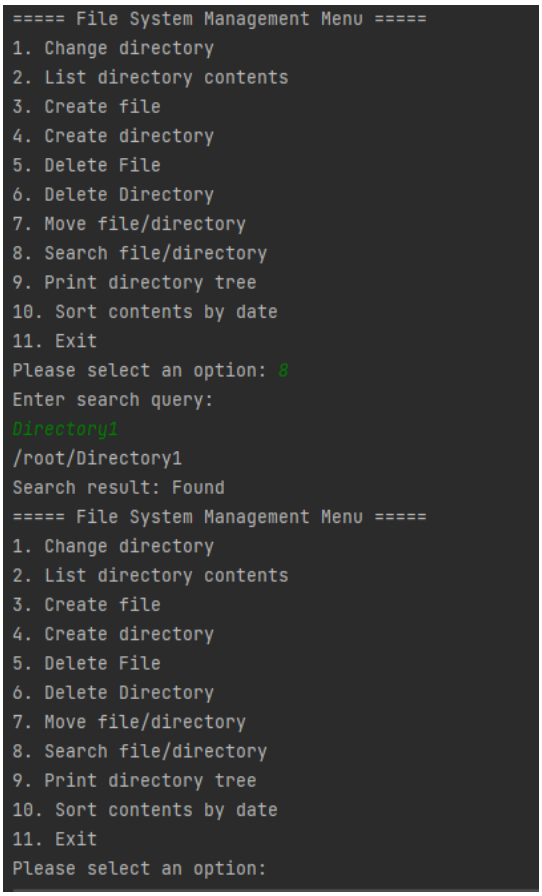

Figure 7.1

8.Printing Directory Tree: The printDirectoryTree method prints the hierarchical structure of the file system as a tree, visualizing the relationships between directories and files.Also it writes current directory.

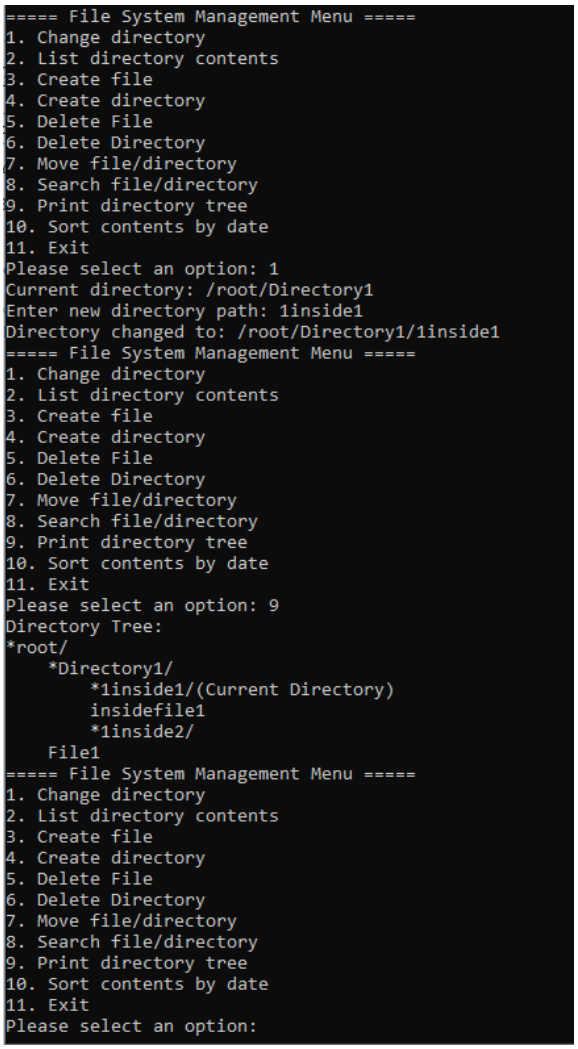

Figure 8.1

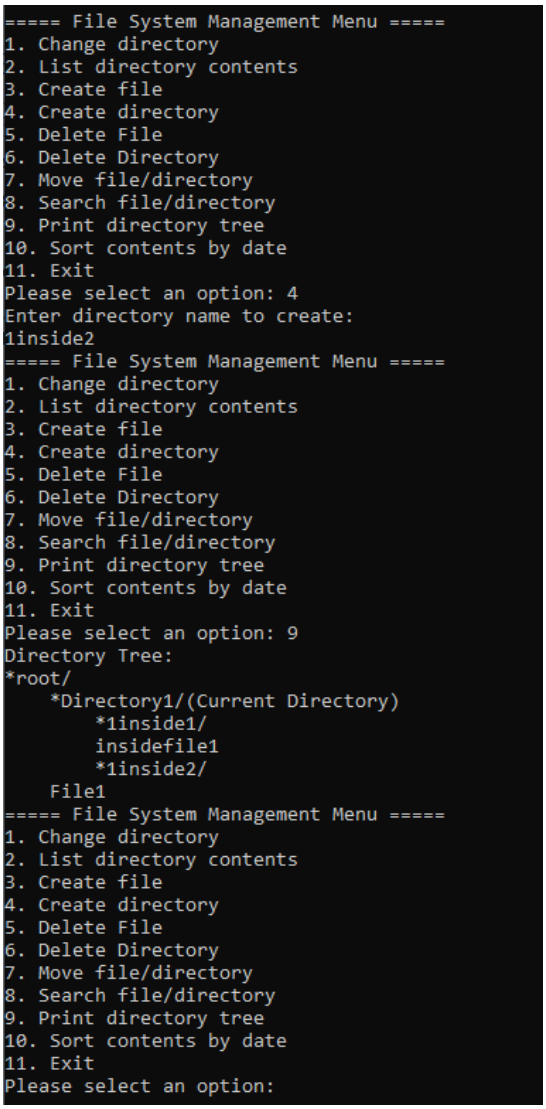

Figure 8.2

5.Main Class:

The Main class serves as the entry point and user interface for interacting with the file system management system. It provides a menu-driven interface to perform various file system operations such as navigating directories, creating and deleting files/directories, searching for elements, and printing directory structures. This report presents an overview of the Main class, its functionalities, and user interaction.

Class Overview:

The Main class orchestrates user interaction and delegates file system operations to the FileSystem class. It leverages a Scanner object to receive user input and presents a menu-based interface to facilitate user interaction with the file system management system.

User Interface:

The Main class presents a menu-driven interface to the user, allowing them to select from a range of file system operations. Each menu option corresponds to a specific operation, enabling users to perform tasks such as changing directories, listing contents, creating files/directories, deleting files/directories, moving elements, searching, printing directory tree, and sorting contents.

File System Operations:

Change Directory: Enables users to navigate through the file system hierarchy by specifying directory paths.

List Directory Contents: Displays the contents of the current directory, providing visibility into files and subdirectories.

Create File/Directory: Allows users to create files or directories within the file system structure.

Delete File/Directory: Enables users to delete files or directories from the file system.

Move Element: Facilitates the movement of files or directories within the file system hierarchy.

Search: Provides functionality to search for specific files or directories within the file system.

Print Directory Tree: Prints the hierarchical structure of the file system as a tree, visualizing the relationships between directories and files.

Sort Contents by Date: Sorts the contents of a directory based on the creation date of the file system elements.

Implementation Details:

The Main class leverages the FileSystem class to perform file system operations. It handles user input validation and error handling to ensure a smooth user experience. The class follows a modular design, with each operation implemented as a separate method to promote code readability and maintainability.