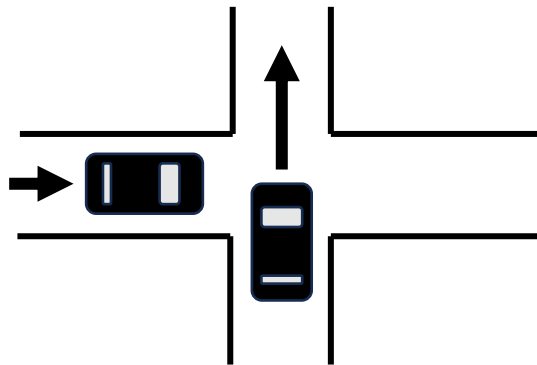


JC2002 Java Programming - Practical 8 (Day 10)

The goal of this practical is to get familiar multithreading, timed events, and synchronisation of threads. In this practical, the task is to implement a simple traffic simulator with timed events. Let us assume that there are two one-way streets crossing, one from south to north, and another one from west to east. When a car arrives in the crossroads, it proceeds, unless the crossing is already occupied by another car. We assume that it takes two seconds for one car to cross the road. When the crossing is cleared, another car waiting (if any) can enter the crossing.



1. Implement main class **TrafficSimulator** and two classes **WestArrivalThread** and **SouthArrivalThread** inherited from **Thread**, simulating cars to arrive from the west and from the south. To avoid duplicate implementation of all the functionality, you can optionally implement a joint superclass **ArrivalThread** inherited from **Thread**.

We can assume that cars from both directions arrive at random intervals between 0.5 and 4 seconds. Therefore, **run()** method of both threads should include an infinite loop that sleeps for a random time between 500 and 4000 milliseconds, and then prints text "Car arrived from the west" or "Car arrived from the south", respectively, and repeats.

Hint: you can create random integers between 0 and $n-1$ by instantiating class **java.util.Random**, and then invoking method **nextInt(n)**.

In the **main()** method of **TrafficSimulator**, instantiate and start threads for both west and south arrivals. Run the program to ensure that it works as expected.

2. Implement class **CrossingThread** inherited from **Thread** to simulate cars crossing. The thread should be instantiated and started every time when a car arrives in the crossroads either from the west or the south. Implement a synchronized public static method **cross()** in **TrafficSimulator** that sleeps for 2000 milliseconds and returns. In the **run()** method of **CrossingThread** class, **TrafficSimulator.cross()** should be invoked. By synchronization we can prevent two cars entering the crossing at the same time.
3. Finally, implement a mechanism to keep track of how many cars are queuing to the crossing from the west and from the south. For this purpose, you need to use static integer variables in **TrafficSimulator** to store the number of cars in both queues. You should increase the number of cars when a new car arrives (from the arrival threads),

and decrease the number of cars in method **cross()** after sleeping 2000 milliseconds, just before returning. Therefore, you need to make some kind of mechanism to tell method **cross()** if the arriving car comes from the west or from the south.

Run the program to test if it works properly. You should print the number of cars in the queues every time the values change. You can try to use different maximum and minimum arrival intervals to see how it affects the development of the queues.