**Telecom Call Billing System**

**✅ \*\*Project Summary:**

A console-based Core Java application\*\* that simulates a simplified postpaid billing system. It tracks user calls, SMS, and data usage, then generates a monthly bill with itemized charges and taxes.

**🧾 Key Functionalities:**

1. **Customer Management**
   * Add, update, or remove customer records
   * Store: Name, Phone Number, Plan Type (Basic/Pro), Plan Rate
2. **Usage Entry Module**
   * Add call records (duration in minutes)
   * Add SMS records
   * Add data usage (in MB or GB)
3. **Billing System**
   * Calculate charges based on:
     + Per minute rate for calls
     + Per SMS rate
     + Per MB/GB for data
   * Apply taxes (e.g., 18% GST)
   * Discounts for pro users (optional)
4. **Generate Monthly Bill**
   * Total usage summary
   * Taxes and total amount
   * Save to .txt file as invoice
5. **Search/View Customer Bill**
   * View all customers with billing summary

**🛠 Core Java Concepts Used:**

| **Feature** | **Java Concept** |
| --- | --- |
| Customer/Usage Models | OOP, Classes, Inheritance |
| Storing Records | ArrayList, HashMap |
| Calculations | Conditionals, Loops |
| Billing Logic | Encapsulation, Abstraction |
| File Output | File I/O (BufferedWriter) |
| CLI Interface | Scanner, Switch-case |
| Error Handling | Try-Catch Blocks |

**🧪 Sample CLI Menu:**

----- Telecom Call Billing System -----

1. Add New Customer

2. Enter Usage Details (Call/SMS/Data)

3. Generate Monthly Bill

4. View Customer Bill

5. List All Customers

6. Exit

Enter your choice:

**💡 Optional Enhancements:**

* **Plan Management**: Different plans with free usage limits
* **Login system**: Admin login to manage customers
* **Serialization**: Persist customer data across sessions
* **Report**: Total revenue generated for the month

**📂 Example Output for Bill File:**

Customer Name: Raj Malhotra

Phone Number: 9876543210

Plan: Pro

Usage Summary:

- Calls: 120 minutes @ ₹1.00 = ₹120.00

- SMS: 30 @ ₹0.50 = ₹15.00

- Data: 2.5 GB @ ₹10.00 = ₹25.00

Subtotal: ₹160.00

GST (18%): ₹28.80

-----------------------

Total Bill: ₹188.80

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.io.IOException;

import java.util.\*;

class Customer {

private String name;

private String phoneNumber;

private String plan;

private List<Call> calls = new ArrayList<>();

private int smsCount = 0;

private double dataUsageGB = 0;

// Rates (could be loaded/configured)

private static final double CALL\_RATE = 1.0; // per minute

private static final double SMS\_RATE = 0.5; // per SMS

private static final double DATA\_RATE = 10.0; // per GB

private static final double GST\_PERCENT = 18.0;

public Customer(String name, String phoneNumber, String plan) {

this.name = name;

this.phoneNumber = phoneNumber;

this.plan = plan;

}

public String getPhoneNumber() {

return phoneNumber;

}

public String getName() {

return name;

}

public void addCall(int minutes) {

calls.add(new Call(minutes));

}

public void addSms(int count) {

smsCount += count;

}

public void addData(double gb) {

dataUsageGB += gb;

}

public double calculateBill() {

double callCost = totalCallMinutes() \* CALL\_RATE;

double smsCost = smsCount \* SMS\_RATE;

double dataCost = dataUsageGB \* DATA\_RATE;

// Example discount for Pro plan

double subtotal = callCost + smsCost + dataCost;

if ("Pro".equalsIgnoreCase(plan)) {

subtotal \*= 0.9; // 10% discount

}

double gst = subtotal \* GST\_PERCENT / 100;

return subtotal + gst;

}

public int totalCallMinutes() {

int sum = 0;

for (Call c : calls) {

sum += c.getDuration();

}

return sum;

}

public void printBill() {

System.out.println("----- Bill for " + name + " (" + phoneNumber + ") -----");

System.out.printf("Plan: %s\n", plan);

System.out.printf("Calls: %d minutes @ ₹%.2f = ₹%.2f\n", totalCallMinutes(), CALL\_RATE, totalCallMinutes() \* CALL\_RATE);

System.out.printf("SMS: %d @ ₹%.2f = ₹%.2f\n", smsCount, SMS\_RATE, smsCount \* SMS\_RATE);

System.out.printf("Data: %.2f GB @ ₹%.2f = ₹%.2f\n", dataUsageGB, DATA\_RATE, dataUsageGB \* DATA\_RATE);

double subtotal = totalCallMinutes() \* CALL\_RATE + smsCount \* SMS\_RATE + dataUsageGB \* DATA\_RATE;

if ("Pro".equalsIgnoreCase(plan)) {

subtotal \*= 0.9; // discount

System.out.println("Pro plan 10% discount applied.");

}

double gst = subtotal \* GST\_PERCENT / 100;

System.out.printf("Subtotal: ₹%.2f\n", subtotal);

System.out.printf("GST (%.0f%%): ₹%.2f\n", GST\_PERCENT, gst);

System.out.println("-----------------------");

System.out.printf("Total Bill: ₹%.2f\n", subtotal + gst);

}

public void saveBillToFile() {

String filename = phoneNumber + "\_bill.txt";

try (BufferedWriter writer = new BufferedWriter(new FileWriter(filename))) {

writer.write("Customer Name: " + name + "\n");

writer.write("Phone Number: " + phoneNumber + "\n");

writer.write("Plan: " + plan + "\n\n");

writer.write("Usage Summary:\n");

writer.write(String.format("- Calls: %d minutes @ ₹%.2f = ₹%.2f\n",

totalCallMinutes(), CALL\_RATE, totalCallMinutes() \* CALL\_RATE));

writer.write(String.format("- SMS: %d @ ₹%.2f = ₹%.2f\n",

smsCount, SMS\_RATE, smsCount \* SMS\_RATE));

writer.write(String.format("- Data: %.2f GB @ ₹%.2f = ₹%.2f\n\n",

dataUsageGB, DATA\_RATE, dataUsageGB \* DATA\_RATE));

double subtotal = totalCallMinutes() \* CALL\_RATE + smsCount \* SMS\_RATE + dataUsageGB \* DATA\_RATE;

if ("Pro".equalsIgnoreCase(plan)) {

subtotal \*= 0.9;

writer.write("Pro plan 10% discount applied.\n");

}

double gst = subtotal \* GST\_PERCENT / 100;

writer.write(String.format("Subtotal: ₹%.2f\n", subtotal));

writer.write(String.format("GST (%.0f%%): ₹%.2f\n", GST\_PERCENT, gst));

writer.write("-----------------------\n");

writer.write(String.format("Total Bill: ₹%.2f\n", subtotal + gst));

System.out.println("Bill saved to " + filename);

} catch (IOException e) {

System.out.println("Error writing bill to file: " + e.getMessage());

}

}

}

class Call {

private int duration; // in minutes

public Call(int duration) {

this.duration = duration;

}

public int getDuration() {

return duration;

}

}

public class TelecomBillingSystem {

private static Map<String, Customer> customers = new HashMap<>();

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

boolean exit = false;

while (!exit) {

printMenu();

int choice = readInt();

switch (choice) {

case 1 -> addCustomer();

case 2 -> addUsage();

case 3 -> generateBill();

case 4 -> viewCustomerBill();

case 5 -> listCustomers();

case 6 -> exit = true;

default -> System.out.println("Invalid choice! Try again.");

}

}

System.out.println("Thank you for using Telecom Billing System.");

}

private static void printMenu() {

System.out.println("\n----- Telecom Call Billing System -----");

System.out.println("1. Add New Customer");

System.out.println("2. Enter Usage Details (Call/SMS/Data)");

System.out.println("3. Generate Monthly Bill & Save");

System.out.println("4. View Customer Bill");

System.out.println("5. List All Customers");

System.out.println("6. Exit");

System.out.print("Enter your choice: ");

}

private static void addCustomer() {

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

String name = scanner.nextLine().trim();

System.out.print("Enter Phone Number: ");

String phone = scanner.nextLine().trim();

if (customers.containsKey(phone)) {

System.out.println("Customer with this phone number already exists.");

return;

}

System.out.print("Enter Plan Type (Basic/Pro): ");

String plan = scanner.nextLine().trim();

if (!plan.equalsIgnoreCase("Basic") && !plan.equalsIgnoreCase("Pro")) {

System.out.println("Invalid plan type! Defaulting to Basic.");

plan = "Basic";

}

customers.put(phone, new Customer(name, phone, plan));

System.out.println("Customer added successfully.");

}

private static void addUsage() {

System.out.print("Enter Phone Number: ");

String phone = scanner.nextLine().trim();

Customer customer = customers.get(phone);

if (customer == null) {

System.out.println("Customer not found!");

return;

}

System.out.println("Select Usage Type:");

System.out.println("1. Call");

System.out.println("2. SMS");

System.out.println("3. Data");

System.out.print("Enter choice: ");

int usageType = readInt();

switch (usageType) {

case 1 -> {

System.out.print("Enter Call Duration (minutes): ");

int minutes = readInt();

if (minutes > 0) {

customer.addCall(minutes);

System.out.println("Call record added.");

} else System.out.println("Invalid duration.");

}

case 2 -> {

System.out.print("Enter number of SMS: ");

int sms = readInt();

if (sms > 0) {

customer.addSms(sms);

System.out.println("SMS record added.");

} else System.out.println("Invalid SMS count.");

}

case 3 -> {

System.out.print("Enter Data Usage (GB): ");

double data = readDouble();

if (data > 0) {

customer.addData(data);

System.out.println("Data usage added.");

} else System.out.println("Invalid data usage.");

}

default -> System.out.println("Invalid usage type.");

}

}

private static void generateBill() {

System.out.print("Enter Phone Number: ");

String phone = scanner.nextLine().trim();

Customer customer = customers.get(phone);

if (customer == null) {

System.out.println("Customer not found!");

return;

}

customer.printBill();

customer.saveBillToFile();

}

private static void viewCustomerBill() {

System.out.print("Enter Phone Number: ");

String phone = scanner.nextLine().trim();

Customer customer = customers.get(phone);

if (customer == null) {

System.out.println("Customer not found!");

return;

}

customer.printBill();

}

private static void listCustomers() {

if (customers.isEmpty()) {

System.out.println("No customers available.");

return;

}

System.out.println("Customers List:");

for (Customer c : customers.values()) {

System.out.printf("- %s (%s) Plan: %s\n", c.getName(), c.getPhoneNumber(), c.plan);

}

}

private static int readInt() {

while (true) {

try {

String line = scanner.nextLine();

return Integer.parseInt(line.trim());

} catch (NumberFormatException e) {

System.out.print("Invalid number, try again: ");

}

}

}

private static double readDouble() {

while (true) {

try {

String line = scanner.nextLine();

return Double.parseDouble(line.trim());

} catch (NumberFormatException e) {

System.out.print("Invalid decimal number, try again: ");

}

}

}

}

**How to run:**

1. Copy this code into a file named TelecomBillingSystem.java.
2. Compile with javac TelecomBillingSystem.java.
3. Run with java TelecomBillingSystem.
4. Follow the CLI menu to add customers, usage, generate bills, and save files.

Prepare class diagram :