Final exam

Due Thursday by 11:31am **Points** 52 **Submitting** a file upload **File Types** zip
Available Dec 9 at 8:30am - Dec 9 at 11:31am about 3 hours

This assignment was locked Dec 9 at 11:31am.

This exam has two parts – Part 1 and Part 2. Part2 builds upon Part1. Your submission must have both parts working for full grade. Only last submission will be graded.

Problem Statement Part1:

Write a program that simulates **Traffic** on a **Road** that has one **TrafficLight**. Traffic creates **Vehicle** at an interval anywhere from **MIN_VEHICLE_DELAY** ms to **MAX_VEHICLE_DELAY** ms and asks it to **joinVehicleQ**.

The TrafficLight has two states: Green and Red. The Road polls away vehicles when the traffic light is green. If it is red, the vehicle waits in vehicleQ and will be polled by Road when traffic light will turn green again. The TrafficLight changes color after every **TRAFFIC_LIGHT_DELAY** ms.

The program should run for **maxVehicles** as input by the user. As these vehicles come to TrafficLight, a stream of print statements is generated showing the status. At the end, the program prints the Traffic Report as shown in Fig. 1-3.

```
Part 1 or 2?
1
How many vehicles?
0
-----------TRAFFIC REPORT--------------------------
The program ran for 0 ms
Max Q length at traffic light was 0
Final Q length at traffic light was 0
```

Fig.1: Output with 0 vehicles

```
Part 1 or 2?
1
How many vehicles?
10
Green: Vehicle 1 passed. Q length: 0
Green: Vehicle 2 passed. Q length: 0
Green: Vehicle 3 passed. Q length: 0
Green: Vehicle 4 passed. Q length: 0
Green: Vehicle 5 passed. Q length: 0
Green: Vehicle 6 passed. Q length: 0
Green: Vehicle 7 passed. Q length: 0
```

```
Part 1 or 2?
1
How many vehicles?
20
Green: Vehicle 1 passed. Q length: 0
Green: Vehicle 2 passed. Q length: 0
Green: Vehicle 3 passed. Q length: 0
Green: Vehicle 4 passed. Q length: 0
Green: Vehicle 5 passed. Q length: 0
Green: Vehicle 6 passed. Q length: 0
Green: Vehicle 7 passed. Q length: 0
Green: Vehicle 8 passed. Q length: 0
Green: Vehicle 9 passed. Q length: 0
   RED: Vehicle 10 in Q. Q length 1
   RED: Vehicle 11 in Q. Q length 2
   RED: Vehicle 12 in Q. Q length 3
   RED: Vehicle 13 in Q. Q length 4
   RED: Vehicle 14 in Q. Q length 5
   RED: Vehicle 15 in Q. Q length 6
   RED: Vehicle 16 in Q. Q length 7
   RED: Vehicle 17 in Q. Q length 8
   RED: Vehicle 18 in Q. Q length 9
```

```
Green: Vehicle 8 passed. Q length: 0
   RED: Vehicle 9 in Q. Q length 1
   RED: Vehicle 10 in Q. Q length 2
-----------TRAFFIC REPORT---------------------------
The program ran for 220 ms
Max Q length at traffic light was 2
Final Q length at traffic light was 2
```

Fig. 2: Output with 10 vehicles

```
Green: Vehicle 10 passed. Q length: 8
Green: Vehicle 11 passed. Q length: 7
Green: Vehicle 12 passed. Q length: 6
Green: Vehicle 13 passed. Q length: 5
Green: Vehicle 14 passed. Q length: 4
Green: Vehicle 15 passed. Q length: 3
Green: Vehicle 16 passed. Q length: 2
Green: Vehicle 17 passed. Q length: 1
Green: Vehicle 18 passed. Q length: 0
Green: Vehicle 19 passed. Q length: 0
-----------TRAFFIC REPORT---------------------------
The program ran for 306 ms
Max Q length at traffic light was 9
Final Q length at traffic light was 1
```

Fig.3: Output with 20 vehicles

Note: The last vehicle didn't 'pass' because it just got into the queue.

Solution Design:

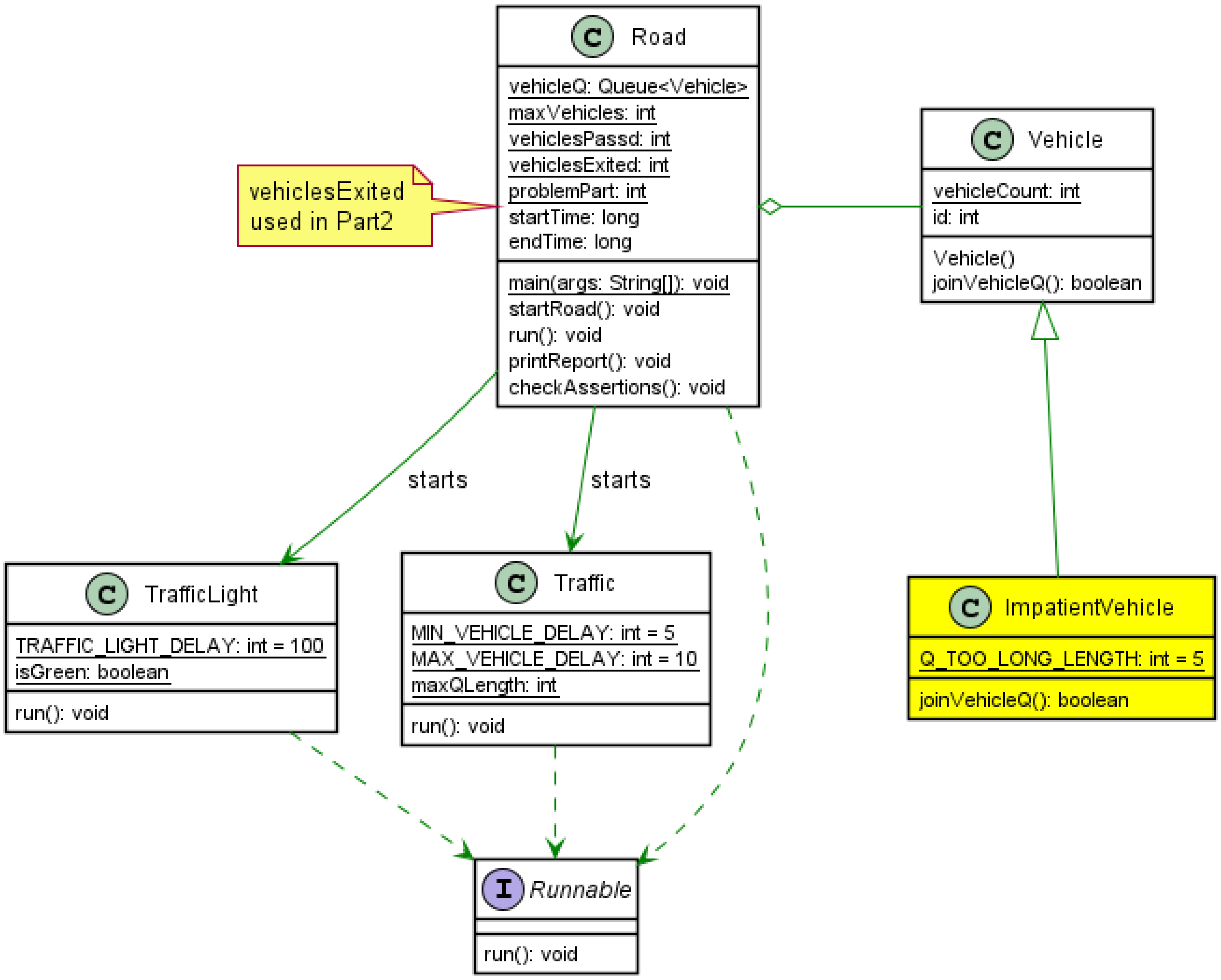

Fig.4: Class diagram

As shown in Fig. 4, TrafficLight, Traffic, and Road are Runnables. All classes are in a package named **finals.**

- **Road**: has the variable problemPart whose value is input by user to run Part1 or Part 2 of the exam.
    - main(): Insert the following code:

```
public static void main(String[] args) {
    Road road = new Road();
    road.startRoad();
    road.printReport();
    road.checkAssertions();
}
```

    - startRoad(): takes user inputs, creates the threads, and gets them started.
    - printReport(): prints the Traffic Report as shown in Fig.1-3. Your print format should be as close to the format shown here. Minor variations may be ignored provided the output is readable and well formatted.
    - run(): runs while number of vehicles is less than maxVehicles and polls vehicleQ if TrafficLight isGreen. If it isGreen, prints Green… message
    - checkAssertions(): insert the following code:

```
void checkAssertions() {
    assertEquals(maxVehicles, vehiclesPassed + vehiclesExited + vehicleQ.size());
    assertTrue(Traffic.maxQLength >= vehicleQ.size());
    assertTrue(Vehicle.vehicleCount == maxVehicles );
}
```

- **Traffic**: run(): creates new vehicles as long as vehicleCount is less than maxVehicles. It invokes their joinVehicleQ(). If TrafficLight is not green, prints Red… message
- **TrafficLight**: run(): Runs while vehicleCount is less than maxVihicles. It changes isGreen to true and false alternatively after every **TRAFFIC_LIGHT_DELAY** ms.
- **Vehicle**: In part A, joinVehicleQ() simply adds itself to the vehicleQ and returns true. This will be overridden in Part 2.

You may add new member variables or methods as needed. However, the basic design, number, and name of classes should be as shown.

Problem Part 2: The traffic simulation now needs to accommodate one change. *On average*, one out of every four vehicles is of type '**ImpatientVehicle'** that takes an alternate route *if it finds the queue length equal to or greater than* **Q_TOO_LONG_LENGTH**. In other words, an ImpatientVehicle inherits from Vehicle and overrides joinVehicleQ() in which, if the above condition is true, it doesn't join the vehicleQ.

Some screenshots shown below further explain the scenario. The print statements marked with ************ indicate ImpatientVehicles exiting the traffic. Two additional values are printed in Traffic Report at the end: Vehicles passed and Vehicles exited.

Part 1 or 2? 2 How many vehicles? 20	Part 1 or 2? 2 How many vehicles? 30

```
Green: Vehicle 1 passed. Q length: 0
Green: Vehicle 2 passed. Q length: 0
Green: Vehicle 3 passed. Q length: 0
Green: ImpatientVehicle 4 passed. Q length: 0
Green: ImpatientVehicle 5 passed. Q length: 0
Green: Vehicle 6 passed. Q length: 0
Green: ImpatientVehicle 7 passed. Q length: 0
Green: Vehicle 8 passed. Q length: 0
Green: Vehicle 9 passed. Q length: 0
Green: Vehicle 10 passed. Q length: 0
Green: Vehicle 11 passed. Q length: 0
   RED: Vehicle 12 in Q. Q length 1
   RED: Vehicle 13 in Q. Q length 2
   RED: ImpatientVehicle 14 in Q. Q length 3
   RED: Vehicle 15 in Q. Q length 4
   RED: Vehicle 16 in Q. Q length 5
   RED: Vehicle 17 in Q. Q length 6
   RED: Vehicle 18 in Q. Q length 7
   RED: Vehicle 19 in Q. Q length 8
   RED: Vehicle 20 in Q. Q length 9
-----------TRAFFIC REPORT--------------------------
The program ran for 207 ms
Max Q length at traffic light was 9
Final Q length at traffic light was 9
Vehicles passed: 11
Vehicles exited: 0
```

Fig. 5: Output with 20 vehicles

```
Green: Vehicle 1 passed. Q length: 0
Green: Vehicle 2 passed. Q length: 0
Green: Vehicle 3 passed. Q length: 0
Green: Vehicle 4 passed. Q length: 0
Green: Vehicle 5 passed. Q length: 0
Green: Vehicle 6 passed. Q length: 0
Green: Vehicle 7 passed. Q length: 0
Green: Vehicle 8 passed. Q length: 0
Green: Vehicle 9 passed. Q length: 0
Green: Vehicle 10 passed. Q length: 0
Green: Vehicle 11 passed. Q length: 0
   RED: Vehicle 12 in Q. Q length 1
   RED: Vehicle 13 in Q. Q length 2
   RED: Vehicle 14 in Q. Q length 3
   RED: Vehicle 15 in Q. Q length 4
   RED: Vehicle 16 in Q. Q length 5
********* Red: ImpatientVehicle 17 exiting. Q length 5
   RED: Vehicle 18 in Q. Q length 6
   RED: Vehicle 19 in Q. Q length 7
   RED: Vehicle 20 in Q. Q length 8
********* Red: ImpatientVehicle 21 exiting. Q length 8
Green: Vehicle 12 passed. Q length: 7
Green: Vehicle 13 passed. Q length: 6
Green: Vehicle 14 passed. Q length: 5
Green: Vehicle 15 passed. Q length: 4
Green: Vehicle 16 passed. Q length: 3
Green: Vehicle 18 passed. Q length: 2
Green: Vehicle 19 passed. Q length: 1
Green: Vehicle 20 passed. Q length: 0
Green: Vehicle 22 passed. Q length: 0
Green: ImpatientVehicle 23 passed. Q length: 0
Green: Vehicle 24 passed. Q length: 0
Green: ImpatientVehicle 25 passed. Q length: 0
Green: Vehicle 26 passed. Q length: 0
Green: Vehicle 27 passed. Q length: 0
Green: ImpatientVehicle 28 passed. Q length: 0
Green: Vehicle 29 passed. Q length: 0
-----------TRAFFIC REPORT--------------------------
The program ran for 314 ms
Max Q length at traffic light was 8
Final Q length at traffic light was 1
Vehicles passed: 27
Vehicles exited: 2
```

Fig. 6: Output with 30 vehicles

	Note: The last vehicle didn't 'pass' because it just got into the queue.

Rubric	Part1	Part2
Console outputs	30	7
Correct use of synchronization, volatile, static, polymorphism, random, thread-sleep, thread-join, etc.	5	2
Print format	5	3
Total	40	12

Instructions:

- Create a package named finals and create all your Java files in it.
- Write your name and AndrewID in all your Java files
- Zip them as AndrewId-finals.zip and submit
- Any submission issues may cost you upto 2 points.

Good luck!