|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  Programming Fundamentals  Lab Mid-Term | **submitted by:**  **Shahzaneer Ahmed**  **registration number:**  **sp21-bcs-087**  **submitted to:**  **Mr. rizwan rashid**  **date of submission:**  **november 21, 2021** |

Credit Card Question

Source Code with Comments

*//-----------------------------------------------------------  
//----------------Lab Mid Term-------------------------------  
//----------------Shahzaneer Ahmed---------------------------  
//----------------SP21-BCS-087-------------------------------  
//-----------------------------------------------------------  
  
import* java.util.Scanner;  
*public class* CreditCard{  
 *public static void* main(String[] args) {  
 Scanner input = *new* Scanner(System.***in***);  
 System.***out***.println("Enter a Credit Card Number : "); *// taking creditCard number from the user.  
 long* CreditCardNumber = input.nextLong(); *//saving input as a long  
  
  
  
 if* (*isValid*(CreditCardNumber)) System.***out***.printf("The %d is valid ",CreditCardNumber); *//if condition comes  
 // to be true  
 else* System.***out***.printf("The %d is invalid ", CreditCardNumber); *//false case* }  
*// we have called methods for different functionalities and have called them in a function that will check all the  
// parameters  
  
 public static boolean* isValid(*long* number){  
 *boolean* isValid = *false*;  
 String creditCardNum = number + "" ;  
 *if* (creditCardNum.length() >= 13 && creditCardNum.length() <= 16) {  
*// firstly , we check if the length of CCN is in accordance with the rules or not.  
 if* (*prefixMatched*(number, 4) || *prefixMatched*(number, 6) || *prefixMatched*(number, 5) || *prefixMatched*(number, 37)){  
*// secondly checking the prefixes .  
 int* sumEven = *sumOfDoubleEvenPlace*(number);  
*// getting sum of double of even places in CCN  
 int* sumOdd = *sumOfOddPlaces*(number);  
*// getting sum of CCN's odd places  
 int* total = sumEven + sumOdd;  
*// their total  
 if* (total%10 == 0) isValid = *true*;  
*// finally checking their mod that is our last condition* }  
 }  
 *return* isValid;  
 }  
 *public static int* sumOfDoubleEvenPlace(*long* number){  
*// this method calculates the sum of doubled values at even places in CCN  
 int* evenSum = 0;  
 String creditCardNum = number + "";  
 *for* (*int* i = creditCardNum.length()-2;i>=0;i-=2){  
 evenSum+= *getDigit*(Character.*getNumericValue*(creditCardNum.charAt(i)));  
*// getDigit method is called here to validate the sum if the sum of doubled digits consists of two integers.* }  
 *return* evenSum;  
  
 }  
 *public static int* getDigit(*int* number){  
*// this method the digit if it is single and if the digit is two numbered it will  
// return the sum of those two digits!  
  
 int* digit = number \* 2;  
 *return* (digit/10) + (digit%10);  
*//* }  
 *public static int* sumOfOddPlaces(Long number){  
*// this method will return the sum of oddly placed digits in the CCN.  
 int* oddSum = 0;  
 String creditCardNum = number + "";  
 *for*(*int* i = creditCardNum.length()-1;i>=0;i-=2){  
 oddSum+=Character.*getNumericValue*(creditCardNum.charAt(i));  
 }  
 *return* oddSum;  
 }  
 *public static boolean* prefixMatched(*long* number, *int* d){  
*// this method will check it the required prefixes are present in the CNN or not.* String creditCardNum = number + "";  
 String prefix = d + "";  
 Boolean isPrefixedMatched = *false*;  
 *if* (creditCardNum.startsWith(prefix)) isPrefixedMatched = *true*;  
  
 *return* isPrefixedMatched;  
 }  
 *public static int* getSize(*long* d){  
*// This Method will get the size of creditCardNum digits.* String creditNum = d + ""; *// creditNum will Store a String version of creditCard.  
 int* noOfDigits = 0; *// counter to count the number of digits  
  
 for* (*int* i= 0; i<creditNum.length();i++){  
 *if* (Character.*isDigit*(creditNum.charAt(i))){  
*// if the character at i's place is digit counter counts the digits!* noOfDigits+=1;  
 }  
 }  
 *return* noOfDigits;  
  
 }  
 *public static long* getPrefix(*long* number, *int* k){  
*// this method will get the prefix from the CCN.  
 if* (*getSize*(number) < k) {  
 *return* number;  
 }  
 *else* {  
 String s = number + "";  
 *return* Long.*parseLong*(s.substring(0, k));  
 }  
 }  
  
  
  
  
}

Output Screenshot

  
