Assumptions:

General:

- We do not deal with filenames containing dash (-) (e.g. -1.txt)
- ClassNamingConventions violation for the integration tests are suppressed because project description for integration tests says to follow the naming convention *IT.java

Basic Functionalities:

ECHO:

- The echo command outputs the specified arguments followed by a newline character. If invoked without any arguments, it prints an empty line at the end.

CD:

- Executing the cd command with null or blank arguments will not navigate to the root directory; instead, it leaves the shell in the current working directory.
- Executing the cd command with multiple arguments will result in an error, as it does not handle simultaneous navigation to multiple directories.
- `Cd ..` changes the current directory to the parent directory of the current directory

WC:

- Order of the flags does not matter
- The wc command will always output an indent in front of every line and output a `STRING_NEWLINE` at the end of every line
- For stdin, a dash (-) will be added at the end of the line (for the respective nth argument) instead of the filename regardless of how many files are used
- If no arguments are specified, will read from stdin

MKDIR:

- If -p option is not provided and a specified directory already exists, an error should be raised

SORT:

- Order of the flags does not matter

CAT:

- When concatenating 2 files into one, only use “>” redirection (eg. cat file1.txt file2.txt > union.txt)
- If there is no redirection, the output will only be printed in stdout
- If file is directory, will display exception “path: This is a directory”

EXIT:

- The exit command can be invoked with any number of arguments without affecting its functionality

Quoting:

- Double quotes can disable the interpretation of all special symbols except backquote, while single quotes disable the interpretation of all special symbols
- Double quotes refer to straight double quotes and single quotes refer to straight single quote

Extended functionalities 1:

LS:

- If `ls` is run on a file (not a directory), the expected output should be 'ls: cannot access ‘file'', indicating that 'ls' can only list contents within directories, not files.
- If there are multiple files in the argument, the output will be separated by a STRING_NEWLINE

PASTE:

- Output does not have STRING_NEWLINE at the end of the string
- Line separator is used instead of \n for new line
- Have a nopmd at the class because we implemented each combination separately, hence it seems like it is very long, having a possibility of a godclass warning
- No flag, null input and output stream, file error will have paste exceptions

UNIQ:

- Have a nopmd at the class because we implemented each combination separately, hence it seems like it is very long, having a possibility of a godclass warning

MV:

- If same argument is specified for source and dest and the argument specified is not a directory, no exception is thrown
- If same argument is specified for source and dest and the argument specified is a directory, exception is thrown

- If source is moved to the same folder, exception is thrown
- If given source does not exist, exception is thrown

Extended functionalities 2:

CUT:

- The cut command is able to process more than 1 comma separating number
- Output does not have `STRING_NEWLINE` at the end of the string

RM:

- The rm command processes files and directories in the order they are specified as parameters. For instance, in the command `rm file1 file2`, if `file2` does not exist, `file1` will still be removed. Conversely, in `rm file2 file1`, if `file2` does not exist, the removal of `file1` will not be attempted.
- The -d flag is applicable to both regular files and directories. This means commands like `rm -d file.txt` (for files) and `rm -d dir` (for directories) are both valid.
- The placement of the -d flag in the command is flexible. For example, `rm dir -d` is valid and will remove the specified directory.

TEE:

- If there is no operator provided, multiple files means all of the input by user will be saved into all of these files specified
- Prints everything after Ctrl+D

GREP:

- Order of flags does not matter
- The second argument will always be interpreted as the pattern, so if no third argument is specified, it will read from stdin
- If pattern is given and empty (“”) and file(s) are given from the third argument onwards, no exception is thrown and output all the lines in the file(s), following the behaviour in linux bash
- If a given file does not exist, an error will be shown, and then, later files will be evaluated in a similar way
- Filenames or (standard input) is shown by default when multiple files are used
- The grep command will always output a `STRING_NEWLINE` at the end of every line

Basic Functionalities:

Semicolon Operator:

- A new line is printed after every command output even if the command output throws an exception

Globbing:

- When all file are displayed, files within a file cannot be output
- The symbol * (asterisk) can not be used to find which file or folder is from which folder. Eg: “.*\test.txt” is not accepted