Advance java data structures project

REPORT

Electricity Bill Generator

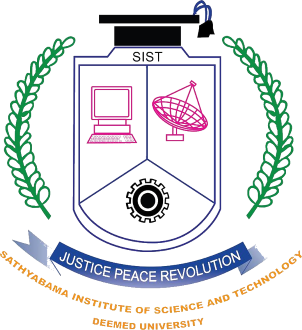
*By*

*Lekhasri V(41130277)*

*Akshaya sri Saravanan(41130016)*

*Boddu Lakshmi Sameera(41130076)*

*V Surya Deepika(41130656)*



Sathyabama

Institute of Science and Technology

(Deemed to be University)

**Abstract**

The Electricity Bill Generator project aims to develop a simple utility for calculating electricity bills based on the units consumed by the user. This project provides a straightforward solution for individuals or small businesses to generate accurate electricity bills without the need for complex billing systems. The application prompts the user to input the number of units consumed, calculates the total bill amount using a fixed rate per unit, and displays a summary of the bill. The report outlines the objectives, features, implementation details, and includes a sample code snippet for reference.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Contents** | **PG.NO** |
| 1. | Introduction | 1 |
| 2. | Features | 1 |
| 3. | Implementation details | 2 |
| 4. | Source code | 3 |
| 5. | Result | 8 |
| 6. | Conclusion | 10 |

**Introduction**

The Electricity Bill Generator project aims to provide a simple utility for generating electricity bills based on the number of units consumed by the user. The application prompts the user to input the number of units consumed and calculates the total bill amount using a fixed rate per unit.

**Features**

* *Input Units:* Users can input the number of units consumed.
* *Calculation:* The application calculates the total bill amount based on the fixed rate per unit.
* *Display Bill:* It displays a summary of the bill including units consumed, rate per unit, and total amount due.

1

**Implementation Details**

**Technologies Used:**

* *Java:* The core programming language used for development.
* *Scanner Class:* Utilized for user input.
* *Basic Arithmetic:* Simple calculations are performed to determine the total bill amount.

**Code Structure:**

* *ElectricityBillGenerator.java:* Main class containing the application logic.

**Class Overview:**

* *ElectricityBillGenerator:* Contains the main method and logic for generating electricity bills.

2

**Source code:**

package electricitybill;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.util.Scanner;

abstract class Customer {

protected String name;

protected int unitsConsumed;

protected LocalDate lastDateToPay;

public Customer(String name, int unitsConsumed, LocalDate lastDateToPay) {

this.name = name;

this.unitsConsumed = unitsConsumed;

this.lastDateToPay = lastDateToPay;

}

public abstract double calculateBill();

}

class ResidentialCustomer extends Customer {

3

private static final double RES\_RATE = 2.5;

public ResidentialCustomer(String name, int unitsConsumed, LocalDate lastDateToPay) {

super(name, unitsConsumed, lastDateToPay);

}

@Override

public double calculateBill() {

return unitsConsumed \* RES\_RATE;

}

}

class CommercialCustomer extends Customer {

private static final double COM\_RATE = 5.5;

public CommercialCustomer(String name, int unitsConsumed, LocalDate lastDateToPay) {

super(name, unitsConsumed, lastDateToPay);

}

@Override

public double calculateBill() {

return unitsConsumed \* COM\_RATE;

}

}

4

public class ElectricityBillSimulation {

public static void main(String[] args) {

@SuppressWarnings("resource")

Scanner scanner = new Scanner(System.in);

while (true) {

try {

System.out.println("Enter customer details:");

System.out.print("Name: ");

String name = scanner.nextLine();

System.out.print("EB card number: ");

int EB\_NO = scanner.nextInt();

System.out.print("Units consumed: ");

int unitsConsumed = scanner.nextInt();

scanner.nextLine();

int daysUntilPayment = 15;

LocalDate today = LocalDate.now();

LocalDate lastDateToPay = today.plusDays(daysUntilPayment);

5

System.out.print("Customer type (1 for Residential, 2 for Commercial): ");

int customerType = scanner.nextInt();

Customer customer;

if (customerType == 1) {

customer = new ResidentialCustomer(name, unitsConsumed, lastDateToPay);

}

else

{

customer = new CommercialCustomer(name, unitsConsumed, lastDateToPay);

}

double billAmount = customer.calculateBill();

System.out.println("Electricity bill for " + customer.name + ": Rs." + billAmount);

System.out.println("\*\*\*\*\*\*Here is your bill\*\*\*\*\*\*");

6

System.out.println("NAME: " + name);

System.out.println("EB Card Number: " + EB\_NO);

System.out.println("Units consumed: " + unitsConsumed);

System.out.println("Bill amount: Rs." + billAmount);

System.out.println("Last date to pay your bill: " + lastDateToPay.format(DateTimeFormatter.ofPattern("dd-MM-yyyy")));

System.out.println("\*\*\*\*Thank you for visiting\*\*\*\*\*");

break;

}

catch (Exception e) {

System.out.println("Error occurred: " + e.getMessage());

System.out.println("Please enter the details again.");

scanner.nextLine();

}

}

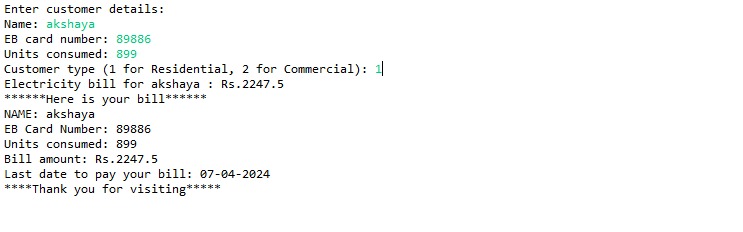
7

}

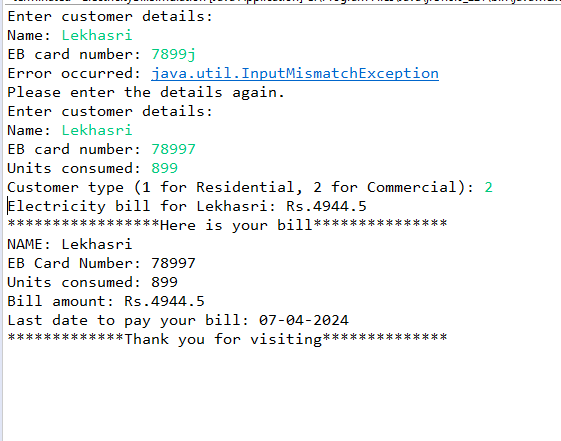
}

**Result**

*Output:*

**

8

**

9

**Conclusion**

The Electricity Bill Generator project provides a basic yet functional solution for generating electricity bills. It can be further extended with additional features such as tiered pricing, discounts, and user authentication for more comprehensive utility.

This report outlines the objectives, features, implementation details, and provides a sample code snippet for the Electricity Bill Generator project. It serves as documentation for understanding the purpose and functionality of the application.

10