OO File System

Xia Li

2022.11.7

1. Abstraction

I developed this OO File System for registering file documents and managing the file system. I will use the overall design, table, data type, and design flow, these 4 aspects to introduce the entire system.

2. Technical proposal

2.1 Overall design

OO File System uses Java development, it can simulate and process Drive, Folder, Text File, and Zip Files four types of files in memory. The system supports browsing files under Driver and Folder, reading contents of Text File, and viewing compressed folders and files under Zip File. Supports Folder creation, Text File, Zip File creation, movement, and deletion; The content of a Text File can be modified

2.2 Class design

FileTypeEnum	
Type：	Enum class
Enum Type	Notes
TEXT	Text file
ZIP	Compressed file

BaseEntity		
Type：	Entity class	
Remark：	The base class of the file entity class, which stores the basic information of the file	
Type of variable	**Name**	**Notes**
String	fileName	Name of file
Double	fileSize	File size
String	filePath	File path
FileTypeEnum	fileType	File type

TextFileEntity		
Type：	Entity class	
Extends：	BaseEntity	
Remark：	Text file entity class	
Type of variable	**Name**	**Notes**
String	content	Content of text

ZipFileEntity		
Type：	Entity class	
Extends：	BaseEntity	
Remark：	Text file entity class	
Type of variable	**Name**	**Notes**
Map<String,List<BaseEntity>>	fileMap	In the compressed file hash table, the key stores the directory path under the compressed package, and the value stores the file in the directory.

FileManager		
Type：	File Management Class	
Remark：	Be in charge of Driver， Folder， File	
Type of variable	**Name**	**Notes**
Map<String,Map<String,List<BaseEntity>>>	fileMap	File hash table, the key used to store the Driver drive letter; value is also a hash table. The key of value is used to store the directory path under the Driver, and the value is used to store the collection of files under the path.

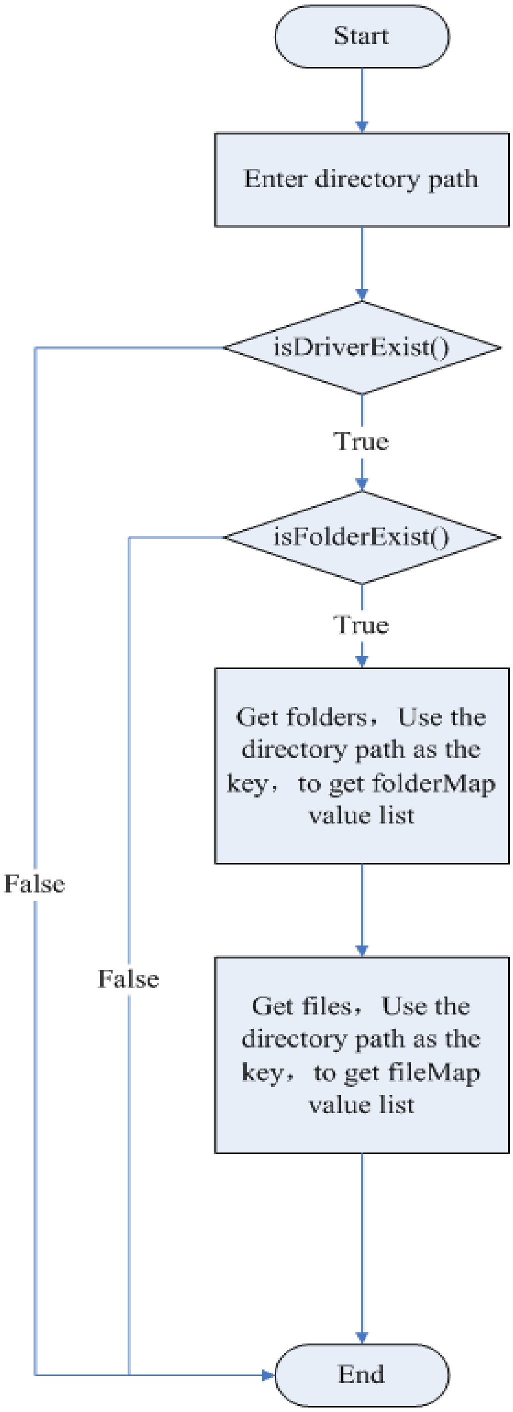
Map<String,Map<String,List<String>>>	folderMap	Directory hash table, key used to store the Driver drive letter; value is also a hash table. The key of value is used to store the parent directory path of the Driver, and value is used to store the collection of subdirectories of the path.
Method of Determine	**Method return value**	**Notes**
isDriverExist(String filePath)	Boolean	Check whether the drive letter exists according to the directory path. If the drive letter exists, return true; if it does not exist, return false
isFolderExist(String filePath)	Boolean	Check whether the directory under the drive letter exists according to the directory path. If yes, return true; if no, return false
isFileExist(String filePath)	Boolean	Check whether files exist in the directory under the drive letter according to the directory path. If files exist, return true; if files do not exist, return false
Method of Present	**Method return value**	Notes

listFolder(String folderPath)	List<BaseEntity>	Returns a collection of files in a directory based on the directory path
listZipFile(String zipFilePath)	List<BaseEntity>	According to the path of the compressed package, the directory and files in the compressed package are returned
readTxtFile(String filePath)	String	Read the contents of the text according to the path of the text file
Method of writing	**Method return value**	**Notes**
createDriver(String driverName)	Boolean	Create a Driver based on the drive letter
createFolder(String filePath)	Boolean	Create a directory based on the directory path
createTxtFile(TextFileEntity entity,String filePath)	Boolean	Create a TXT file based on the TXT path and TXT content
editTxtFile(TextFileEntity entity,String filePath)	Boolean	Modify the TXT file based on the TXT path and the latest TXT content
zipFolder(String zipFilePath,List<String> folderList)	Boolean	Create a ZIP file based on the ZIP path and transfer the ZIP file to the compressed file. If the ZIP file already exists, overlay the ZIP file to the original one

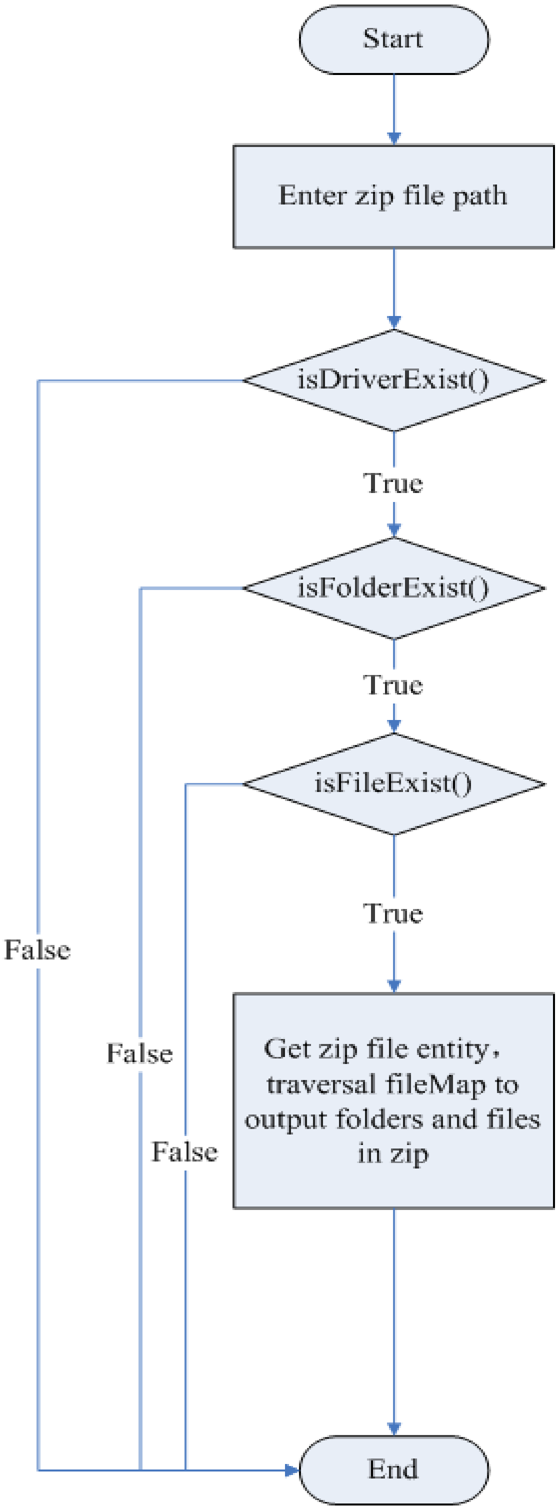
zipTxtFile(String zipFilePath,TxtFileEntity entity)	Boolean	Create a ZIP file based on the ZIP path, and transfer the txt file to the compressed file. If the ZIP file already exists, add the ZIP file to the original one
moveFile(String srcFilePath,String destFilePath)	Boolean	Move a folder or file based on the original and destination paths
deleteFile(String filePath)	Boolean	Delete a directory or file based on the path

2.3 Design Flow

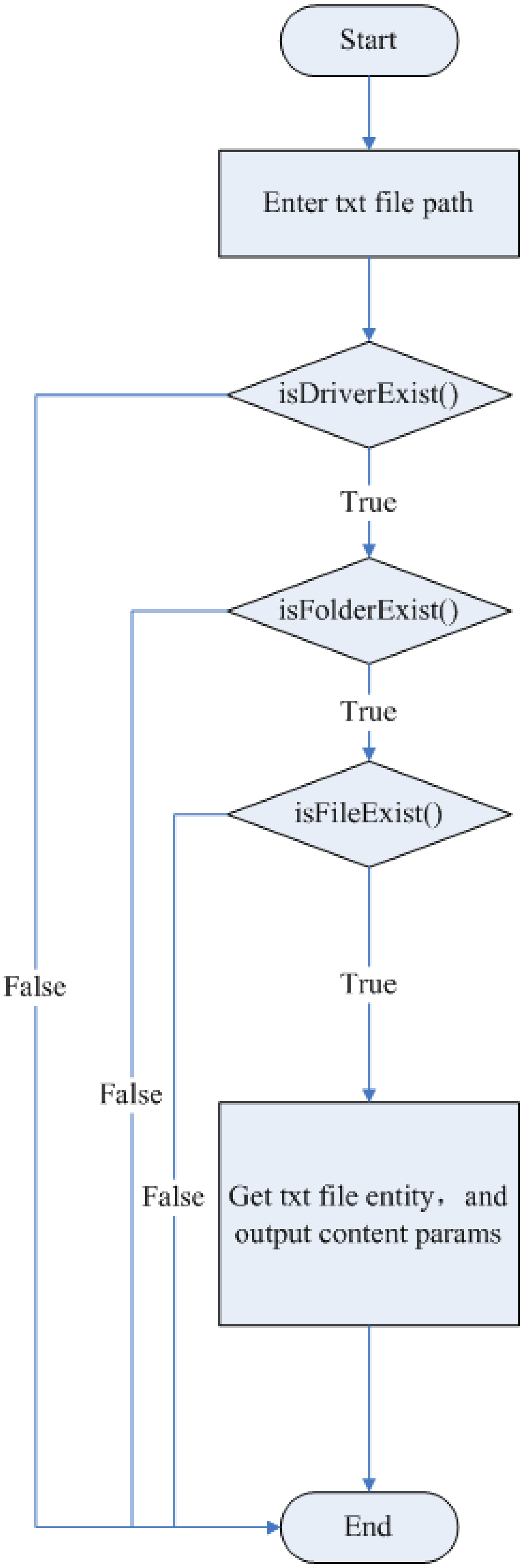
Lists the folders and files in the directory（listFolder）


Start
Enter directory path
isDriverExist()
True
isFolderExist()
True
Get folders，Use the directory path as the key，to get folderMap value list
Get files，Use the directory path as the key，to get fileMap value list
False
False
End

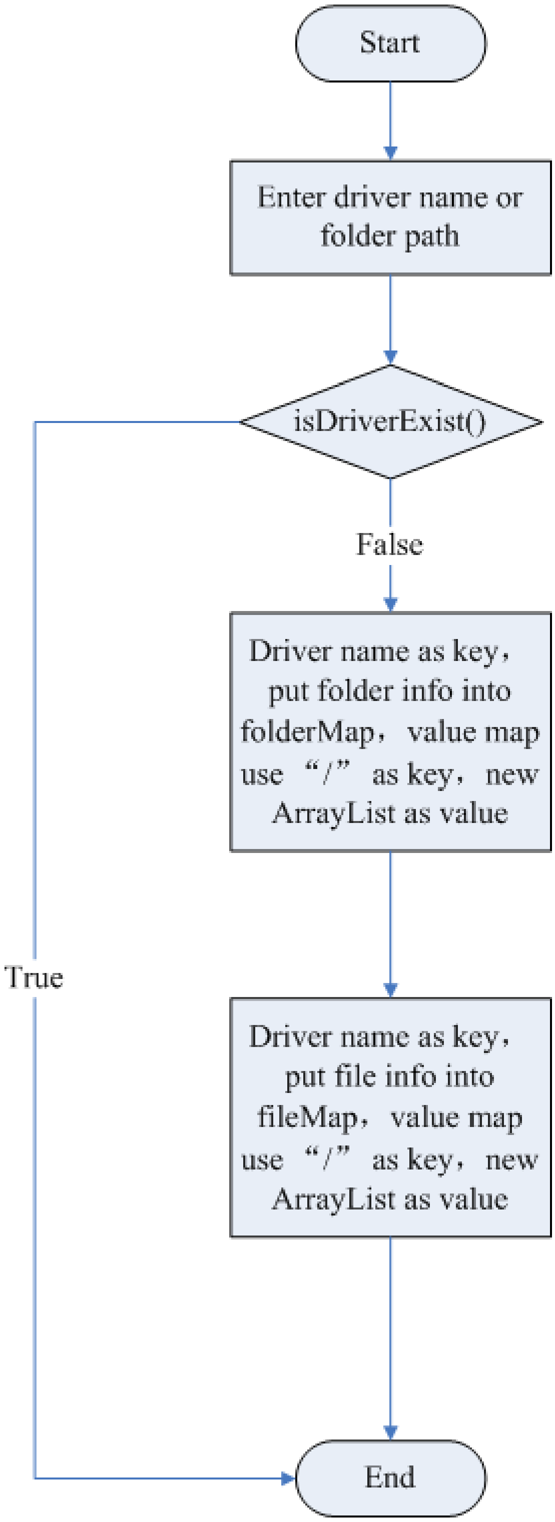
Lists the folders and files under the zip package（listZipFile）



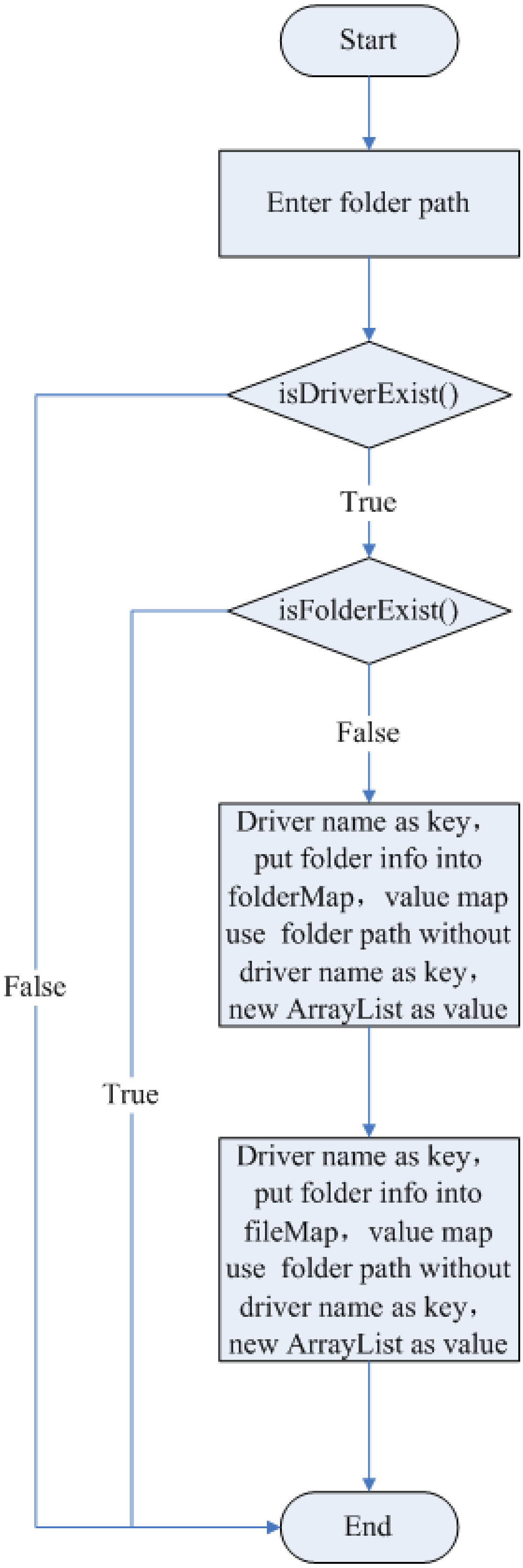
Read the contents of the TXT file（readTxtFile）



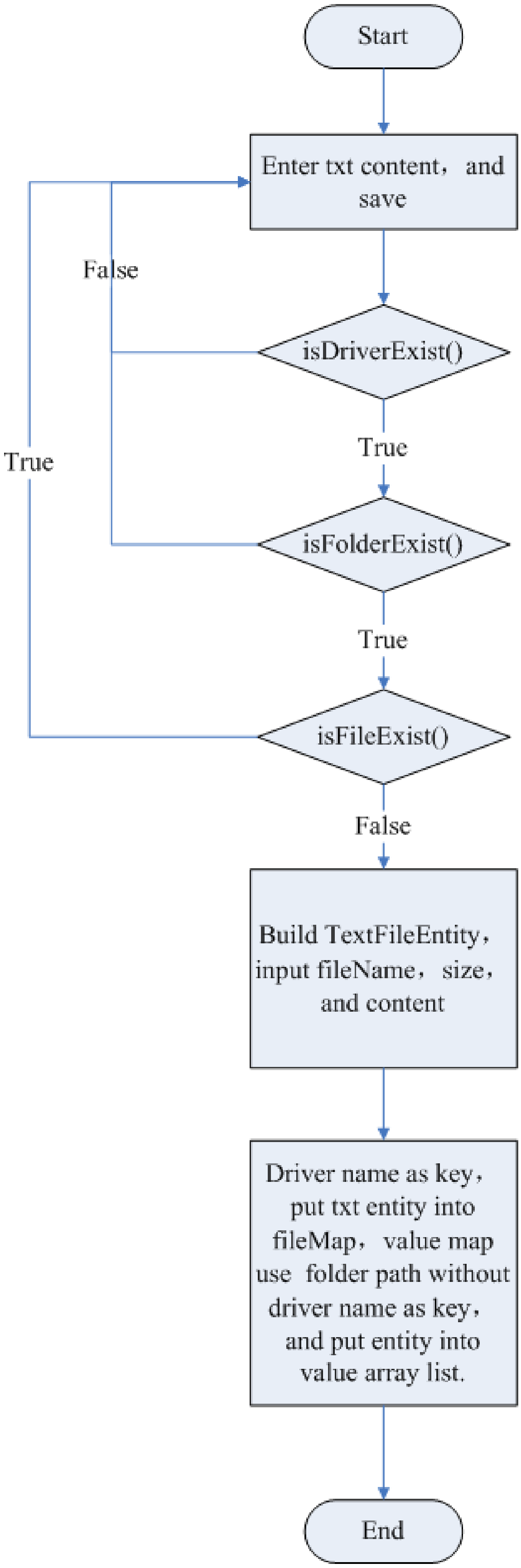
Creating a Drive Letter（createDriver）


Start
Enter driver name or folder path
isDriverExist()
False
True
Driver name as key，put folder info into folderMap，value map use “/” as key，new ArrayList as value
Driver name as key，put file info into fileMap，value map use “/” as key，new ArrayList as value
End

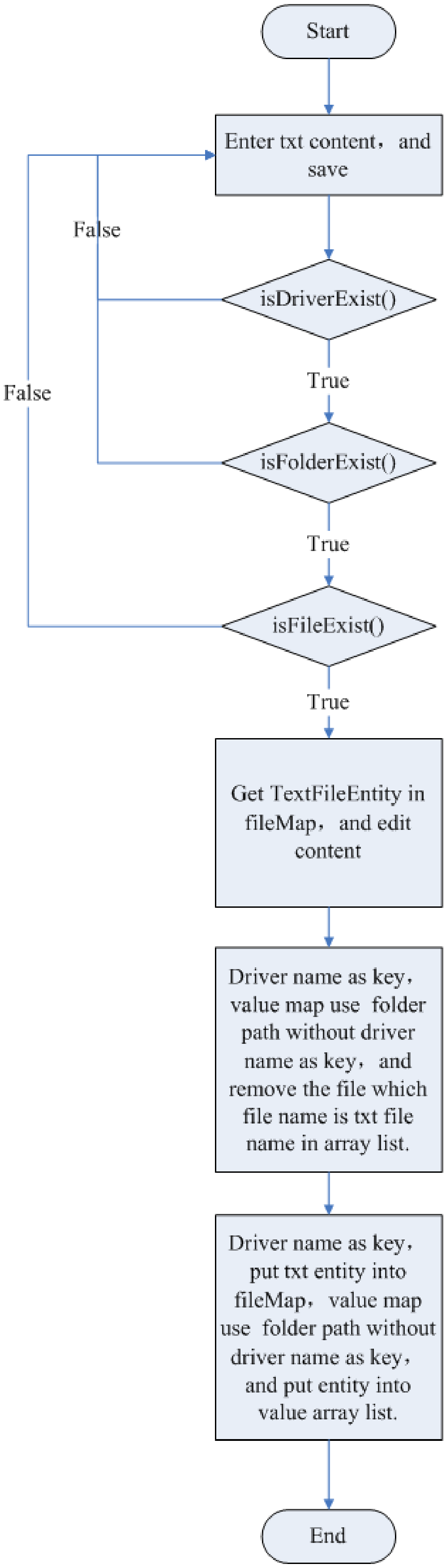
Create a Folder（createFolder）


Start
Enter folder path
isDriverExist()
True
isFolderExist()
False
Driver name as key，put folder info into folderMap，value map use folder path without driver name as key，new ArrayList as value
Driver name as key，put folder info into fileMap，value map use folder path without driver name as key，new ArrayList as value
False
True
End

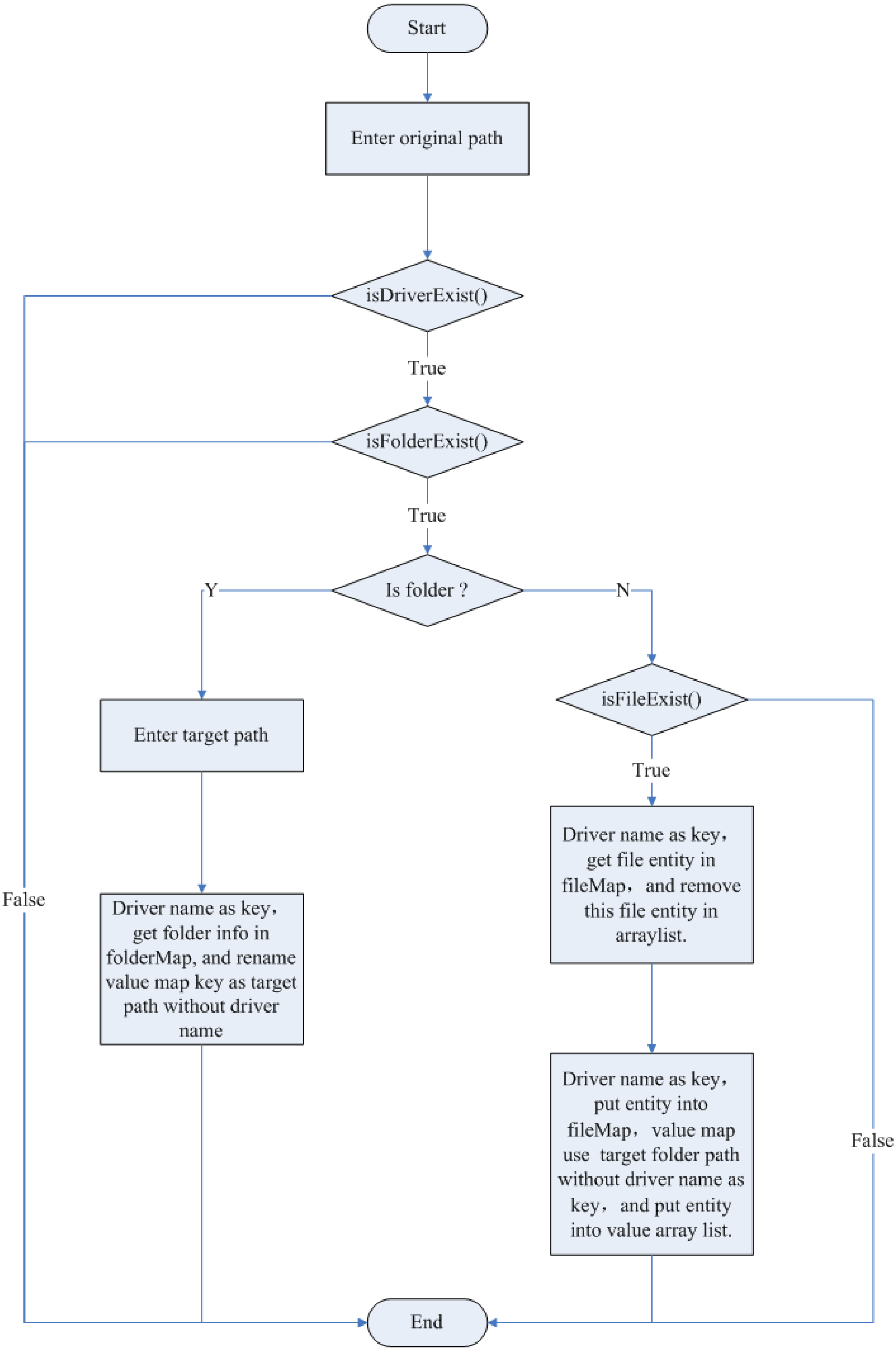
Create TXT file（createTxtFile）


Start
Enter txt content， and save
False
isDriverExist()
True
True
isFolderExist()
True
isFileExist()
False
Build TextFileEntity， input fileName， size， and content
Driver name as key， put txt entity into fileMap， value map use folder path without driver name as key， and put entity into value array list.
End

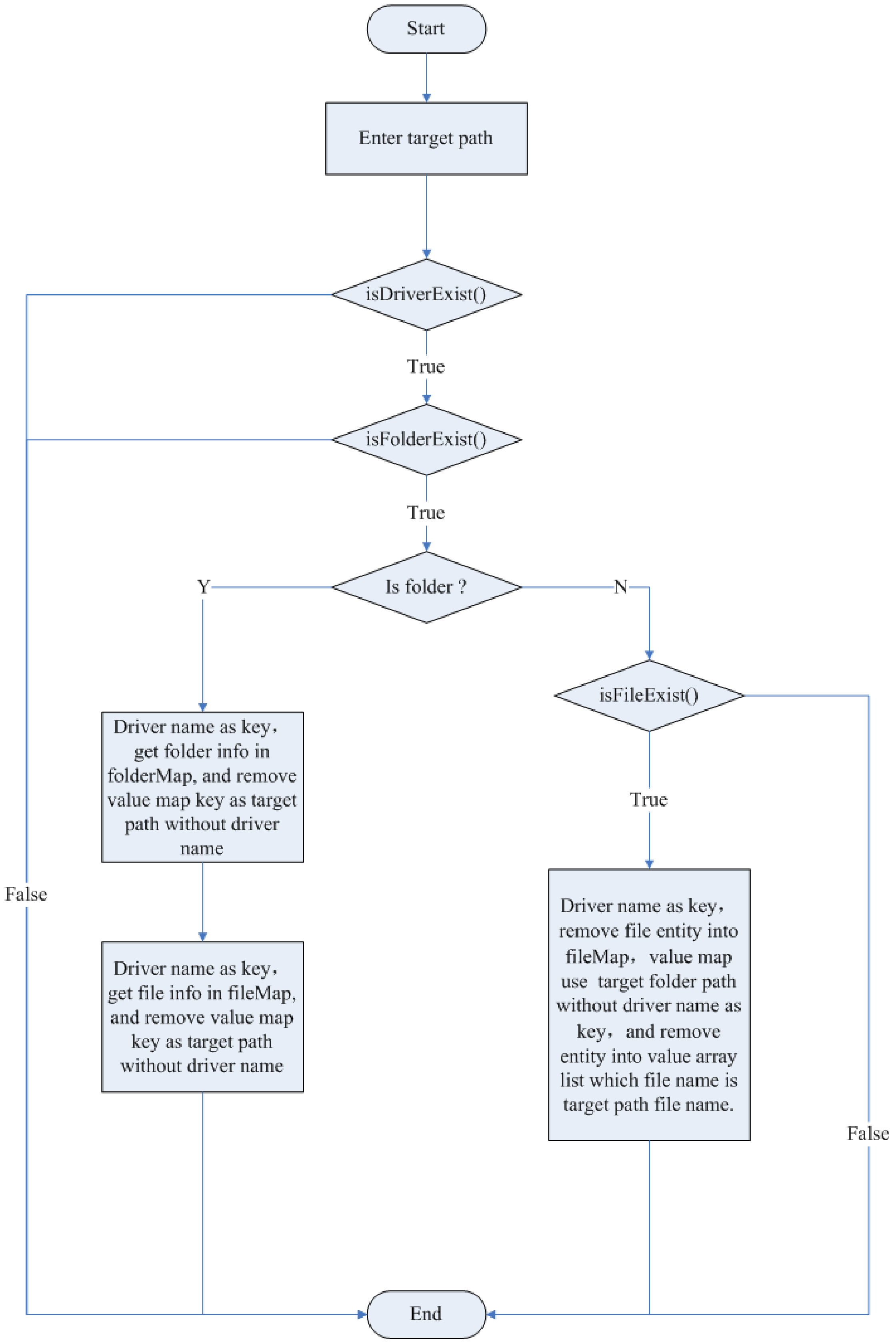
Change TXT file（editTxtFile）


Start
Enter txt content， and save
isDriverExist()
False
True
isFolderExist()
True
False
isFileExist()
True
Get TextFileEntity in fileMap， and edit content
Driver name as key， value map use folder path without driver name as key， and remove the file which file name is txt file name in array list.
Driver name as key， put txt entity into fileMap， value map use folder path without driver name as key， and put entity into value array list.
End

Move a folder or file（moveFile）


Start
Enter original path
isDriverExist()
True
isFolderExist()
True
Is folder ?
Y
N
Enter target path
isFileExist()
True
Driver name as key，get file entity in fileMap，and remove this file entity in arraylist.
False
Driver name as key，get folder info in folderMap, and rename value map key as target path without driver name
Driver name as key，put entity into fileMap，value map use target folder path without driver name as key，and put entity into value array list.
False
End

Delete a folder or file（deleteFile）


Start
Enter target path
isDriverExist()
True
isFolderExist()
True
Is folder ?
Y
N
isFileExist()
Driver name as key，get folder info in folderMap, and remove value map key as target path without driver name
True
False
Driver name as key，get file info in fileMap, and remove value map key as target path without driver name
Driver name as key，remove file entity into fileMap，value map use target folder path without driver name as key，and remove entity into value array list which file name is target path file name.
False
End