|  |  |  |
| --- | --- | --- |
| **LAB211 Assignment** | **Type:** | **Short Assignment** |
| **Code:** | **J1.S.P0078** |
| **LOC:** | **100** |
| **Slot(s):** | **2** |

**Title**

Create a program to copy file

**Background**

NA

**Program Specifications**

Write a program with file config.properties

1. COPY\_FOLDER=D:\ \Data.

2. DATA\_TYPE=\*.CSV,\*.WAV.

3. PATH=D:\\ Copy Data

When program run

If file config is not exist, prompt user to input file config with

* Enter Copy Folder:
* Enter Data Type:
* Enter Path:

After input, program create file config and perform next steps

If file config is existed, check file content for following cases

* COPY\_FOLDER
  + Not input yet.
  + Path to source folder is not existed
* DATA\_TYPE
  + Not input yet
* PATH
  + Not input yet
  + Input but not existed -> create a path.

If checking file config task has error, show error message and stop program

If file config completed then copy file config from COPY\_FOLDER to PATH

***Function details:***

**Function 1:** Display a menu and ask users to select an option.

* Users run the program. The program prompts users to select an option.
* Users select an option, perform **Function** **2**.

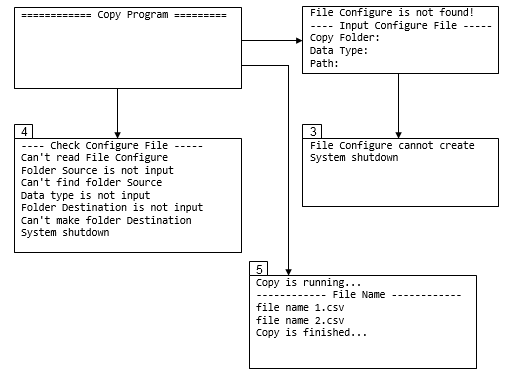
**Function 2:**

* If file config is not exist, prompt user to input config file with following format:
* Copy Folder:
* Data Type:
* Path:

Perform next steps with existed file

* If file existed
* Verify data in config file
* COPY\_FOLDER
  + Not input yet
  + Path to source folder is not existed
* DATA\_TYPE
  + Not input yet
* PATH
  + Not input yet
  + Input but not existed -> create a path.
* If checking file config task has error, show error message and stop program
* If file config completed then copy file config from COPY\_FOLDER to PATH

### *Expectation of User interface:*



**Guidelines**

**Student must implement methods**

readFileConfig

createFileConfig

checkConfig

copyFile

**in startup code.**

**Suggestion**

**Function 1:** Read file config.proprties

* Implement function: public Config readFileConfig(Config config) throws ExceptionHandle
  + Input:
* config: Object contains configure information.
  + Return value
* Object config has been fill all information
* ExceptionHandle list.

**Function 2:** Create file config if not existed

* Implement function: public void createFileConfig(Config config) throws ExceptionHandle
  + Input:
* config: Object contains configure information input from keyboard.
  + Return value:
* ExceptionHandle list.

**Function 3:** Verify file config

* Implement function: public void checkConfig(Config config) throws ExceptionHandle
  + Input:
* config: Object contains configure information.
  + Return value
* ExceptionHandle list.

**Function 4:** Perform copy task if file config is valid

* Implement function: public List<String> copyFile(Config config)
  + Input:
* config: Object contains configure information.
  + Return value:
* List of files copied