

/\*

Final Design (Psuedocode)

/\*

### **CLASS PasswordCheckerUtility**

METHOD isValidPassword(pass: STRING) RETURNS BOOLEAN

IF length of pass < 6 THEN

    THROW LengthException("The password must be at least 6 characters long")

END IF

SET uppercaseFound = FALSE

FOR each character c IN pass

    IF c is uppercase THEN

        SET uppercaseFound = TRUE

        BREAK

    END IF

END FOR

IF NOT uppercaseFound THEN

    THROW NoUpperAlphaException("The password must contain at least one uppercase alphabetic character")

END IF

SET lowercaseFound = FALSE

FOR each character c IN pass

    IF c is lowercase THEN

        SET lowercaseFound = TRUE

        BREAK

    END IF

END FOR

IF NOT lowercaseFound THEN

    THROW NoLowerAlphaException("The password must contain at least one lowercase alphabetic character")

END IF

SET digitFound = FALSE

FOR each character c IN pass

    IF c is a digit THEN

        SET digitFound = TRUE

        BREAK

    END IF

END FOR

IF NOT digitFound THEN

    THROW NoDigitException("The password must contain at least one digit")

END IF

SET specialCharFound = FALSE

FOR each character c IN pass

    IF c is NOT a letter OR digit THEN

```

        SET specialCharFound = TRUE
        BREAK
    END IF
END FOR
IF NOT specialCharFound THEN
    THROW NoSpecialCharacterException("The password must contain at least one special
character")
END IF

FOR i FROM 0 TO length of pass - 3
    IF pass[i] == pass[i + 1] AND pass[i + 1] == pass[i + 2] THEN
        THROW InvalidSequenceException("The password cannot contain more than two of the same
character in sequence")
    END IF
END FOR

RETURN TRUE
END METHOD

METHOD isWeakPassword(pass: STRING) RETURNS BOOLEAN
    TRY
        CALL isValidPassword(pass)
    CATCH Exception e
        RETURN FALSE
    END TRY

    IF length of pass BETWEEN 6 AND 9 THEN
        THROW WeakPasswordException("The password is OK but weak - it contains fewer than 10
characters.")
    END IF

    RETURN FALSE
END METHOD

METHOD getInvalidPasswords(passwords: ARRAYLIST of STRING) RETURNS ARRAYLIST of
STRING
    DECLARE invalidPasswords AS new ArrayList of STRING

    FOR EACH pass IN passwords
        TRY
            CALL isValidPassword(pass)
        CATCH Exception e
            ADD pass + " " + e.getMessage() TO invalidPasswords
        END TRY
    END FOR

    RETURN invalidPasswords
END METHOD

```

```
METHOD comparePasswordsWithReturn(a: STRING, b: STRING) RETURNS BOOLEAN
  IF a EQUALS b THEN
    RETURN TRUE
  ELSE
    THROW UnmatchedException("Passwords do not match!")
  END IF
END METHOD

END CLASS
```

### **EXCEPTION CLASSES FORMAT**

```
public class blankException extends Exception {
  public blankException(String message) {
    super(message);
  }
}
```

Replace blank exception with LengthException, NoUpperAlphaException, NoLowerAlphaException, NoDigitException, NoSpecialCharacterException, InvalidSequenceException, WeakPasswordException, UnmatchedException