ICT Electricity Bill

Automatic evaluation[+]

EBBill/src/db.properties

```
db.classname=com.mysql.jdbc.Driver
db.url=jdbc:mysql://localhost:3306/ElectricityBill
db.username=
db.password=
```

EBBill/src/DBHandler.java

```
import java.sql.Connection;
import java.io.FileInputStream;
import java.io.IOException;
import java.sql.*;
import java.util.Properties;
import java.io.FileNotFoundException;

public class DBHandler {

        public Connection establishConnection() throws ClassNotFoundException, SQLException,
FileNotFoundException {

            Connection con = null;
            Properties props = new Properties();
            // this try block reads the db Properties file and establishConnection.
            try{
                FileInputStream fis = new FileInputStream("src/db.properties");
                props.load(fis);

                Class.forName(props.getProperty("db.classname"));

                con =
DriverManager.getConnection(props.getProperty("db.url"),props.getProperty("db.username"),props.getProperty("
db.password"));
            }
            catch(IOException e){
                e.printStackTrace();
            }

            return con;


        //fill code here

        }
}

```

EBBill/src/ElectricityBill.java

```

//This is the POJO/model class

public class ElectricityBill {

    private String consumerNumber;
    private String consumerName;
    private String consumerAddress;
```

```
9   private int unitsConsumed;
10   private double billAmount;
11
12   public String getConsumerNumber() {
13                       return consumerNumber;
14          }
15
16          public void setConsumerNumber(String consumerNumber) {
17                       this.consumerNumber = consumerNumber;
18          }
19
20          public String getConsumerName() {
21                       return consumerName;
22          }
23
24          public void setConsumerName(String consumerName) {
25                       this.consumerName = consumerName;
26          }
27
28          public String getConsumerAddress() {
29                       return consumerAddress;
30          }
31
32          public void setConsumerAddress(String consumerAddress) {
33                       this.consumerAddress = consumerAddress;
34          }
35
36          public int getUnitsConsumed() {
37                       return unitsConsumed;
38          }
39
40          public void setUnitsConsumed(int unitsConsumed) {
41                       this.unitsConsumed = unitsConsumed;
42          }
43
44          public double getBillAmount() {
45                       return billAmount;
46          }
47
48          public void setBillAmount(double billAmount) {
49                       this.billAmount = billAmount;
50          }
51
52
53          //Write the required business logic as expected in the question description
54          public void calculateBillAmount() {
55              // method for calaculating the bill amount.
56              int units = unitsConsumed;
57              double bill = 0;
58
59              if(units <= 100){
60                  bill = 0;
61              }
62              if(units > 100 && units <= 300){
63                  bill = (units-100) * 1.5;
64              }
65              if(units > 300 && units <= 600){
66                  bill = 200 * 1.5 + (units-300) * 3.5;
67              }
68              if(units > 600 && units <= 1000){
69                  bill = 200 * 1.5 + 300 * 3.5 + (units-600) * 5.5;
70              }
71              if(units > 1000){
72                  bill = 200 * 1.5 + 300 * 3.5 + 400 * 5.5 + (units-1000) * 7.5;
73              }
74
75              setBillAmount(bill);
```

```
76
77
78          //fill the code
79
80    }
81
82 }
```

EBBill/src/ElectricityBill.txt

```
1 0191919191,John,Chennai,650
2 0191919192,Peter,Mumbai,1100
3 1919191919,Rose,Mumbai,453
4 0191919193,Tom,Hyderabad,750
5 01919191945,Raj,Chennai,120
6 0191919194,Sam,Chennai,250
7 0191919195,Anya,Chennai,34
```

EBBill/src/ElectricityBoard.java

```
1  import java.util.List;
2  import java.util.*;
3  import java.io.FileReader;
4  import java.io.File;
5  import java.io.BufferedReader;
6  import java.io.FileNotFoundException;
7  import java.io.IOException;
8  import java.util.regex.Pattern;
9  import java.sql.SQLException;
10 import java.sql.Connection;
11 import java.sql.PreparedStatement;
12
13 public class ElectricityBoard {
14
15    //write the required business logic methods as expected in the question description
16    public void addBill(List<ElectricityBill> billList) {
17
18        DBHandler db =  new DBHandler();
19
20        try(Connection con = db.establishConnection()){
21
22            PreparedStatement stmt = con.prepareStatement("insert into ElectricityBill values(?,?,?,?,?);");
23
24            // for loop to insert the values into the table
25            for(ElectricityBill obj : billList){
26                stmt.setString(1,obj.getConsumerNumber());
27                stmt.setString(2,obj.getConsumerName());
28                stmt.setString(3,obj.getConsumerAddress());
29                stmt.setInt(4,obj.getUnitsConsumed());
30                stmt.setDouble(5,obj.getBillAmount());
31
32                stmt.execute();
33            }
34        }
35        catch(ClassNotFoundException e){
36            e.printStackTrace();
37        }
38
39        catch(FileNotFoundException e){
40            e.printStackTrace();
41        }
42
43        catch(SQLException e){
44            e.printStackTrace();
45        }
46
47
```

```
48
49
50      //fill the code
51
52  }
53
54  public List<ElectricityBill> generateBill(String filePath) {
55
56      List <ElectricityBill> list = new ArrayList<>();
57      File f = new File (filePath);
58
59      // this try block is for opening and reading the file
60      try(BufferedReader br = new BufferedReader(new FileReader(f)))
61      {
62          String line = null;
63
64          while((line = br.readLine())!= null)
65          {
66              String records[] = null;
67              String consumerNumber = "";
68              String consumerName = "";
69              String consumerAddress = "";
70              int unitsConsumed = 0;
71
72              records = line.split(",");
73              consumerNumber = records[0];
74              consumerName = records[1];
75              consumerAddress = records[2];
76              unitsConsumed = Integer.parseInt(records[3]);
77
78
79          //this try block checks for the validated consumerNumber
80              try{
81                  if(validate(consumerNumber)){
82                      ElectricityBill obj = new ElectricityBill();
83                      obj.setConsumerNumber(consumerNumber);
84                      obj.setConsumerName(consumerName);
85                      obj.setConsumerAddress(consumerAddress);
86                      obj.setUnitsConsumed(unitsConsumed);
87                      obj.calculateBillAmount();
88
89                      list.add(obj);
90                  }
91              }
92              catch(InvalidConsumerNumberException e){
93                  System.out.println(e.getMessage());
94              }
95          }
96      }
97      catch(FileNotFoundException e){
98          e.printStackTrace();
99      }
100      catch(IOException e){
101          e.printStackTrace();
102      }
103
104  return list;
105      //fill the code
106
107  }
108
109  public boolean validate(String consumerNumber) throws InvalidConsumerNumberException {
110
111      // method for validating the consumerNumber
112          boolean isValid = Pattern.matches("^[0][0-9]{9}" , consumerNumber);
113
114          if(!isValid){
```

```
            throw new InvalidConsumerNumberException("Invalid Consumer Number");
        }

        return true;

                    //fill the code

    }

}

```

EBBill/src/InvalidConsumerNumberException.java

```

//make the required changes to this class so that InvalidConsumerNumberException is of type exception.

public class InvalidConsumerNumberException extends Exception{

    public InvalidConsumerNumberException(String message)
    {
        super(message);
    }

     //fill the code

}
```

EBBill/src/Main.java

```
import java.util.*;
import java.util.List;
import java.util.ArrayList;
public class Main {

    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);

        String filePath = "src/ElectricityBill.txt";

        List<ElectricityBill> list = new ArrayList<>();

        ElectricityBoard eb = new ElectricityBoard();
        list = eb.generateBill(filePath);

        for(ElectricityBill obj: list){
            System.out.println(obj.getConsumerNumber() + " " + obj.getConsumerName() + " " +
obj.getBillAmount());

        }

        eb.addBill(list);

        System.out.println("Successfully Inserted");

        sc.close();
            //fill your code here

    }

}

```

script.sql

```
drop database if exists EBBill;

```

```
3 create database EBBill;
4
5 use EBBill;
6
7 create table ElectricityBill(consumer_number varchar(15) primary key,name varchar(2
8 5),address varchar(30),units_consumed int(5),bill_amount float(6,2));
9
10 COMMIT;
```

Grade

Reviewed on Tuesday, 19 January 2021, 8:00 PM by Automatic grade
Grade 98.75 / 100
Assessment report
[+]**Grading and Feedback**