**Name:**Shreyas S

**Regno:**192321039

**Smart Traffic Signal Optimization**

**Scenario:** You are part of a team working on an initiative to optimize traffic signal management in a busy city to reduce congestion and improve traffic flow efficiency using smart technologies.

import java.util.Random;

import java.util.Scanner;

class TrafficSignal {

enum SignalState {

RED, GREEN, YELLOW

}

private SignalState currentState;

private int vehicleCount;

public TrafficSignal() {

this.currentState = SignalState.RED;

this.vehicleCount = 0;

}

public void changeSignal() {

switch (currentState) {

case RED:

currentState = SignalState.GREEN;

break;

case GREEN:

currentState = SignalState.YELLOW;

break;

case YELLOW:

currentState = SignalState.RED;

break;

}

System.out.println("Signal changed to: " + currentState);

}

public void updateVehicleCount(int count) {

this.vehicleCount = count;

System.out.println("Vehicle count updated to: " + vehicleCount);

}

public void optimizeSignal() {

if (vehicleCount > 10 && currentState == SignalState.RED) {

changeSignal();

}

}

public SignalState getCurrentState() {

return currentState;

}

public int getVehicleCount() {

return vehicleCount;

}

}

public class TrafficSignalSystem {

public static void main(String[] args) {

TrafficSignal signal = new TrafficSignal();

Scanner scanner = new Scanner(System.in);

Random random = new Random();

while (true) {

System.out.println("\nTraffic Signal System Menu:");

System.out.println("1. Change Signal Manually");

System.out.println("2. Update Vehicle Count");

System.out.println("3. Optimize Signal");

System.out.println("4. Check Signal State");

System.out.println("5. Exit");

System.out.print("Choose an option: ");

int option = scanner.nextInt();

switch (option) {

case 1:

signal.changeSignal();

break;

case 2:

System.out.print("Enter vehicle count: ");

int count = scanner.nextInt();

signal.updateVehicleCount(count);

break;

case 3:

signal.optimizeSignal();

break;

case 4:

System.out.println("Current Signal State: " + signal.getCurrentState());

System.out.println("Current Vehicle Count: " + signal.getVehicleCount());

break;

case 5:

System.out.println("Exiting the Traffic Signal System. Visit again");

scanner.close();

return;

default:

System.out.println("Invalid option. Please choose again.");

}

}

}

}

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

class TrafficData {

private int vehicleCount;

private int speed;

private String intersectionId;

public TrafficData(int vehicleCount, int speed, String intersectionId) {

this.vehicleCount = vehicleCount;

this.speed = speed;

this.intersectionId = intersectionId;

}

public int getVehicleCount() {

return vehicleCount;

}

public int getSpeed() {

return speed;

}

public String getIntersectionId() {

return intersectionId;

}

@Override

public String toString() {

return "Intersection: " + intersectionId + ", Vehicle Count: " + vehicleCount + ", Speed: " + speed;

}

}

class TrafficSignal {

enum SignalState {

RED, GREEN, YELLOW

}

private SignalState currentState;

private String intersectionId;

public TrafficSignal(String intersectionId) {

this.currentState = SignalState.RED;

this.intersectionId = intersectionId;

}

public void changeSignal() {

switch (currentState) {

case RED:

currentState = SignalState.GREEN;

break;

case GREEN:

currentState = SignalState.YELLOW;

break;

case YELLOW:

currentState = SignalState.RED;

break;

}

System.out.println("Signal at " + intersectionId + " changed to: " + currentState);

}

public void optimizeSignal(TrafficData data) {

if (data.getVehicleCount() > 10 && currentState == SignalState.RED) {

changeSignal();

} else if (data.getVehicleCount() < 3 && currentState == SignalState.GREEN) {

changeSignal();

}

}

public SignalState getCurrentState() {

return currentState;

}

public String getIntersectionId() {

return intersectionId;

}

}

public class TrafficSignalSystem {

private static Map<String, TrafficSignal> signals = new HashMap<>();

private static Map<String, TrafficData> trafficDataMap = new HashMap<>();

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\nTraffic Signal System Menu:");

System.out.println("1. Add Intersection");

System.out.println("2. Update Traffic Data");

System.out.println("3. Optimize Signal");

System.out.println("4. Check Signal State");

System.out.println("5. Exit");

System.out.print("Choose an option: ");

int option = scanner.nextInt();

scanner.nextLine();

switch (option) {

case 1:

System.out.print("Enter intersection ID: ");

String intersectionId = scanner.nextLine();

signals.put(intersectionId, new TrafficSignal(intersectionId));

System.out.println("Intersection " + intersectionId + " added.");

break;

case 2:

System.out.print("Enter intersection ID: ");

intersectionId = scanner.nextLine();

System.out.print("Enter vehicle count: ");

int vehicleCount = scanner.nextInt();

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

int speed = scanner.nextInt();

trafficDataMap.put(intersectionId, new TrafficData(vehicleCount, speed, intersectionId));

System.out.println("Traffic data for intersection " + intersectionId + " updated.");

break;

case 3:

System.out.print("Enter intersection ID: ");

intersectionId = scanner.nextLine();

TrafficSignal signal = signals.get(intersectionId);

TrafficData data = trafficDataMap.get(intersectionId);

if (signal != null && data != null) {

signal.optimizeSignal(data);

} else {

System.out.println("Intersection ID not found or no traffic data available.");

}

break;

case 4:

System.out.print("Enter intersection ID: ");

intersectionId = scanner.nextLine();

signal = signals.get(intersectionId);

if (signal != null) {

System.out.println("Current Signal State at " + intersectionId + ": " + signal.getCurrentState());

} else {

System.out.println("Intersection ID not found.");

}

break;

case 5:

System.out.println("Exiting the Traffic Signal System. Goodbye!");

scanner.close();

return;

default:

System.out.println("Invalid option. Please choose again.");

}

}

}

}



