TMA01

# Question 1.a

/\*\*

\* Write a description of class CreditCardChecker here.

\*

\* @author (Michael Witts)

\* @version (20201127)

\* @description (Answer to TMA01 Question 1)

\*/

public class CreditCardChecker

{

// instance variables - replace the example below with your own

/\*\*

\* Constructor for objects of class CreditCardChecker

\*/

public CreditCardChecker()

{

// initialise instance variables

}

}

# Question 1.b

Graphical user interface, application

Description automatically generated

# Question 1.c.i

// check length of string

public boolean isCorrectLength() {

if (longNumber.length() < 17) {

return true;

} else { return false;}

}

# Question 1.c.ii

// get first fifteiien charactiers

public String firstFifteen(){

String result = longNumber.substring(0,15);

return result;

}

# Question 1.d

Here is a Java expression for C where S is of type int

C = S/10 \* 10 + 10 – S

Show that this expression gives a value of 2 when S contains the int value 78. You must show every step in your working, performing exactly one operation on each line.

ANSWER:

78/10 = 7.8 (int would be round down to 7)

7\*10 = 10

70+10=80

80-78=2

# Question 1.e

public int calculateCheckNumber(){  
 String s = getLongNumber();  
 s = firstFifteen();  
 // = s.substring(0,s.length()-1); //get first 15  
 // initialise for algorithm.  
 int sum = 0;  
 boolean isEven = false;  
 for(int i = s.length()-1; i >= 0; i--)  
 {  
 int n = Integer.parseInt(s.substring(i,i+1));  
 if(isEven) //double even values  
 {  
 n \*= 2;  
 if (n > 9) //handle double digit values  
 {  
 n=(n%10)+1;  
 }  
 }  
 sum += n;  
 isEven = !isEven; //flip boolean

}  
 System.out.println("sum is: "+sum);  
 int c = (sum/10)\*10+10-sum;  
 return c;  
 }

# Question 1.f

public void validNumberResult()

{

int num = Integer.parseInt(longNumber.substring(longNumber.length()-1));

System.out.println(num);

int check = calculateCheckNumber();

System.out.println(check);

if(!isCorrectLength() || num != check){

System.out.println("Isn't valid");

} else {

System.out.println("could be valid");

}

}

# Question 1.g

Public methods are accessible from outside of the class in which they are declared. The validNumberResult() method is calling the other methods to ensure that all criterion are met and therefore is the only one which needs to be accessible from outside of the class itself.

# Question 2.a

Graphical user interface, application

Description automatically generated

# Question 2.b

Overall, iCMA41 re-introduced some interesting topics which I have not reviewed for a number of years and the content was delivered in a succinct and easy to digest manner. However, I found it cumbersome to navigate and complete the iCMA from a mobile device. These navigational difficulties made the ability to attempt questions multiple times invaluable, if not somewhat annoying. Overall, however, I found the experience pleasant and informative.