Shenabeth Jenkins

Professor Alexander

CMSC 204

2/3/2020

**UML Class Diagram**

|  |
| --- |
| **PasswordCheckerUtility** |
|  |
| + PasswordCheckerUtility()  + isValidPassword(passwordString:String): static Boolean  + isWeakPassword(passwordString:String): static Boolean  + invalidPasswords(passwords: String array): static ArrayList |

**Pseudocode**

1. Start
2. User inputs a password (or loads a file)
3. Check password validation:
   1. Check for at least 6 characters
      1. Loop through the string and add 1 to the counter every time.
      2. If the counter is lower than 6, it is not a valid password.
         1. Show message: “The password must be at least 6 characters long”
   2. Check password strength (similar to **a**)
      1. Loop through the string and add 1 to the counter every time.
      2. If the counter is 10 or higher, it is a strong message
      3. If the counter is between 6 and 10, it is an OK password
         1. Show message: “Password is OK, but weak”
   3. Check for at least 1 numeric character
      1. Loop through the string
      2. If the character is between 0 and 9, add 1 to the counter
      3. If the counter is still 0 at the end of the loop, there was no numeric character
         1. Show message: “The password must contain at least one digit”
   4. Check for at least 1 uppercase alphabetic character
      1. Loop through the string
      2. If the character is between **A** and **Z**, add 1 to the counter (ascii values)
      3. If the counter is still 0 at the end of the loop, there was no uppercase letter
         1. Show message: “Password must contain at least one uppercase alphabetic character”
   5. Check for at least 1 lowercase alphabetic character
      1. Loop through the string
      2. If the character is between **a** and **z**, add 1 to the counter (ascii values)
      3. If the counter is still 0 at the end of the loop, there was no lowercase letter
         1. Show message: “Password must contain at least one lowercase alphabetic character”
   6. Check for no more than 2 of the same character in a sequence
      1. Loop through the string and set the counter to the 1st character of the string
      2. If the counter character equals the next character and the one after, there are 3 of the same characters in sequence
         1. Show message: “The password cannot contain more than two of the same character in sequence”
      3. If there are less than 2 characters left to compare, the password is still valid
4. Check if passwords match
   1. Loop through all of the characters of string 1 and string 2
   2. If there is a difference in the characters compared, the passwords are different
      1. Show message: “Passwords do not match”
5. Password is valid if it passes all of the checks
   1. Show message: “Password is valid”
6. End