Name of the Course : Complete Java SE8 Developer Bootcamp

Level : Moderate

Tool Stack : Java8 and Junit4

Problem Statement : Provide a code solution to arrange alphabets and digits present in a single String in proper order. The validation of input is done through java regular expression and java.lang.Exception.

Description :Pan card’s id consists of alphanumerical figure. Length is always 10. Alphabets are in capital letters. You are required to accept a Pan card id then form a new String in such a way that the numbers in ascending order on the left hand side and alphabets are in descending order on the right hand side. If any alphabet is in lower case, it will be converted into upper case. Please check input is always alphanumerical figure only i.e no special character, no alphabets only, no digit only and the length is always 10.

Create only one class Main with three methods :-

1.public static boolean inputValidation(String pancard): It will check the pancard id is valid or not through java regex. In case input is invalid, it will display error message “Invalid Pan Card Id”.It will return boolean value.

2.public static String arrangeAlphanumerics (String pancard): It arranges the numerics in ascending order on the left hand side and alphabets in descending order on the right hand side of same String. Finally it returns the new String.

3. public static void main(String arg[]): It accepts a Pan Card id and send for validation. If the input is valid then invokes arrangeAlphanumerics() and display output.

Code:

**import** java.util.Arrays;

**import** java.util.Scanner;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** Main {

**public** **static** **boolean** inputValidation (String pancard)

{

**try**

{

**if**(pancard.length()!=10)

**throw** **new** Exception();

pancard=pancard.toUpperCase();

Pattern pattern = Pattern.*compile*("[A-Z]", Pattern.***CASE\_INSENSITIVE***);

Matcher matcher = pattern.matcher(pancard);

**if**(!matcher.find())

**throw** **new** Exception();

pattern = Pattern.*compile*("[0-9]", Pattern.***CASE\_INSENSITIVE***);

matcher = pattern.matcher(pancard);

**if**(!matcher.find())

**throw** **new** Exception();

pattern = Pattern.*compile*("[^A-Z0-9 ]", Pattern.***CASE\_INSENSITIVE***);

matcher = pattern.matcher(pancard);

**if**(matcher.find())

**throw** **new** Exception();

**return** **true**;

}**catch**(Exception exception)

{

System.***out***.println("Invalid Pan Card Id");

**return** **false**;

}

}

**public** **static** String arrangeAlphanumerics (String pancard)

{

String digit="";

String letter="";

**for**(**int** i=0;i<pancard.length();i++)

{

**char** ch=pancard.charAt(i);

**if**(Character.*isDigit*(ch))

digit=digit+ch;

**else** **if**(Character.*isLetter*(ch))

letter=letter+ch;

}

**char** []digitArray=digit.toCharArray();

**char** []letterArray=letter.toCharArray();

Arrays.*sort*(digitArray);

Arrays.*sort*(letterArray);

String output=String.*valueOf*(digitArray);

**for**(**int** i=letterArray.length-1;i>=0;i--)

output=output+letterArray[i];

**return** output;

}

**public** **static** **void** main(String[] args) {

Scanner scanner=**new** Scanner(System.***in***);

System.***out***.println("Enter Pan Card Id.");

String pancard=scanner.nextLine();

**if**(*inputValidation*(pancard))

System.***out***.println(*arrangeAlphanumerics*(pancard));

}

}

Junit Testing

**import** **static** org.junit.Assert.\*;

**import** org.junit.Test;

**public** **class** MainTest {

@Test

**public** **void** testInputValidation() {

*assertEquals*(**true**, Main.*inputValidation*("ATR724S3P5"));

*assertEquals*(**false**, Main.*inputValidation*("ATR28SP5"));

*assertEquals*(**false**, Main.*inputValidation*("TAKYNMOPQE"));

*assertEquals*(**false**, Main.*inputValidation*("5712963210"));

*assertEquals*(**false**, Main.*inputValidation*("ATR2#8SP57"));

}

@Test

**public** **void** testArrangeAlphanumerics() {

*assertEquals*("23457TSRPA", Main.*arrangeAlphanumerics*("ATR724S3P5"));

}

}

Test Data1

Enter Pan Card Id.

ATR724S3P5

23457TSRPA

Test Data2

Enter Pan Card Id.

ATR28SP5

Invalid Pan Card Id

Test Data3

Enter Pan Card Id.

TAKYNMOPQE

Invalid Pan Card Id

Test Data4

Enter Pan Card Id.

5712963210

Invalid Pan Card Id

Test Data5

Enter Pan Card Id.

ATR2#8SP57

Invalid Pan Card Id

Learning outcome: Participant could able to learn how to use the java Regex and java array sorting .