GENERATE/VALIDATE USERNAME PROGRAM

Software Development
Continuous Assessment

- Semester 1, 2022/23 –

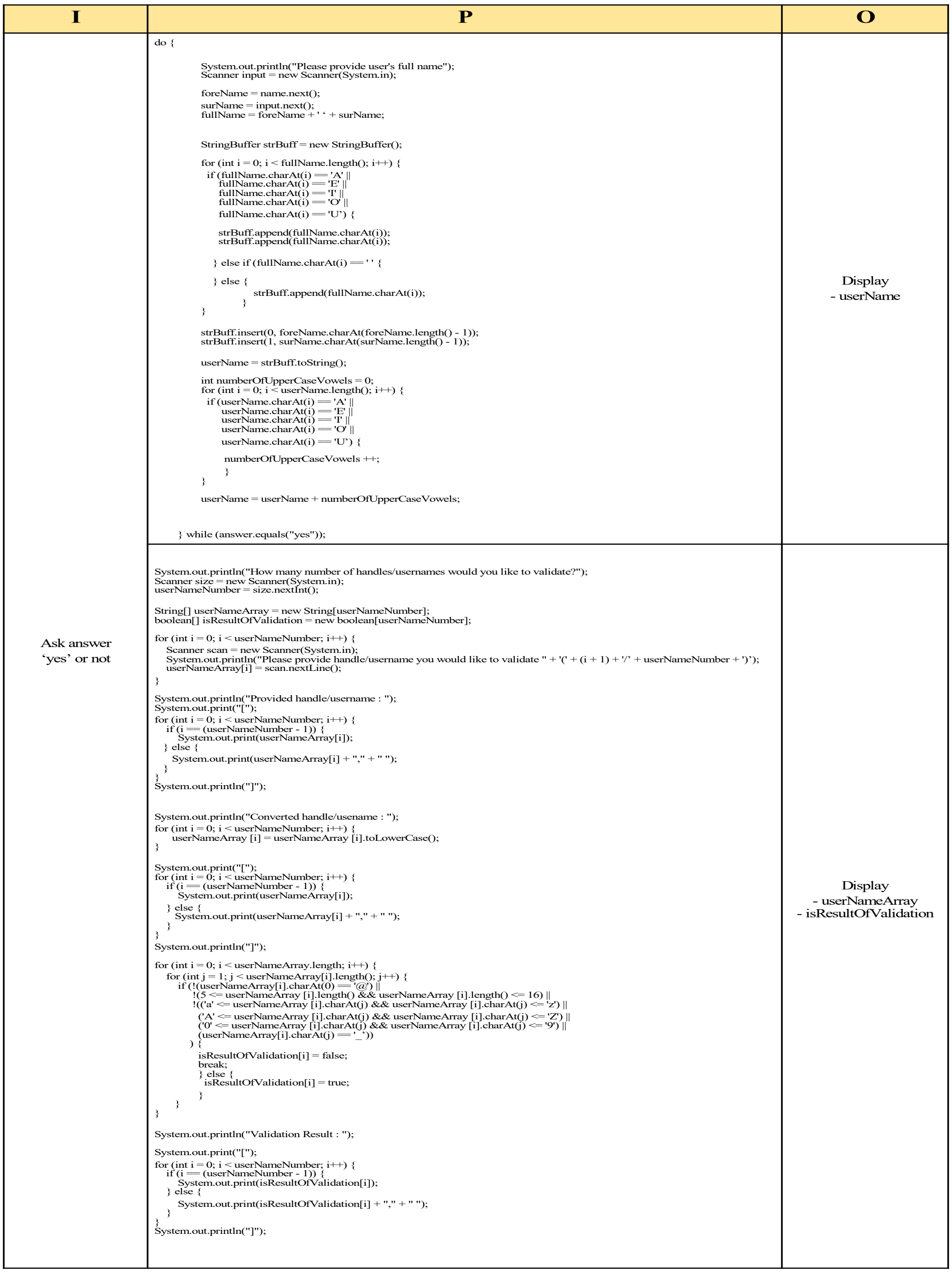
Submitted by

DONGHYEOK LEE (21234175)

Table of Contents

1. IPO (Input-Process-Output) Diagram

- The main input is the user's answer (yes or not). Regardless of the user’s input, username generator is started. So, ‘Do-while loop’ is used for that.
- Once the username generator is finished, the program asks to user if the user would like to generate another username or not. If user inputs ‘yes’, the program will be started again. When the user enters anything else than “yes”, the program is finished, and validation program is started.
- This IPO diagram shows how the code is mapped overall. Design about instantiable classes is explained in the class diagram below.

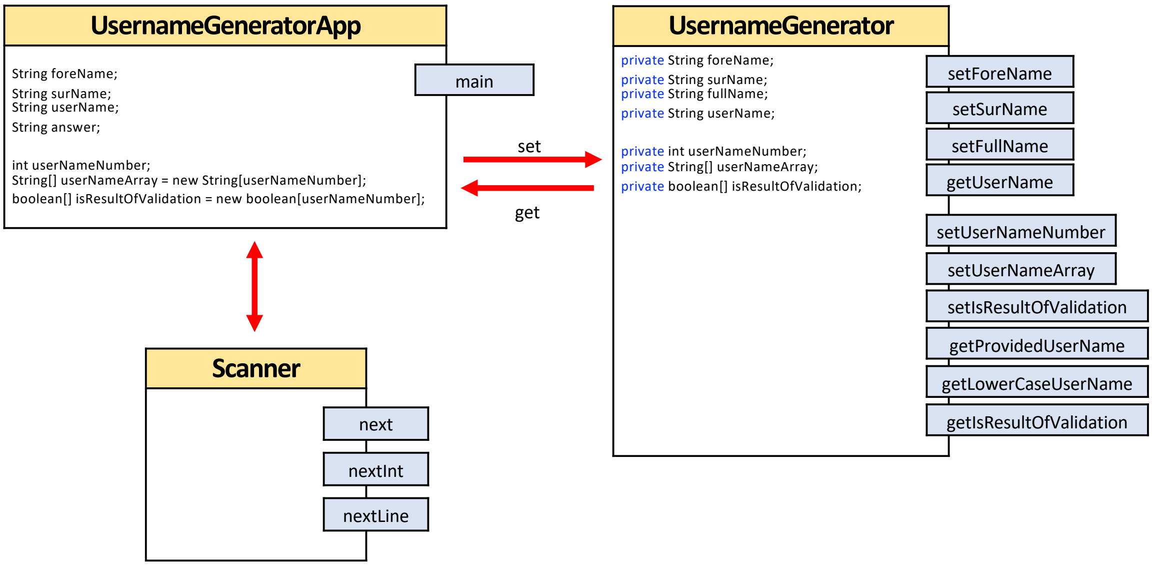


```
do {

        System.out.println("Please provide user's full name");
        Scanner input = new Scanner(System.in);

        foreName = name.next();
        surName = input.next();
        fullName = foreName + ' ‘ + surName;


        StringBuffer strBuff = new StringBuffer();

        for (int i = 0; i < fullName.length(); i++) {
          if (fullName.charAt(i) == 'A' ||
             fullName.charAt(i) == 'E' ||
             fullName.charAt(i) == 'I' ||
             fullName.charAt(i) == 'O' ||
             fullName.charAt(i) == 'U’) {

             strBuff.append(fullName.charAt(i));
             strBuff.append(fullName.charAt(i));

           } else if (fullName.charAt(i) == ' ' {

           } else {
                 strBuff.append(fullName.charAt(i));
               }
        }

        strBuff.insert(0, foreName.charAt(foreName.length() - 1));
        strBuff.insert(1, surName.charAt(surName.length() - 1));

        userName = strBuff.toString();

        int numberOfUpperCaseVowels = 0;
        for (int i = 0; i < userName.length(); i++) {
          if (userName.charAt(i) == 'A' ||
             userName.charAt(i) == 'E' ||
             userName.charAt(i) == 'I' ||
             userName.charAt(i) == 'O' ||
             userName.charAt(i) == 'U’) {

             numberOfUpperCaseVowels ++;
             }
        }

        userName = userName + numberOfUpperCaseVowels;


    } while (answer.equals("yes"));
```

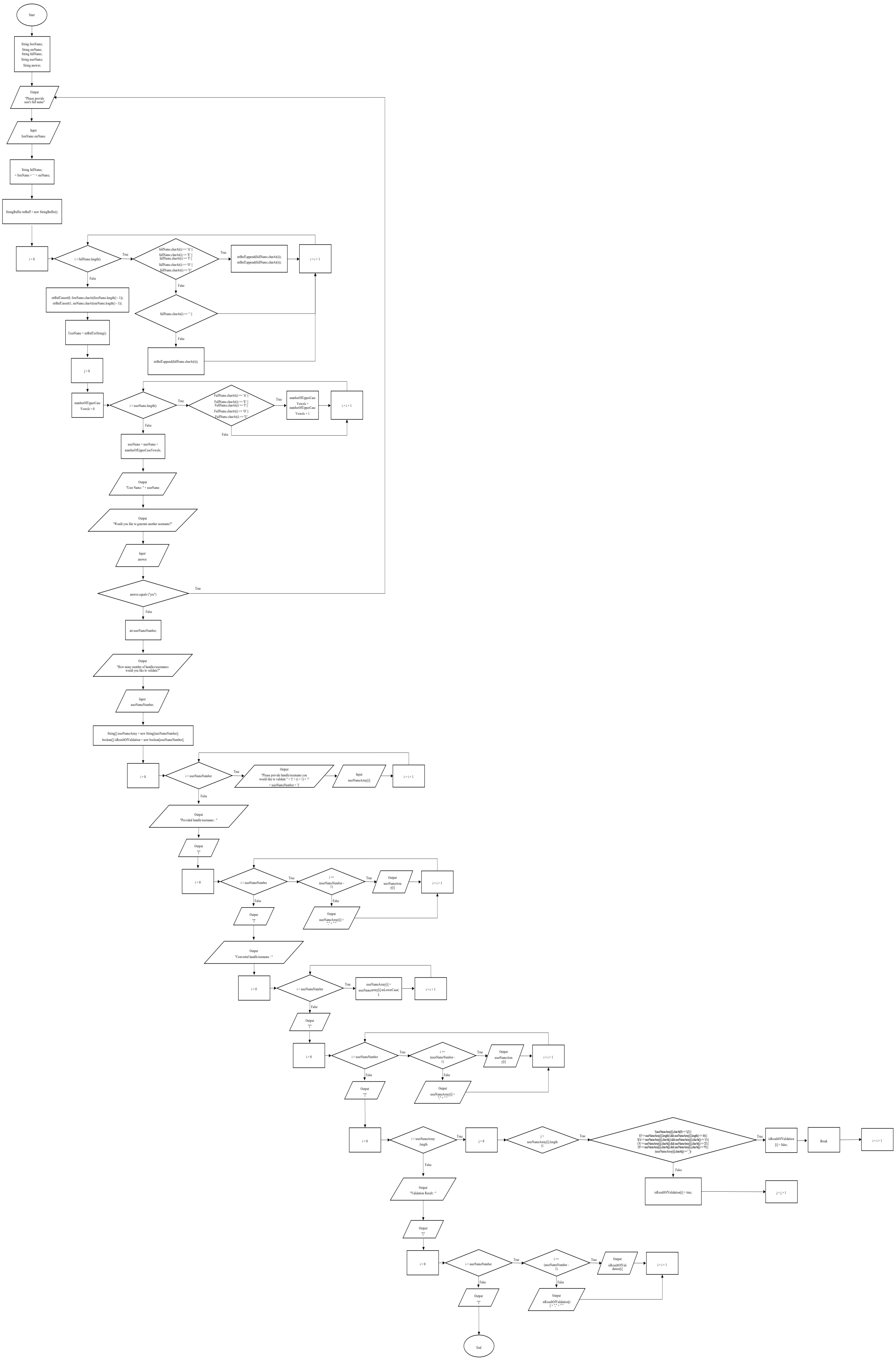
```
System.out.println("How many number of handles/usernames would you like to validate?");
Scanner size = new Scanner(System.in);
userNameNumber = size.nextInt();

String[] userNameArray = new String[userNameNumber];
boolean[] isResultOfValidation = new boolean[userNameNumber];

for (int i = 0; i < userNameNumber; i++) {
  Scanner scan = new Scanner(System.in);
  System.out.println("Please provide handle/username you would like to validate " + '(' + (i + 1) + '/’ + userNameNumber + ')’);
  userNameArray[i] = scan.nextLine();
}

System.out.println("Provided handle/username : ");
System.out.print("[");
for (int i = 0; i < userNameNumber; i++) {
  if (i == (userNameNumber - 1)) {
    System.out.print(userNameArray[i]);
  } else {
   System.out.print(userNameArray[i] + "," + " ");
  }
}
System.out.println("]");


System.out.println("Converted handle/usename : ");
for (int i = 0; i < userNameNumber; i++) {
   userNameArray [i] = userNameArray [i].toLowerCase();
}

System.out.print("[");
for (int i = 0; i < userNameNumber; i++) {
  if (i == (userNameNumber - 1)) {
    System.out.print(userNameArray[i]);
  } else {
   System.out.print(userNameArray[i] + "," + " ");
  }
}
System.out.println("]");

for (int i = 0; i < userNameArray.length; i++) {
  for (int j = 1; j < userNameArray[i].length(); j++) {
    if (!(userNameArray[i].charAt(0) == '@') ||
       !(5 <= userNameArray [i].length() && userNameArray [i].length() <= 16) ||
       !(('a' <= userNameArray [i].charAt(j) && userNameArray [i].charAt(j) <= 'z') ||
        ('A' <= userNameArray [i].charAt(j) && userNameArray [i].charAt(j) <= 'Z') ||
        ('0' <= userNameArray [i].charAt(j) && userNameArray [i].charAt(j) <= '9') ||
        (userNameArray[i].charAt(j) == '_’))
      ) {
        isResultOfValidation[i] = false;
        break;
        } else {
         isResultOfValidation[i] = true;
        }
    }
}

System.out.println("Validation Result : ");

System.out.print("[");
for (int i = 0; i < userNameNumber; i++) {
  if (i == (userNameNumber - 1)) {
    System.out.print(isResultOfValidation[i]);
  } else {
    System.out.print(isResultOfValidation[i] + "," + " ");
  }
}
System.out.println("]");
```

2. Class Diagram

- This class diagram shows how 2 java files interact on each other.
- Between ‘UsernameGeneratorApp’ class and ‘UsernameGenerator’ class, all essential data are conveyed by Set and Get method.
- For reading user input from the keyboard, Scanner class is used.
- This program uses 2 Java files. In ‘UsernameGenerator’ class, there are 2 methods. One method is for generating username and the other method is for validating handles/usename that user would like to check.



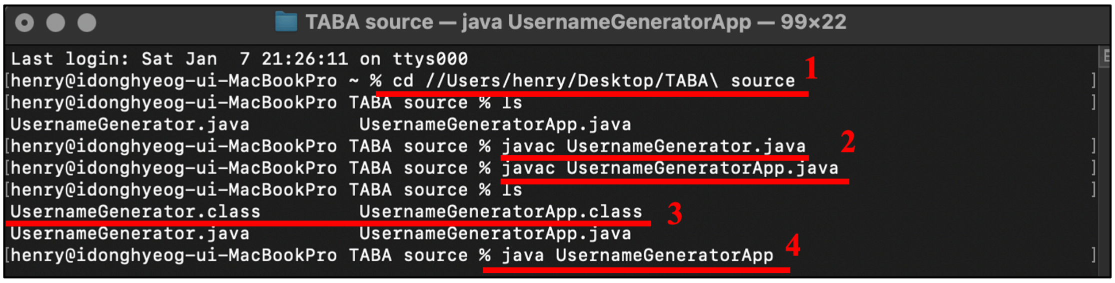
3. Flow chart


Start
String foreName;
String surName;
String fullName;
String userName;
String answer;
Output
"Please provide user's full name"
Input
foreName surName
String fullName;
= foreName + ' ' + surName;
StringBuffer strBuff = new StringBuffer();
i = 0
i < fullName.length()
True
fullName.charAt(i) == 'A' ||
fullName.charAt(i) == 'E' ||
fullName.charAt(i) == 'I' ||
fullName.charAt(i) == 'O' ||
fullName.charAt(i) == 'U'
True
strBuff.append(fullName.charAt(i));
strBuff.append(fullName.charAt(i));
i = i + 1
False
fullName.charAt(i) == ' ' ||
False
strBuff.append(fullName.charAt(i));
False
strBuff.insert(0, foreName.charAt(foreName.length() - 1));
strBuff.insert(1, surName.charAt(surName.length() - 1));
UserName = strBuff.toString();
j = 0
numberOfUpperCase
Vowels = 0
i < userName.length()
True
FullName.charAt(i) == 'A' ||
FullName.charAt(i) == 'E' ||
FullName.charAt(i) == 'I' ||
FullName.charAt(i) == 'O' ||
FullName.charAt(i) == 'U'
True
numberOfUpperCase
Vowels =
numberOfUpperCase
Vowels + 1
i = i + 1
False
False
userName = userName +
numberOfUpperCaseVowels;
Output
"User Name: " + userName
Output
"Would you like to generate another username?"
Input
answer
answer.equals ("yes")
True
False
int userNameNumber;
Output
"How many number of handles/usernames
would you like to validate?"
Input
userNameNumber;
String[] userNameArray = new String[userNameNumber];
boolean[] isResultOfValidation = new boolean[userNameNumber];
i = 0
i < userNameNumber
True
Output
"Please provide handle/username you
would like to validate " + '(' + (i + 1) + '/'
+ userNameNumber + ')'
Input
userNameArray[i]
i = i + 1
False
Output
"Provided handle/username : "
Output
"["
i = 0
i < userNameNumber
True
i ==
(userNameNumber -
1)
True
Output
userNameArra
y[i]
i = i + 1
False
Output
userNameArray[i] +
"," + " "
False
Output
"]"
Output
"Converted handle/usename : "
i = 0
i < userNameNumber
True
userNameArray[i] =
userNameArray[i].toLowerCase(
);
i = i + 1
False
Output
"["
i = 0
i < userNameNumber
True
i ==
(userNameNumber -
1)
True
Output
userNameArra
y[i]
i = i + 1
False
Output
userNameArray[i] +
"," + " "
False
Output
"]"
i = 0
i < userNameArray
.length
True
j = 0
j <
userNameArray[i].length
()
True
!(userNameArray[i].charAt(0) == '@') ||
!(5 <= userNameArray[i].length() && userNameArray[i].length() <= 16) ||
!(('a' <= userNameArray[i].charAt(j) && userNameArray[i].charAt(j) <= 'z') ||
('A' <= userNameArray[i].charAt(j) && userNameArray[i].charAt(j) <= 'Z') ||
('0' <= userNameArray[i].charAt(j) && userNameArray[i].charAt(j) <= '9') ||
(userNameArray[i].charAt(j) == '_'))
True
isResultOfValidation
[i] = false;
Break
i = i + 1
False
isResultOfValidation[i] = true;
j = j + 1
False
Output
"Validation Result : "
Output
"["
i = 0
i < userNameNumber
True
i ==
(userNameNumber -
1)
True
Output
isResultOfVali
dation[i]
i = i + 1
False
Output
isResultOfValidation[i
] + "," + " "
False
Output
"]"
End

4. Screenshots of the application's screens/output

4.1. Run java file by terminal.

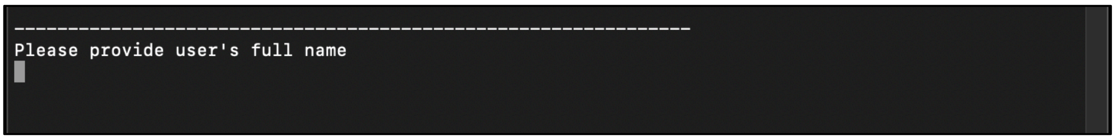
1) After opening terminal, write the folder address where java files are stored.
2) Compile java files by writing javac (java file name).
3) Class files are created.
4) Run 'UsernameGeneratorApp' java file.



```
Last login: Sat Jan  7 21:26:11 on ttys000
[henry@idonghyeog-ui-MacBookPro ~ % cd //Users/henry/Desktop/TABA\ source
[henry@idonghyeog-ui-MacBookPro TABA source % ls
UsernameGenerator.java          UsernameGeneratorApp.java
[henry@idonghyeog-ui-MacBookPro TABA source % javac UsernameGenerator.java
[henry@idonghyeog-ui-MacBookPro TABA source % javac UsernameGeneratorApp.java
[henry@idonghyeog-ui-MacBookPro TABA source % ls
UsernameGenerator.class         UsernameGeneratorApp.class
UsernameGenerator.java          UsernameGeneratorApp.java
[henry@idonghyeog-ui-MacBookPro TABA source % java UsernameGeneratorApp
```

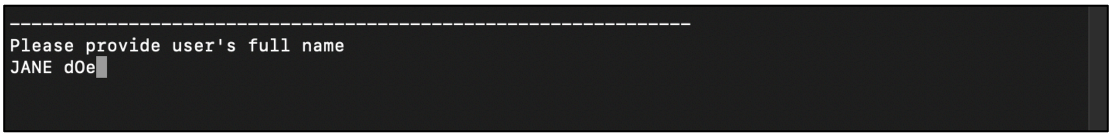
4.2. The output of an application when the application starts .

1) The terminal shows a message and request user to input 'user's full name' like below.



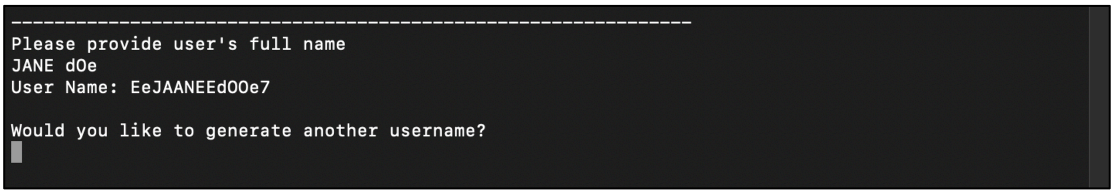
```
---------------------------------------------------------------
Please provide user's full name
```

2) User can input 'user's full name' (forename surname).



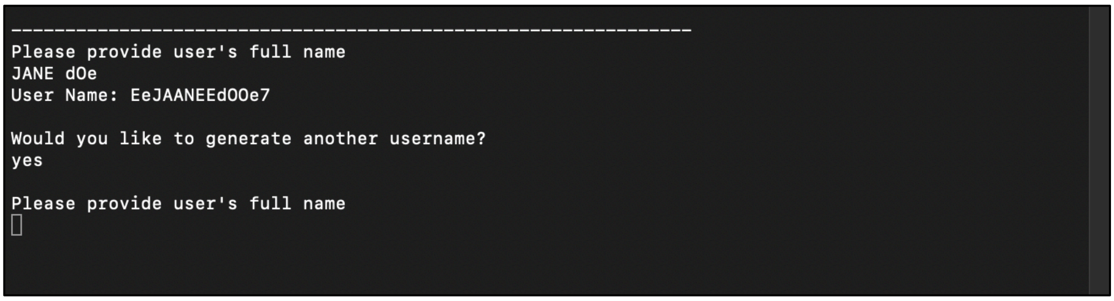
```
---------------------------------------------------------------
Please provide user's full name
JANE dOe
```

3) The terminal shows 'Username' corresponding with a program's logic and ask to user if user would like to generate another username or not.



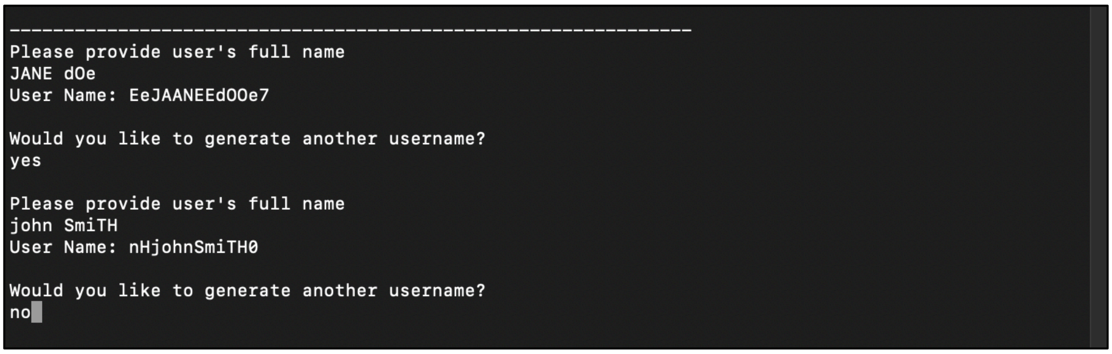
```
---------------------------------------------------------------
Please provide user's full name
JANE dOe
User Name: EeJAANEEdOOe7

Would you like to generate another username?
```

4) User can input ‘yes’ or ‘no’(or whatever except ‘yes’). In case of inputting ‘yes’, the terminal again will show message and request user to input ‘user’s full name’ like below.



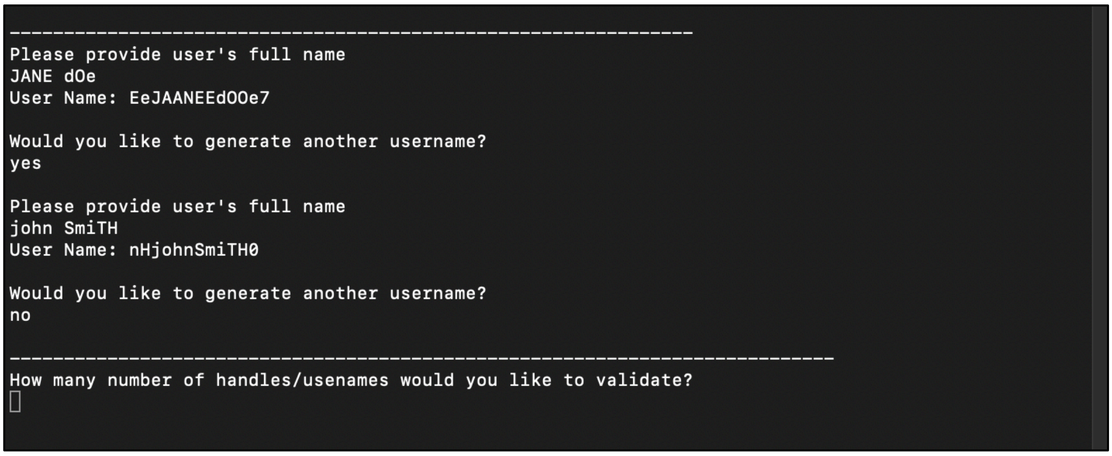
```
---------------------------------------------------------------
Please provide user's full name
JANE dOe
User Name: EeJAANEEdOOe7

Would you like to generate another username?
yes

Please provide user's full name
```

5) User can input another ‘user’s full name’ (forename surname) and get username. In case of inputting ‘no’(or whatever except ‘yes’). A username generator program will be finished.



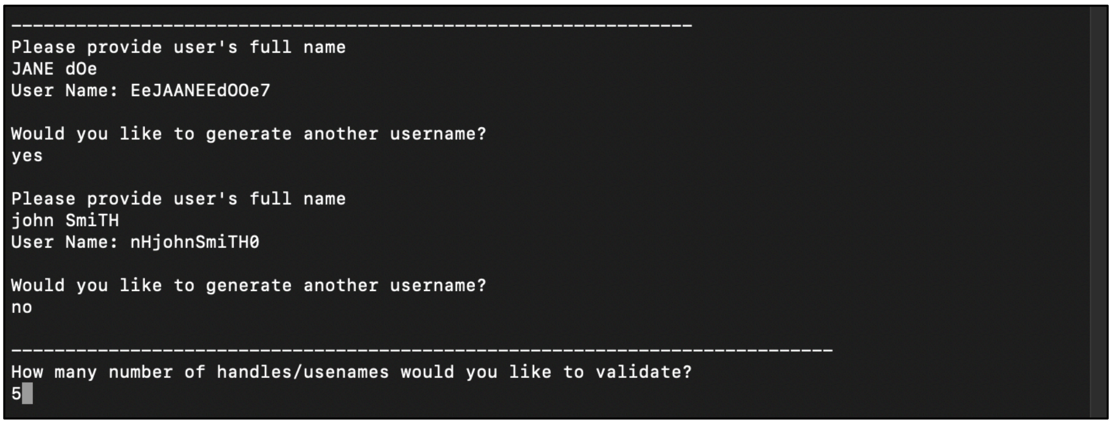
```
---------------------------------------------------------------
Please provide user's full name
JANE dOe
User Name: EeJAANEEdOOe7

Would you like to generate another username?
yes

Please provide user's full name
john SmiTH
User Name: nHjohnSmiTH0

Would you like to generate another username?
no
```

6) As soon as the username generator program is finished, a validation program is started. The terminal shows a message how many number of handles/usernames user want to validate.



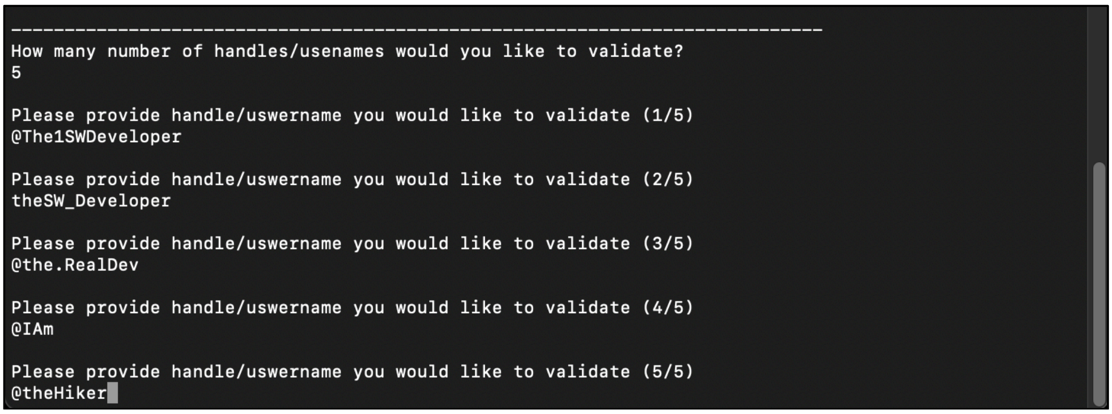
```
---------------------------------------------------------------
Please provide user's full name
JANE dOe
User Name: EeJAANEEdOOe7

Would you like to generate another username?
yes

Please provide user's full name
john SmiTH
User Name: nHjohnSmiTH0

Would you like to generate another username?
no

----------------------------------------------------------------------------
How many number of handles/usenames would you like to validate?
```

7) The user can input the number the user would like to input.



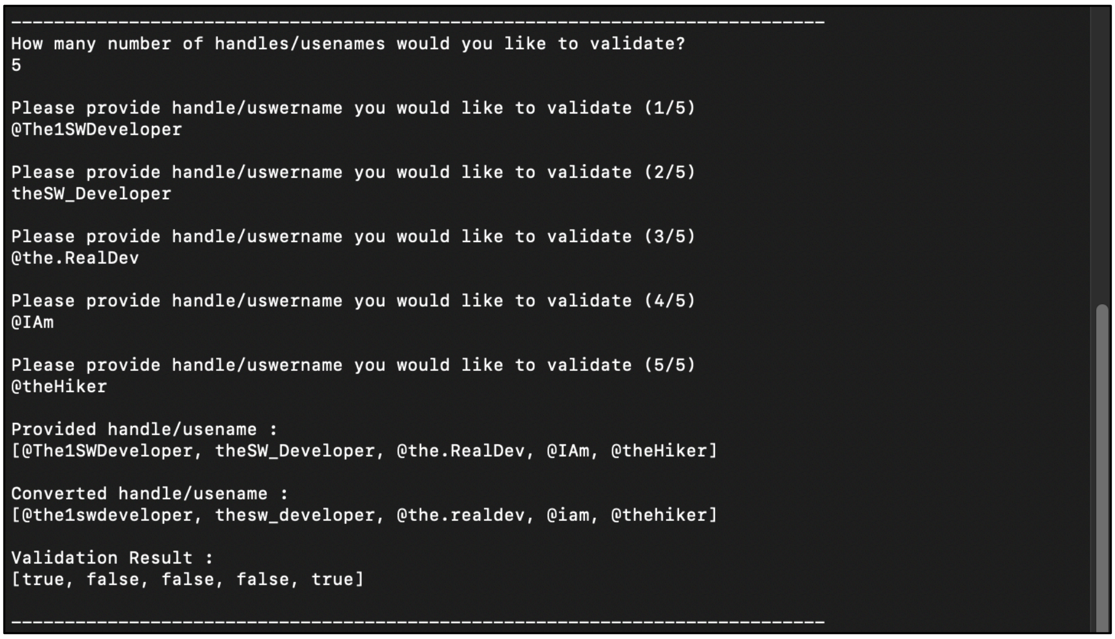
```
---------------------------------------------------------------
Please provide user's full name
JANE dOe
User Name: EeJAANEEdOOe7

Would you like to generate another username?
yes

Please provide user's full name
john SmiTH
User Name: nHjohnSmiTH0

Would you like to generate another username?
no

----------------------------------------------------------------------------
How many number of handles/usenames would you like to validate?
5
```

8) In case of inputting 5, the user can input the handle/username 5 times.



```
---------------------------------------------------------------------------
How many number of handles/usenames would you like to validate?
5

Please provide handle/uswername you would like to validate (1/5)
@The1SWDeveloper

Please provide handle/uswername you would like to validate (2/5)
theSW_Developer

Please provide handle/uswername you would like to validate (3/5)
@the.RealDev

Please provide handle/uswername you would like to validate (4/5)
@IAm

Please provide handle/uswername you would like to validate (5/5)
@theHiker
```

9) When it’s complete to input 5 handles/usernames, the terminal shows some results (Provided handle/username, Converted handle/username (Uppercase into Lowercase), and validation Result) like below.



```
----------------------------------------------------------------------------
How many number of handles/usenames would you like to validate?
5

Please provide handle/uswername you would like to validate (1/5)
@The1SWDeveloper

Please provide handle/uswername you would like to validate (2/5)
theSW_Developer

Please provide handle/uswername you would like to validate (3/5)
@the.RealDev

Please provide handle/uswername you would like to validate (4/5)
@IAm

Please provide handle/uswername you would like to validate (5/5)
@theHiker

Provided handle/usename :
[@The1SWDeveloper, theSW_Developer, @the.RealDev, @IAm, @theHiker]

Converted handle/usename :
[@the1swdeveloper, thesw_developer, @the.realdev, @iam, @thehiker]

Validation Result :
[true, false, false, false, true]

----------------------------------------------------------------------------
```

5. Appendix (Source code)

5.1. UsernameGeneratorApp.java

```
/*
 * UsernameGeneratorApp.java
 * @author Donghyeok.Lee
 * 08/01/2023
 */


import java.util.Scanner;

public class UsernameGeneratorApp {
    public static void main(String args[]) {

        // Question 1

        // Declare variables
        String foreName;
        String surName;
        String userName;
        String answer;

        // Declare and create objects
        UsernameGenerator myU = new UsernameGenerator();
        System.out.println();
        System.out.println("----------------------------------------------------------------------------");


        // The while loop started at least one time. So, do-while loop is used.
        do {
            // Input
            System.out.println("Please provide user's full name");
            Scanner name = new Scanner(System.in);

            foreName = name.next();
            myU.setForeName(foreName);

            surName = name.next();
            myU.setSurName(surName);

            myU.setFullName(foreName, surName);

            // Process
            myU.generateUserName();

            // Output
            userName = myU.getUserName();
            System.out.println("UserName: " + userName);
            System.out.println();
            System.out.println("Would you like to generate another username?");
            answer = name.next();
            System.out.println();

        // In the information related to approachs to entering multiple full names in the Word file,
        // it said that only 'yes' can be accepted to repeat while and it's not mentioned about accepting Uppercase.
        // So, '.equals' method is used (Not '.equalsIgnoreCase')
        } while (answer.equals("yes"));
        System.out.println("----------------------------------------------------------------------------");



        // Question 2

        int userNameNumber;

        // Declare variables and Input

        System.out.println("How many number of handles/usernames would you like to validate?");
        Scanner size = new Scanner(System.in);
        userNameNumber = size.nextInt();
        myU.setUserNameNumber(userNameNumber);

        // The sizes of String array and boolean array are determined by userNameNumber
        String[] userNameArray = new String[userNameNumber];
        boolean[] isResultOfValidation = new boolean[userNameNumber];
        myU.setIsResultOfValidation(isResultOfValidation);
        // Through for loop, the userNameArray[i] values are declared
        for (int i = 0; i < userNameNumber; i++) {
            Scanner scan = new Scanner(System.in);
            System.out.println();
            System.out.println("Please provide handle/username you would like to validate " + '(' + (i + 1) + '/'
                    + userNameNumber + ')');
                    userNameArray[i] = scan.nextLine();
        }
        System.out.println();

        // userNameArray is delivered to instantiable class by set method
        // and serNameArray(Before converting) maded with certain format is got by get method
        System.out.println("Provided handle/username : ");
        myU.setUserNameArray(userNameArray);
        myU.getProvidedUserName();
        System.out.println();

        // userNameArray(After converting) maded with certain format is got by get method
        System.out.println("Converted handle/username : ");
        myU.getLowerCaseUserName();
        System.out.println();


        // Process
        // validationUserName method is used for validating usernames

        myU.validateUserName();


        // Output
        // isResultOfValidation maded with certain format is got by get method

        System.out.println("Validation Result : ");
        myU.getIsResultOfValidation();
        System.out.println();

        System.out.println("————————————————————————————————————————————");
    }
}
```

5.2. UsernameGenerator.java

```java
/*
 * UsernameGenerator.java
 * @author Donghyeok.Lee
 * 08/01/2023
 */

public class UsernameGenerator {

    // Declare variables (Qestion 1)
    private String foreName;
    private String surName;
    private String fullName;
    private String userName;

    // Declare variables (Qestion 2)
    private int userNameNumber;
    private String[] userNameArray;
    private boolean[] isResultOfValidation;

    // Constructor
    public UsernameGenerator() {
        foreName = "";
        surName = "";
        fullName = "";
        userName = "";

        userNameNumber = 0;
        userNameArray = new String[userNameNumber];
        isResultOfValidation = new boolean[userNameNumber];
    }

    // Method1 (Qestion 1) : This method is for generating username corresponding
    // with rules.
    public void generateUserName() {

        // Stringbuffer object is made to make username
        StringBuffer strBuff = new StringBuffer();

        // Through for loop, the programe checks each letter of the full name
        // and through if, else if, else condition, strBuff is made.
        for (int i = 0; i < fullName.length(); i++) {

            if (fullName.charAt(i) == 'A' ||
                    fullName.charAt(i) == 'E' ||
                    fullName.charAt(i) == 'I' ||
                    fullName.charAt(i) == 'O' ||
                    fullName.charAt(i) == 'U') {
                strBuff.append(fullName.charAt(i));
                strBuff.append(fullName.charAt(i));

            } else if (fullName.charAt(i) == ' ') {

            } else {
                strBuff.append(fullName.charAt(i));
            }
        }

        // After escaping for loop, the last letter of forename and the last name of
        // surname are added at the index[0], [1] place.
        strBuff.insert(0, foreName.charAt(foreName.length() - 1));
        strBuff.insert(1, surName.charAt(surName.length() - 1));

        // Strbuff is changed to String type
        userName = strBuff.toString();

        // To calcuate how many String userName has uppercase vowels, for loop is used.
        int numberOfUpperCaseVowels = 0;
        for (int i = 0; i < userName.length(); i++) {
            if (userName.charAt(i) == 'A' ||
                    userName.charAt(i) == 'E' ||
                    userName.charAt(i) == 'I' ||
                    userName.charAt(i) == 'O' ||
                    userName.charAt(i) == 'U') {
                numberOfUpperCaseVowels++;
            }
        }

        // Completed userName is maded by adding the number of uppercase vowels.
        userName = userName + numberOfUpperCaseVowels;
```

```
}

// Getter and Setter

// To set forName valuse, set method is used.
public void setForeName(String foreName) {
    this.foreName = foreName;
}

// To set surName valuse, set method is used.
public void setSurName(String surName) {
    this.surName = surName;
}

// fullName is maded by being given forName and surName values.
public void setFullName(String foreName, String surName) {
    this.fullName = foreName + ' ' + surName;
}

// Completed userName value is delivered by get method.
public String getUserName() {
    return userName;
}

// Method2 (Qestion 2) : This method is for validating username corresponding
// with rules.
public void validateUserName() {

    // Through for loop, the programe checks each value of userNameArray
    // and through the other for loop in the for loop, the programe checks each
    // letter of value of userNameArray
    // and through if, else condition, the program validates each value of
    // userNameArray and declare value to isResultOfValidation array
    for (int i = 0; i < userNameArray.length; i++) {

        for (int j = 1; j < userNameArray[i].length(); j++) {
            if (!(userNameArray[i].charAt(0) == '@') ||

                    !(5 <= userNameArray[i].length() && userNameArray[i].length() <= 16) ||

                    !(('a' <= userNameArray[i].charAt(j) && userNameArray[i].charAt(j) <= 'z') ||
                            ('A' <= userNameArray[i].charAt(j) && userNameArray[i].charAt(j) <= 'Z') ||
                            ('0' <= userNameArray[i].charAt(j) && userNameArray[i].charAt(j) <= '9') ||
                            (userNameArray[i].charAt(j) == '_'))

            ) {
                isResultOfValidation[i] = false;
                break;
            }

            else {
                isResultOfValidation[i] = true;
            }
        }

    }

}

// Getter and Setter

// To set userNameNumber valuse, set method is used.
public void setUserNameNumber(int userNameNumber) {
    this.userNameNumber = userNameNumber;
}

// To set userNameArray valuse, set method is used.
public void setUserNameArray(String[] userNameArray) {
    this.userNameArray = userNameArray;
}

// To set isResultOfValidation valuse, set method is used.
public void setIsResultOfValidation(boolean[] isResultOfValidation) {
    this.isResultOfValidation = isResultOfValidation;
}

// userNameArray(Before converting) is maded with certain format like below by
// get method.
public String[] getProvidedUserName() {
        System.out.print("[");
        for (int i = 0; i < userNameNumber; i++) {
            if (i == (userNameNumber - 1)) {
                System.out.print(userNameArray[i]);
            } else {
                System.out.print(userNameArray[i] + "," + " ");
            }
        }
        System.out.println("]");
        return userNameArray;
    }

    // userNameArray(After converting) is maded with certain format like below by
    // get method.
    public String[] getLowerCaseUserName() {
        for (int i = 0; i < userNameNumber; i++) {
            userNameArray[i] = userNameArray[i].toLowerCase();
        }

        System.out.print("[");
        for (int i = 0; i < userNameNumber; i++) {
            if (i == (userNameNumber - 1)) {
                System.out.print(userNameArray[i]);
            } else {
                System.out.print(userNameArray[i] + "," + " ");
            }
        }
        System.out.println("]");
        return userNameArray;
    }

    // isResultOfValidation is maded with certain format like below by get method.
    public boolean[] getIsResultOfValidation() {

        System.out.print("[");
        for (int i = 0; i < userNameNumber; i++) {
            if (i == (userNameNumber - 1)) {
                System.out.print(isResultOfValidation[i]);
            } else {
                System.out.print(isResultOfValidation[i] + "," + " ");
            }
        }
        System.out.println("]");
        return isResultOfValidation;

    }

}
```