**Java MiniShell**

**Project Overview**

Java MiniShell is a lightweight, educational command-line interface application that simulates essential Unix/Linux shell functionality. This Java-based implementation provides users with a simplified environment to navigate and manipulate their file system using familiar shell commands.

**Features**

The application supports the following commands:

* pwd: Display the full path of your current location
* ls: View all files and folders in your current directory
* cd [folder]: Navigate to different directories (including parent directory using ..)
* mkdir [name]: Create new directories in your current location
* touch [name]: Generate empty files in your current location
* help: View all available commands with descriptions
* exit: Close the application

**Project Structure**

The application consists of two Java classes:

1. **MiniShell.java**: Core application class that processes user input and manages the command loop
2. **ShellCommandHandler.java**: Helper class that implements file system operations and command logic

**Implementation Details**

**ShellCommandHandler Class**

This utility class handles all file system interactions:

* **Constructor**: Sets up the initial working directory using System.getProperty("user.dir")
* **printWorkingDirectory()**: Outputs the absolute path of your current location
* **listDirectory()**: Displays files and directories with appropriate [FILE] or [DIR] labels
* **changeDirectory(String folder)**: Handles navigation between directories with support for both direct child folders and parent directory
* **makeDirectory(String name)**: Creates new directories with existence validation
* **createFile(String name)**: Generates empty files with error handling for existing files
* **printHelp()**: Shows a user-friendly list of all available commands
* **getCurrentDirectory()**: Provides access to the current File object

**MiniShell Class**

The main execution class that:

1. Initializes the command handler and input scanner
2. Presents a welcome message
3. Runs the main command loop:
   * Displays a prompt showing the current directory path
   * Processes user input to separate commands and arguments
   * Routes commands to appropriate handler functions
   * Provides feedback on command execution
   * Continues until user requests to exit

**How to Run**

1. Compile the source files:

javac ShellCommandHandler.java MiniShell.java

1. Execute the shell:

java MiniShell

**Sample Usage**

Welcome to MiniShell! Type 'help' for a list of commands.

/Users/student > help

Commands you can use:

pwd = see current folder

ls = list files

cd [folder] = change folder

mkdir [name] = make new folder

touch [name] = make new file

help = see this message

exit = quit program

/Users/student > mkdir projects

Made new folder: projects

/Users/student > cd projects

You are now in: /Users/student/projects

/Users/student/projects > touch notes.txt

Made new file: notes.txt

/Users/student/projects > ls

[FILE] notes.txt

/Users/student/projects > cd ..

You are now in: /Users/student

/Users/student > exit

Goodbye!

**Error Handling**

The program includes error handling for common scenarios:

* Missing command arguments (e.g., mkdir without folder name)
* Navigation to non-existent directories
* Attempts to create directories or files that already exist
* IOException handling when creating files
* Invalid command detection with helpful guidance

**Limitations**

* No support for complex shell features (pipes, wildcards, redirections)
* Limited to fundamental file system operations
* No file content editing capabilities
* Simple command parsing without support for quoted arguments
* Cannot handle operations on multiple files simultaneously
* No persistent command history between sessions

**Future Enhancements**

* Implement file reading and writing functionality
* Add command history with up/down arrow navigation
* Support copy, move, and delete operations
* Add wildcard pattern matching for file operations
* Implement simple text editing capabilities
* Add auto-completion for file and directory names
* Support for environment variables
* Add ability to execute external programs