(AS1\_Passwords)

1. Pseudocode\_ PasswordCheckerUtility:

**isValidPassword(String password):**

* Convert password to an array of characters.
* Initialize boolean flags:
* containsNumber: False if no digits, True if a digit is found.
* containsUpperAlpha: False if no uppercase letters, True if one is found.
* containsLowerAlpha: False if no lowercase letters, True if one is found.
* containsSpecialCharacter: False if no special symbols, True if one is found.
* Check length:
* If password.length < 6, throw LengthException.
* Check for digits:
* Loop through passwordArray.
* If a digit is found, set containsNumber to True.
* If containsNumber is False, throw NoDigitException.
* Check for uppercase letters:
* Loop through passwordArray.
* If an uppercase letter is found, set containsUpperAlpha to True.
* If containsUpperAlpha is False, throw NoUpperAlphaException.
* Check for lowercase letters:
* Loop through passwordArray.
* If a lowercase letter is found, set containsLowerAlpha to True.
* If containsLowerAlpha is False, throw NoLowerAlphaException.
* Check for special characters:
* Loop through passwordArray.
* If a special character is found, set containsSpecialCharacter to True.
* If containsSpecialCharacter is False, throw NoSpecialSymbolException.
* Check for repeated sequences:
* Initialize foundInSequence = 0 and lastCharacter = ''.
* Loop through passwordArray.
* If current character equals lastCharacter, increment foundInSequence.
* If foundInSequence >= 3, throw InvalidSequenceException.
* Otherwise, reset foundInSequence and update lastCharacter.
* If all conditions are met, return True.

**isWeakPassword(String password):**

* If password.length ≥ 10, return True.
* Else, throw LengthException with the message "Password is less than 10 characters".

**getInvalidPasswords(ArrayList<String> passwords):**

* Create an empty list invalidPasswords.
* Loop through each password in passwords:
* Try to validate the password using isValidPassword.
* Catch any exception:
* Add the password and exception message to invalidPasswords.
* Return invalidPasswords.

1. UML Class Diagram for created calsses:

|  |  |  |
| --- | --- | --- |
|  | PasswordCheckerUtility |  |
| Methods | + isValidPassword(password: String)  + isWeakPassword(password: String)  + getInvalidPasswords(passwords: ArrayList<String>) : ArrayList<String> |
| Attributes | - passwordArray: char[]  - containsNumber: Boolean  - containsUpperAlpha: Boolean  - containsLowerAlpha: Boolean  - containsSpecialCharacter: Boolean |
| Exceptions | + LengthException  + NoDigitException  + NoUpperAlphaException  + NoLowerAlphaException  + NoSpecialSymbolException  + InvalidSequenceException |

1. Test Cases:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case# | Test | Expected Result | Actual Result | Pass? | Reflection |
| Case 1 | Password: a2cDe | Throws LengthException | Throws LengthException | yes |  |
| Case 2 | Password: 334455BB# | Throws NoLowerAlphaException | Throws NoLowerAlphaException | yes |  |
| Case 3 | Password: Im2cool4U# | Passes all checks.  considered weak because the length is exactly 10 | Valid password passes all | yes | Only show valid message |

Case 1:

A screenshot of a computer screen

Description automatically generated

Case 2:

A screenshot of a computer

Description automatically generated

Case 3:

A screenshot of a computer screen

Description automatically generated