**Computer Science 111**

Computer Science with Java I Fall, 2016

Lab Report – Week 8 - String Assignment

Abraham Schultz CSI 111 900 Fall 2016

**Assignment Analysis and Design**

The problem to be solved with this assignment was that of string manipulation. The program must ask for and receive an input in the form of a string. This String is to be a password that must be checked against various parameters. Before the password is checked against the parameters it must be entered twice to confirm the password. The Output that the program is to print out will be any errors with the entered password if they exist. If not, then the program displays a success message. There are 4 methods my main method, a get password method, a confirm password method, and password is valid method. I wrote another version of the program with 10 methods, one to check each password parameter. But in the end, I decided that I liked the one with 4 methods more because it seemed shorter and more concise.

**Here is the pseudocode that I used as a guide.**

Main method

//Declare variables

Boolean passMatch= false

Boolean pass confirm= false

String password= p

Tell user” please enter password with ………. Requirements”

Call method to get password from user

Call method passcheck to check password for errors

Ask to confirm password if no errors, and call method to compare

If pass confirm = false display “ you entered in your password wrong”

If pass confirm = false call passCheck method to get new password

Print out success message once pass match and pass confirm = true

String PassAsk Method static

Ask for password

Declare instance of input.scanner

Password = next.line

Return password

Boolean static PassCheck method( String password)

Boolean passmatch= false

do {

Calls method which ask for password and returns string value

Calls method that checks password length

Calls method to check for alpha characters

Calls method to check for Numeric Characters

Calls method to Check for non alphanumeric characters

Calls method to check for spaces

Calls method to check for ?! at beginning and end of string

Calls method to check for repeating characters

If passmatch = false then say “ please try another password “

While Boolean = false

End loop when boolean passmatch = true

Method to Checklength (String Password)

Boolean length= false

Loop uses.length method to check that length of password is at least 8, boolean length = true if at least 8

Print out error message if false

Return length

Method to check for alpha characters()

Boolean alpha = false

Loop uses .contain() method to check for Alpha and numeric characters , boolean alpha= true if contains characters

Print out error message if false

Return alpha

Method to check for Numeric Characters()

Boolean numa = false

Loop uses .contain() method to check for Alpha and numeric characters , boolean numa= true if contains characters

Print out error message if false

Return numa

Method to Check for non alphanumeric characters()

Boolean char= false

Loop uses .contain() method to check for at least one non alphanumeric number, boolean char = true if contains characters

Print error message if false

Return char

Method to check for spaces

Boolean space = false

Loop uses .contain() String method to check if password contains spaces ,Boolean space = true if no spaces

Print error message if space = false

Return space

Method to check for ?! at beginning and end of string

Boolean beginend = false

Loop uses .endswith and .startswith methods to check if password starts with or ends with !? , Boolean beginEnd = true if doesn't begin or end with ?!

Print error message if false

Return beginend

Method to check for repeating characters()

/\*password program

\*Console I/O dialog for gettting secure password

\*for CSCI 111

\*last edited october 25th 9:23pm

@author Abraham Schultz

\*/

package password;

import java.util.Scanner;

public class Password {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

String Password;

String Confirm;

do {

Password = enterPassword();// call method to enter password

Confirm = confirmPassword(Password); //call method to confirm password

} while (!Password.equals(Confirm));

// loop repeats until password entries match

Boolean Condition = isValid(Password); // once entries match, program checks password against paramaters

// condition is true if password is valid . program will skip lines 29-42 if correct. otherwise it continues until correct

while (!Condition) { // this loop continues to ask for a new password if there is an error detected

System.out.println("The password entered is not valid!");

do {

/\*

\* Another loop within the error checking loop. This makes sure password entries

\* match in the event that on the first attempt entries DID match but password was NOT VALID,

\* this ensures that on the second attempt entries are not checked for validity until they

match.

\*/

Password = enterPassword();// call method to enter password

Confirm = confirmPassword(Password); //call method to confirm password

} while (!Password.equals(Confirm));// if entries match then check password against paramaters

Condition = isValid(Password); // condition is true if password is valid

}

System.out.println("You have entered a good password!!");

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public static String enterPassword() // method for entering password

{

String password;// declare password variable

Scanner input = new Scanner(System.in);//scanner object within method declared

System.out.println("Please enter the password : ");//asks for password

password = input.nextLine();// next input types = password

return password; //return password to main as String

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public static boolean isValid(String password) { // method for checking if password is valid

/\*

This method checks each of the paramaters agaisnt the entered password.

if all variables are entered correctly then all booleans will return

true. it uses a for loop that iterates through each letter

of the password until reaching the last letter.

\*/

//booleans and variables needed to describe each of the paramaters for password

boolean length = true;

boolean digit = false;

boolean letter = false;

boolean nonnumeric = false;

boolean spaces = true;

boolean exclamation = true;

boolean repeating = true;

String[] characters = password.split(""); // here each letter of the password is put in to a string array called characters

String[] symbols = {"!", "#", "@", "$", "%", "^", "&", "\*", "(", " )", "- ", "\_",

"=", "+", "[", "]", ";", ":", "'", ",", "<", ".", ">", "/", "?"};

// string array to use for nonnumeric symbols comparison

if (password.length() < 8) {// if password is to short then boolean length = false

length = false;

System.out.println("Password is to short. The password should be at least 8 characters long.");

// tells user that the password is to short

}

for (int i = 0; i < password.length(); i++) {

//loop that checks each character of password

if (Character.isDigit(password.charAt(i))) {

digit = true;// once it finds a digit set boolean to true

}

if (Character.isLetter(password.charAt(i))) {

letter = true; // once it finds a letter set boolean to true

}

if (Character.isWhitespace(password.charAt(i))) {

spaces = false; // if loop finds a space it sets boolean as false

}

if (password.endsWith("?") || password.startsWith("?")

|| // if password starts or ends with ? or ! then true

password.endsWith("!") || password.startsWith("!")) {

exclamation = false;// once it finds a symbol sets to true

}

for (int n = 0; (n < symbols.length); n++) { // iterate through character array until it finds a matching symbol

if (password.contains(symbols[n])) {// when it finds a matching symbol = true

nonnumeric = true;

} // end if

}

for (int a = 0; a < characters.length - 2; a++) {

// iterate trough the array containg the characters of the password. becasue we only care about three repeating charcaters we stop loop when a < characters.length-2

if (characters[a].equals(characters[a + 1]) && characters[a + 1].equals(characters[a + 2])) {

repeating = false; // if loop finds three repeating characters then set to false

}

}

}

if (!digit) {

System.out.println("the password entered does not contain at least one number digit!");

}

if (!letter) {

System.out.println("the password entered does not contain at least one letter character!");

}

if (!spaces) {

System.out.println("the password entered should not contain any spaces!");

}

if (!exclamation) {

System.out.println("the password entered should not begin or end with a ? or !");

}

if (!nonnumeric) {

System.out.println("the password entered should contain one non alpha-numeric character!");

}

if (!repeating) {

System.out.println("the password entered should not contain three or more repeating characters!");

}

return (length && digit && letter && spaces && exclamation && nonnumeric && repeating);

/\* return all boolean vairables to main if any one is false then value returned is false

if all true then password is correct and value returned is true

\*/

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public static String confirmPassword(String Password1) // method for confirming password. with first password passed as argument

{

String password; // declare password variable to confirm

Scanner input = new Scanner(System.in);

System.out.println("please re-enter the password to confirm : ");

password = input.nextLine();

if (password.matches(Password1)) {

System.out.println("checking that entries match...");

System.out.println("entries match.");

} else {

System.out.println("checking that entries match...");

System.out.println("Entries do not match, please try again!");

}

return password;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

**Assignment Testing**

In testing the program, I had to run it a lot of times with various types of Strings. I tried to just work on getting one part of the assignment at a time to work correctly. Because There are various loops (including loops within loops) I strategically would put lines of code in loops that were causing me issues. This line of code would print out the word test, this helped me see where and what my loops were doing. I also utilized the break line and jump to next line function of netbeans to ensure that my Booleans were returning the correct values.

**Assignment Evaluation**

I did like this project a lot because it allowed me to use everything that I have learned up to this point. In particular, I made use of arrays to store the values of the symbols which should be included in the password. Also, I used an array to contain each letter of the password. I was not sure how many methods I should have used. I kept going back and forth between two versions I had created, one with four methods and one with 10. The hardest part for me was getting my loops to behave the way I wanted them too and to get the program to accomplish exactly what I wanted to do in the correct order. In the end though I was very happy with how my code turned out.