1.Problem specification：Using java to realize traffic simulation system. The program mainly implements simulated roads, traffic lights and cars.

2.Problem decomposition using UML class diagrams: The program mainly involves three categories, namely road, car and traffic lights.

Car: The main attributes of the car category are: id(public, unique identifier id),speed(private, the current speed of the car), maxspeed(private, Maximum speed limit of the car), position(private, Car's current location), currentRoad(private, The road the car is currently on) It main methods are:

public void gotoNextRoad():Change the road the car is currently driving on; public void move() :Simulate car driving. When the car is moving forward if it encounters a red traffic light, it waits; if there is no road to drive, it stops; the rest will follow the speed of the car. Regarding the speed of the car, a smaller use will be taken according to the maximum speed of the road and the maximum speed of the car itself.

There are also two classes inherited from this class: Bus and Motorbike. These two classes rewrite the id and maximum speed, which also shows that different types of cars in real life will have different speed limits. The id is changed to facilitate viewing when printing.

TrafficLight: The main attributes of the trafficLight category are: id(private),state(private, current traffic light status),position(private, where the traffic lights are), roadAttachedTo(which road this trafficlight belong) It main methods are:

public void operate():The status of traffic lights is randomly generated during initialization. Use radom object obtains a random number, if it is less than 0.5, it is a green light, otherwise, it is set to a red light. chageState():Change the status of traffic lights, invert red and green;

Road: The road class mainly describes roads in practice. Its private fields are: id(Used to identify the road), speedLimit(Speed limit on the road), length(Road length),carsOnRoad(a list of cars on this road), lightsOnRoad(a list of lights on this road), connectedRoads(a list store road that connected this road). It main methods are:

public boolean canPass(int postion):Describe whether a location can pass, mainly to determine whether there is a traffic light at that location, and if so, to determine whether it can pass according to the color of the traffic light.

public void addCar(Car car):add a car to this road public void addTrafficLight(TrafficLight trafficLight):add trafficlight to this road public Road getRandomRoad()：return a random road connected of this road

The road class is mainly a carrier that holds traffic lights and cars driving on the road. He will call the car's move method, and also determine whether the car can pass a certain traffic light. At the same time, when the car runs to the end, it needs to get the next road to provide the car to continue running.

Main: The setting of the main class mainly realizes traffic simulation. The simulation process is the driving of multiple connected roads and multiple cars (including two types).

First, the user is required to enter the number of roads. Then get the information of each road in turn, including the length of the road and the roads connected to it.

Then instantiate each road object, and then add to each road according to the road connection. Set up traffic lights where the roads are connected first, and initialize the colors.

After that, a specified number of cars are randomly generated (the number is entered by the user). The car will also randomly generate two types: Bus and Motorbike.

After that, it will run for a fixed number of seconds. The operation in every second is: the traffic light changes state and the car is driving. If all cars are stopped before the specified number of seconds arrives, the program will also exit.