Lab09 – Files and Strings

Enrique Saracho Felix

100406980

CPSC 1150

23/07/2023

## Program StudentScores

**File name:** StudentScores.java

**Purpose:** Keeps track of student records by giving the user three options, to create a new file, search in the active file, and add more records.

**Packages:** javax.swing.JOptionPane

java.io.File

java.io.FileWriter

java.io.PrintWriter

java.util.Scanner

**Input:** An integer number between 1 and 4 to pick the main options.

A string containing a new file name if option 1 is selected.

A string containing a key to look for in the file if option 2 is picked.

Two strings, one for the student’s name, and another for the score, if option for option 3.

**Output:** The output consists of many messages answering the options picked by the user. And files created or modified in the same directory as the program.

**Pseudocode:**

Algorithm *StudentScores*

START

set activeFile to global File object of name “default\_scores.txt”

(**main**)

set option as integer

do {

read option

if option is 1

call **createFile**()

if option is 2

call **searchInFile**()

if option is 3

call **addRecord**()

if option is 4

print “Exiting program”

else

print “Invalid input”

} while option is not 4

(**createFile**)

set flag as false

do {

read newName

activeFile = new File object of name newName

if activeFile exists

print “File already exists”

else {

if ( create new file activeFile ) returns true {

print “File created successfully”

flag = true

} else {

print “Failed to create file”

}

} while flag is false

(**searchInFile**)

set and read key as string

if the program is not able to read activeFile

print “Permission to read from file denied”

set result and student as empty strings

while there is text in activeFile {

set token as the next piece of text

if **isName**(token) returns true

student += token + “ “

if **isScore**(token) returns true {

student += token

if **hasKey**(student, key) returns true

result += student + new line

student = “”

}

}

print result

(**isName**, parameter: token(string))

for each character in token {

if (character is between ‘A’ and ‘Z’) or (character is between ‘a’ and ‘z’)

return true

}

return false

(**isScore**, parameter: token(string))

if token[0] is a digit

return true

else

return false

(**hasKey**, parameters: student(string), key(string))

set studentLower = student(uppercase)

set keyLower = key(uppercase)

if studentLower contains keyLower

return true

else

return false

(**addRecord**)

set name and score as strings

read name

read score

print name + “ “ + score in activeFile

END *StudentScores*

**Test run(s):**

A screenshot of a computer program

Description automatically generatedA screenshot of a computer error

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA screenshot of a computer screen

Description automatically generated

A screenshot of a computer error

Description automatically generatedA screenshot of a computer program

Description automatically generated