Name of the Course : Learning Java 9 - Object Oriented Programming

Level : Easy

Tool Stack : Java9 and Junit4

Problem Statement : Provide a code solution to calculate electricity bill payment for group of consumers based on the electric consumption over a period based on Java 9 code solution of array etc features.

Description : ***National Electricity Corporation(NEC)*** charges monthly electricity charges based on consumption of it’s consumers. The rate of charge is as follows:

1. Upto 200 unit every consumer must pay flat an amount of Rs 300/- (even no consumption).
2. From 201 to 500 unit rate is Rs 1.25/unit.
3. From 501 to 1000 unit rate is Rs 1.00/unit.
4. From 1001 unit and above rate is 0.75/unit.

Consumption unit must be in whole number.

The computer operator of NEC normally accepts consumer’s details like consumer number, name, unit consumption in a comma(,) separate String to generate bill (eg: 653,Steve Jones,754). The operator normally generates bill for number of consumers at a time. So before starting operation, the operator asks for number of consumers’ bill to prepare. All bills are stored in an array. Finally display the contains of array, all decimal figures must show 2 decimal places.

You need create

1. class Consumer with private member data

String id;

String name;

Integer unitConsumed;

String finalPayment;

Create getter/setter methods and constructors.

override toString() in String.format("%-5s %-20s %-10s %-10s").

1. class BillService with member function

public static String billCalcultion(Consumer consumer): It will calculate payament amount based of above formula.

1. class Main with method public static void main(String [] arg): It will asks for number of consumers, If it is 0 or negative number message “invalid input” will be displayed and stop the application, if it is a valid positive number then create an array of Consumer and ask “Enter details of consumer number 1”, “Enter details of consumer number 2”,... so on. After this each of consumer’s bill will be generated then finally display each consumer’s id, name, unit consumed and amount payable.

Code:

**import** lombok.AllArgsConstructor;

**import** lombok.Data;

**import** lombok.NoArgsConstructor;

@Data

@AllArgsConstructor

@NoArgsConstructor

**public** **class** Consumer {

**private** String id;

**private** String name;

**private** Integer unitConsumed;

**private** String finalPayment;

**public** Consumer(String id, String name, Integer unitConsumed) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.unitConsumed = unitConsumed;

}

@Override

**public** String toString() {

String output=String.*format*("%-5s %-20s %-10s %-10s",id,name,unitConsumed,finalPayment);

**return** output;

}

}

**import** java.text.DecimalFormat;

**public** **class** BillService {

**public** **static** String billCalcultion(Consumer consumer)

{

**double** billValue=0.0;

**int** consumption=consumer.getUnitConsumed();

**if**(consumption<=200)

billValue=300.00;

**else** **if**(consumption<=500)

{

**int** remain=consumption-200;

billValue=300.00+(remain\*1.25);

}

**else** **if**(consumption<=1000)

{

**int** remain=consumption-500;

billValue=300.00+((500-200)\*1.25)+(remain\*1.00);

}

**else**

{

**int** remain=consumption-1000;

billValue=300.00+((500-200)\*1.25)+((1000-500)\*1.00)+(remain\*0.75);

}

DecimalFormat decimalFormat=**new** DecimalFormat("0.00");

String finalPayment=decimalFormat.format(billValue);

**return** finalPayment;

}

}

**import** java.util.Scanner;

**public** **class** Main

{

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

{

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

System.***out***.println("Enter Number of consumers bill to prepare:");

**int** no=Integer.*parseInt*(scanner.nextLine());

**if**(no<=0)

{

System.***out***.println("invalid input");

System.*exit*(0);

}

Consumer consumerArray[]=**new** Consumer[no];

**for**(**int** i=0;i<no;i++)

{

**int** j=i+1;

System.***out***.println("Enter details of consumer number "+j+":");

String input=scanner.nextLine();

String arr[]=input.split(",");

consumerArray[i]=**new** Consumer(arr[0],arr[1],Integer.*parseInt*(arr[2]));

String value=BillService.*billCalcultion*(consumerArray[i]);

consumerArray[i].setFinalPayment(value);

}

System.***out***.println(String.*format*("%-5s %-20s %-10s %-10s","ID","NAME","CONSUME UNIT","AMOUNT"));

**for**(Consumer consumer:consumerArray)

System.***out***.println(consumer);

}

}

Junit Testing

**import** java.io.File;

**import** java.io.FileWriter;

**import** java.io.IOException;

**public** **class** TestUtils {

**public** **static** File *businessTestFile*;

**public** **static** File *boundaryTestFile*;

**public** **static** File *exceptionTestFile*;

**static** {

*businessTestFile* = **new** File("./output\_revised.txt");

*businessTestFile*.delete();

*boundaryTestFile* = **new** File("./output\_boundary\_revised.txt");

*boundaryTestFile*.delete();

*exceptionTestFile* = **new** File("./output\_exception\_revised.txt");

*exceptionTestFile*.delete();

}

**public** **static** **void** yakshaAssert(String testName, Object result, File file) **throws** IOException {

System.***out***.println("\n" + testName + "=" + result);

FileWriter writer = **new** FileWriter(file,**true**);

writer.append("\n" + testName + "=" + result);

writer.flush();

writer.close();

}

**public** **static** String currentTest() {

**return** Thread.*currentThread*().getStackTrace()[2].getMethodName();

}

}

**import** junit.framework.TestCase;

**import** **static** java9.mod.modapp1.TestUtils.\*;

**public** **class** BillServiceTest **extends** TestCase {

**public** **void** testBillCalcultion() **throws** Exception {

Consumer con1=**new** Consumer("421","John",1050);

Consumer con2=**new** Consumer("519","Stella",150);

Consumer con3=**new** Consumer("218","Clare",60);

String value1=BillService.*billCalcultion*(con1);

String value2=BillService.*billCalcultion*(con2);

String value3=BillService.*billCalcultion*(con3);

*yakshaAssert*(*currentTest*(),(value1.equals("1212.50")?"true":"false"),*businessTestFile*);

*yakshaAssert*(*currentTest*(),(value2.equals("300.00")?"true":"false"),*businessTestFile*);

*yakshaAssert*(*currentTest*(),(value3.equals("760.00")?"true":"false"),*businessTestFile*);

}

}

Test Data1

Enter Number of consumers bill to prepare:

-3

invalid input

Test Data2

Enter Number of consumers bill to prepare:

3

Enter details of consumer number 1:

421,John,1050

Enter details of consumer number 2:

519,Stella,150

Enter details of consumer number 3:

218,Clare,760

ID NAME CONSUME UNIT AMOUNT

421 John 1050 1212.50

519 Stella 150 300.00

218 Clare 760 935.00

Learning outcome: Participant could able to use Object array, String spilt() to create objects.