

Traffic Signal Simulation System

Simulating Traffic Flow with **Java Threads** and **Synchronization**



- Change the road geometry by dragging
- Click onto the road to disturb traffic flow
- Drag obstacles or construction vehicles to create new bottlenecks
- Drag traffic lights to the road and click on them to toggle between red and light
- Use the Info button repeatedly for more info

Traffic Flow and General

Density/lane	54/km
Truck Perc	10%
Timewarp	8

Car-Following Behavior

Max Accel a	0.4 m/s ²
Max Speed v0	100 km/h
Time Gap T	1.2 s
Length l	3 m
Const Decel b	3 m/s ²

Lane-Changing Behavior

LC Threshold	0.4 m/s ²
Right Bias Cars	0.05 m/s ²
Right Bias Trucks	0.2 m/s ²

Project Presentation of Exploratory Project (EP)

Traffic Signal Simulation System

Submitted by:

Sneha Chhabra (2210990856)
Harsh Dhiman (2210991612)
Jinny Kapur (2210990462)
Khushi (2210991796)

Supervised by:

Dr. Ajay Katiyar
Associate Professor,
Chitkara University, Punjab

Department of Computer Science and Engineering,
Chitkara University, Punjab

Objective

To simulate traffic lights and vehicle flow using Java threads and synchronization



Thread-based signal control



Synchronization mechanisms



Vehicle flow simulation



Real-time signal transitions



Vehicle 1



Vehicle 2



Vehicle 3

Module Overview



Signal Module

Simulate **Red/Yellow/Green** traffic signals



Vehicle Module

Represent **vehicles waiting** at signals



Controller Module

Manage **timing and transitions** between signals



Threads



Synchronization



Exception Handling

How Data Is stored



Stores simulation data in **CSV format** for analysis and debugging



Timestamp **SignalState** **VehicleCount**

2023-10-29 10:00:00 RED 5



Timestamp

2023-10-29 10:00:30 YELLOW 5



SignalState

2023-10-29 10:01:00 GREEN 3



VehicleCount

2023-10-29 10:01:30 YELLOW 2

2023-10-29 10:02:00 RED 7

Algorithmic flow



Start traffic simulation



Thread changes light every few seconds



Log signal states and vehicle data

Simulation Process



Initialize system components

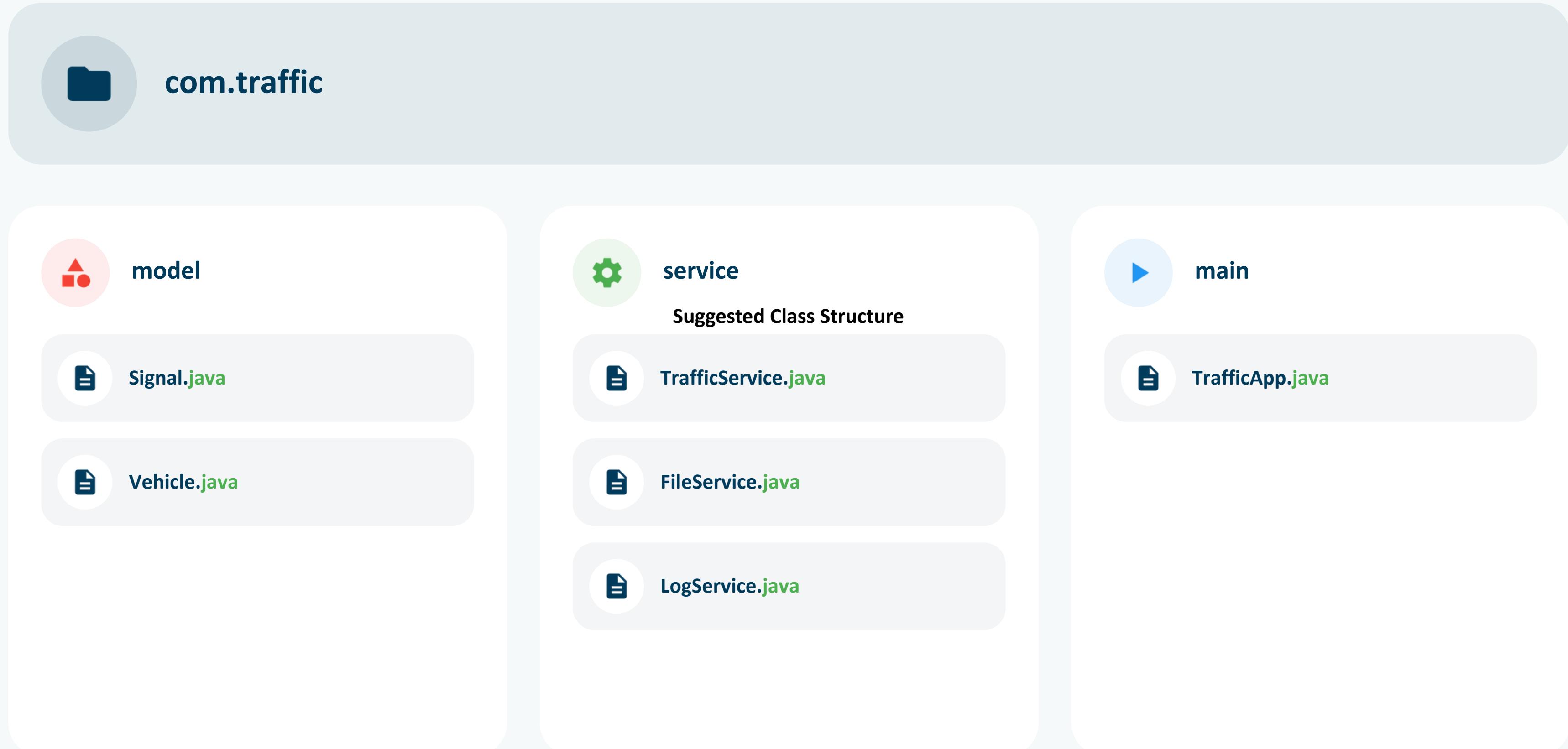


Execute signal transitions



Record simulation data

Suggested Class Structure



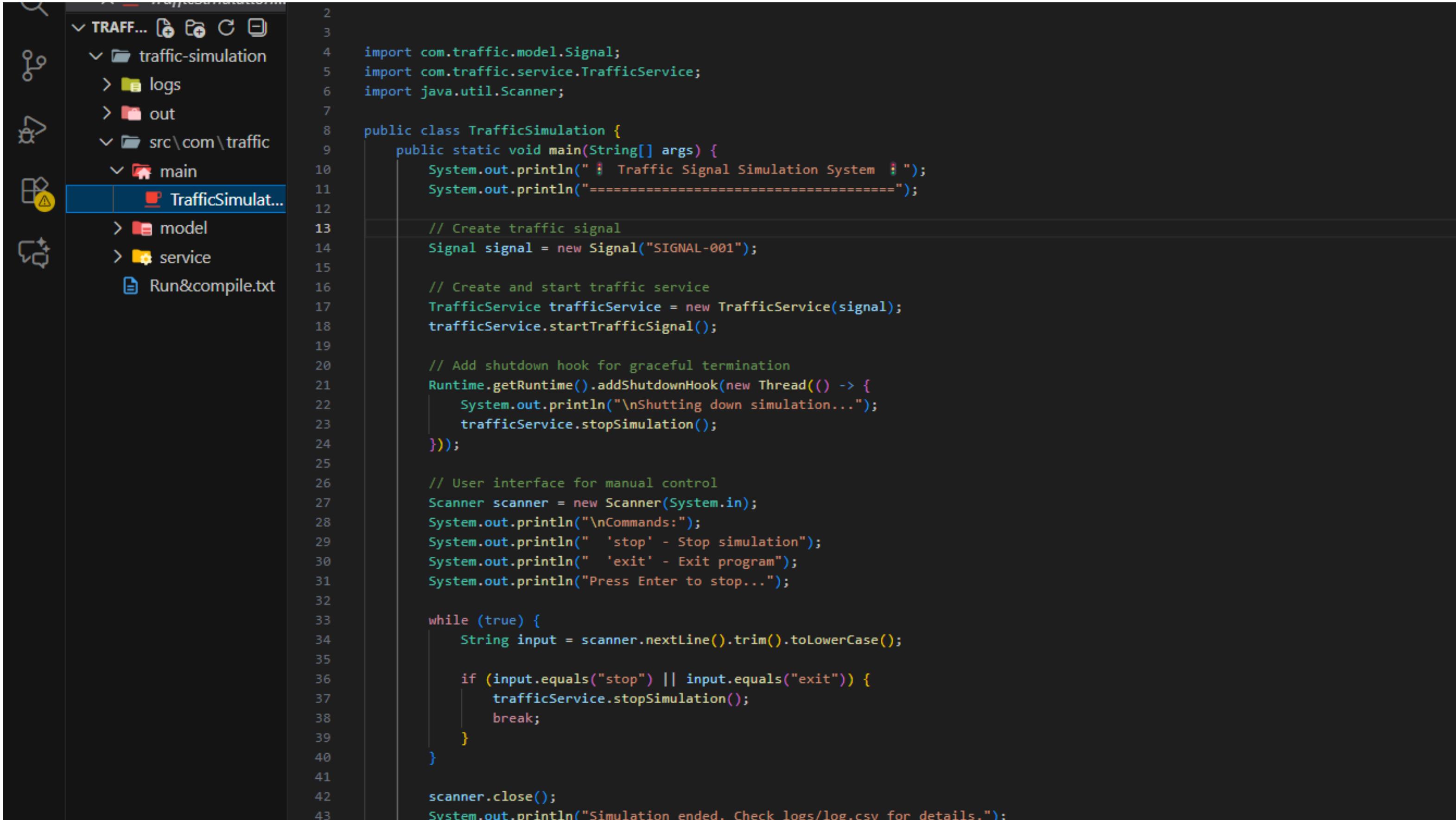
OUTPUT

```
PS C:\Users\Sneha\OneDrive\Desktop\traffic-signal-simulator> cd traffic-simulation
PS C:\Users\Sneha\OneDrive\Desktop\traffic-signal-simulator\traffic-simulation> javac -d out src/com/traffic/model/*.java src/com/traffic/service/*.java src/com/traffic/main/*.java
>>
PS C:\Users\Sneha\OneDrive\Desktop\traffic-signal-simulator\traffic-simulation> java -cp out com.traffic.main.TrafficSimulation
? Traffic Signal Simulation System ?
=====
Traffic signal simulation started...
Signal SIGNAL-001 changed to: RED

Commands:
'stop' - Stop simulation
'exit' - Exit program
Press Enter to stop...
Logged: 2025-11-19 17:03:50,RED,0
New vehicle arrived: Vehicle{vehicleId='V1', type=MOTORCYCLE} | Total vehicles: 1
New vehicle arrived: Vehicle{vehicleId='V2', type=MOTORCYCLE} | Total vehicles: 2
New vehicle arrived: Vehicle{vehicleId='V3', type=CAR} | Total vehicles: 3
Signal SIGNAL-001 changed to: GREEN
Logged: 2025-11-19 17:03:55,GREEN,3
New vehicle arrived: Vehicle{vehicleId='V4', type=BUS} | Total vehicles: 4
1 vehicle(s) departed. Remaining: 3
New vehicle arrived: Vehicle{vehicleId='V5', type=CAR} | Total vehicles: 4
3 vehicle(s) departed. Remaining: 1
New vehicle arrived: Vehicle{vehicleId='V6', type=CAR} | Total vehicles: 2
Signal SIGNAL-001 changed to: YELLOW
Logged: 2025-11-19 17:04:02,YELLOW,2
1 vehicle(s) departed. Remaining: 1
Signal SIGNAL-001 changed to: RED
Logged: 2025-11-19 17:04:04,RED,1
New vehicle arrived: Vehicle{vehicleId='V7', type=BUS} | Total vehicles: 2
stop
Stopping traffic simulation...
Simulation ended. Check logs/log.csv for details.
New vehicle arrived: Vehicle{vehicleId='V8', type=BUS} | Total vehicles: 3
Signal SIGNAL-001 changed to: GREEN
Logged: 2025-11-19 17:04:09,GREEN,3
Signal SIGNAL-001 changed to: YELLOW
Logged: 2025-11-19 17:04:16,YELLOW,3

Shutting down simulation...
Stopping traffic simulation...
```

TRAFFIC SIMULATION CODE



The screenshot shows a Java IDE interface with the following details:

- Project Structure:** The project is named "TRAFF...". It contains a "traffic-simulation" folder with "logs" and "out" subfolders. The "src\com\traffic\main" package contains a "TrafficSimulation.java" file, which is currently selected.
- Code Editor:** The main editor area displays the Java code for "TrafficSimulation". The code initializes a traffic signal, starts a traffic service, adds a shutdown hook for graceful termination, and provides a user interface for manual control via a Scanner.
- Code Lines:** The code is numbered from 1 to 43.
- Output Area:** There is no visible output or logs in the bottom pane.

```
2
3
4 import com.traffic.model.Signal;
5 import com.traffic.service.TrafficService;
6 import java.util.Scanner;
7
8 public class TrafficSimulation {
9     public static void main(String[] args) {
10         System.out.println(" * Traffic Signal Simulation System * ");
11         System.out.println("=====");
12
13         // Create traffic signal
14         Signal signal = new Signal("SIGNAL-001");
15
16         // Create and start traffic service
17         TrafficService trafficService = new TrafficService(signal);
18         trafficService.startTrafficSignal();
19
20         // Add shutdown hook for graceful termination
21         Runtime.getRuntime().addShutdownHook(new Thread(() -> {
22             System.out.println("\nShutting down simulation...");
23             trafficService.stopSimulation();
24         }));
25
26         // User interface for manual control
27         Scanner scanner = new Scanner(System.in);
28         System.out.println("\nCommands:");
29         System.out.println(" 'stop' - Stop simulation");
30         System.out.println(" 'exit' - Exit program");
31         System.out.println("Press Enter to stop...");
32
33         while (true) {
34             String input = scanner.nextLine().trim().toLowerCase();
35
36             if (input.equals("stop") || input.equals("exit")) {
37                 trafficService.stopSimulation();
38                 break;
39             }
40         }
41
42         scanner.close();
43         System.out.println("Simulation ended. Check logs/log.csv for details.");
}
```