**CSE 222**

**Data Structures**

**Report**

Yavuz Selim İkizler

1901042617

Design Explanation:

Class Design:

I made system1, password1, password2 and username classes.

The username class takes string as parameter , this string holds the username . Get method to access this data .

The password1 class takes a string as a parameter. It keeps the password1 data as a string. There is a get method to access this data. There is also a check\_correct method that checks whether password1 is valid or not. This method checks whether the length of the data is valid and whether it contains a valid number of letters and brackets. This method is called in the login method, which I will talk about later.

The password2 class takes integer password2 as a parameter. It keeps the password2 data as an integer, there is a get method to access the data, and there is also a check\_correct method. This method checks whether the entered password is within the valid number range. This method is tested in the login method.

I made a class for security system named system1. This class takes username password1 and password2 objects as parameters. The methods made for user name and password check are defined as private in this class.They are called in public login method for testing.

Methods:

In the login method, it is determined whether the login is successful or not according to the return values ​​from other methods, and a successful or unsuccessful (with the reason for the error) messages are printed depending on the situation.

The checkIfValidUsername method checks if the username is valid (username must be at least 1 character and contain only letters). It takes string username as a parameter. At first, statement checks if the string is empty, if it is, it writes the appropriate error message and returns false. If not, it checks the starting element of the String to see if the ascii number is not in the letter range.If it is not in the range it returns false. If it is in the range, it removes the starting character from the string using substring(1) and recursively calls it. It continues to call until 1 element remains in the string. When 1 element remains, that element checked if letter or not .If it’s letter return true else return false and print suitable error message.

containsUserNameSpirit takes string username and string password1 as parameters. It stores the characters in the username in the stack. While it takes each element of the stack in the loop. There is 1 for loop in this loop. In this loop, each value of password1 is checked with every value taken from the stack. If the same character is found, true is returned. If not found, an error message is written at the end of the method and false is returned.

isBalancedPassword takes String password1 as a parameter. Then it checks the characters of the string in the for loop. If the character is open bracket, it stores it on the stack. If the character is closed bracket, it first checks whether the stack is empty. If the stack is empty, it writes an error message and returns false. If not, it checks the last element of the stack and whether it is the appropriate bracket. If it matches, the open bracket is removed from the stack and the loop continues. If it does not match, an error message is written and false is returned. The checks are done by calling the match method. At the end of the method, it is checked whether the stack is empty or not. If it is empty, true is returned.In this way, it is checked whether all open brackets are matched.

The isExactDivision method takes password2 size and denominations as parameters. It subtracts each denomination value from password2 in the for loop and calls itself recursively. The function value is combined with the returned variable using bitwise OR. In this way, if any return value is true, the returned variable will be true. In the base state, if password2 is equal to 0, there are combinations that give password2 and true is returned. If password2 is a negative value, no result was found, false is returned.

The isPalindromePossible method is a recursive method that calculates whether a palindrome can be created from the given string password1. It uses some helper methods to do this. The remove\_brackets method is called at the beginning of the method.It removes the brackets from password1 to be used.In recursion part of the isPalindromePossible method the characters in strings swapped using swap method .That methods takes two index value as parametre then swaps the characters at that index and returns the new string. In the for loop, starting from the index value, swapping is performed, the value in the parameter is increased by 1, and the method is called again, in that way, all possible permutations of the input string and checks if any of them form a palindrome.To do that. The ispalindrome method is checked in the base case. It checks whether the given string is a palindrome.It is recursive method takes password1 as parametre check if the variable at start and end index are same ,if it’s not returns false means that the string is not polindrome otherwise extract first and last character of the string and cals the method again until there is no character or one character in the string if its happen returns true that the string is polindrome

Time Complexity Analysis:

CheckIfValidUserName:

recursive call takes O(n) + O(n-1) .

substring function takes O(n) or O(1) regarding to the java version.

Overall time complexity : O(n^2) | O(n).

boolean containsUserNameSpirit(String username, String password1):

pushing string elements to stack in for loop takes O(n) time.

popping an element from stack and assign it to a variable takes O(1) time.

searching string via loop takes O(m) time.

if condition takes O(1) time.

while loop takes O(n) time .

overall time complexity:

T(n)= O(n\*m) + O(n) = O(n\*m)

boolean isBalancedPassword(String password1):

for loop depends on passwrd1’s length takes O(n) time .Inside for loop statements takes O(1) time is mathcing function call takes O(n) time return statements takes O(1) time so overall time complexity is takes O(n) time.

private boolean isPalindromePossible(String password1, int index):

remove\_brackets function takes O(n) time it depends on string size.

Is\_palindrome revursive function O(n) time for function call and substring method takes O(n) or O(1) time regarding the java version so overall is\_palindrome function takes O(n^2) or O(n) time.

Swap function takes O(n) or O(1) time regarding the java version.

IsPalindromePossible revursive call in for loops takes O(n!) time since for each call the for loop repeated n-1 time.

Return takes O(1) time.

Overall time complexity :

T(n) = O(n!) + O(n^2) + O(n) + O(1) = O(n!)

boolean isExactDivision(int password2, int [] denominations,int size):

the time complexity is exponantial depends on denominations number and password2 size

time complexity: O(n^m).

Output and Results:

Denominations:



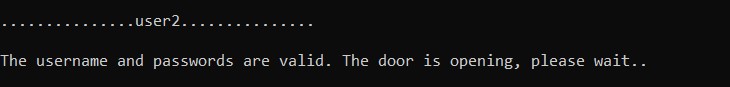


Valid Username, password1, password2:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:



Username length restriction test:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

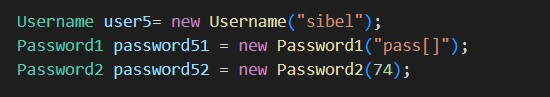
Açıklama otomatik olarak oluşturuldu

Output:

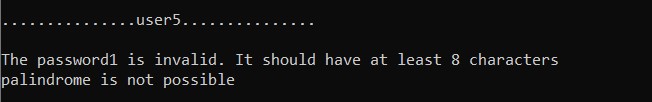
metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Password1 character number checking test (at least 8):



Output:



Password1 bracket number checking test (at least 2 brackets):

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Password1 letter number checking test (at leats 1):

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, siyah, yazı tipi içeren bir resim

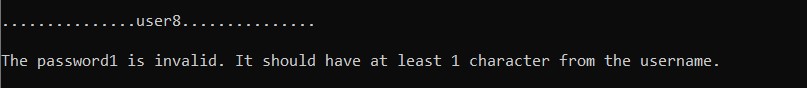
Açıklama otomatik olarak oluşturuldu

Password1 includes at leats 1 character from username test:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:



IsBalancedPassword Test:

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

IsPolindromePossible Test:

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Password2 validation test (between 10 and 10000):

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

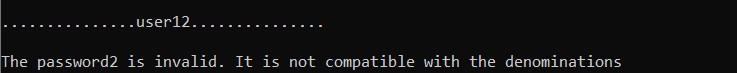
Açıklama otomatik olarak oluşturuldu

IsExactDivision Test:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:



Multiple errors test (all errors can be catch together except character length brackets number and letter numbers fro password1).

metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

IsExactDivison test with different denoinations.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output:

metin, ekran görüntüsü, yazı tipi, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Note:

IsExactDivision takes a lot of time to operate if the given password2 is large since it is very ineficient.Also denomination number must be entered on parametre while calling it.

isPalindromePossible method also ineficient .For large length of string the execution may takes a lot of time.