

IOOM ASSIGNMENT 3

Name- Suruchi Shrey

Enrollment no. -BT18CSE014

Files-

Interfaces- Truck.java, Tollbooth.java

Classes- BarCode.java, Bill.java, Receipt.java, TruckType_1.java, TruckType_2.java, NagpurTollBooth.java

Class extending Thread- StoreReceipts.java

Driver file- ExecuteTollBooth.java

Description-

I have made a system in which the data for trucks ,i.e.,their weights and axles come from a text file(inputTrucks.txt). Trucks keep on arriving at varied intervals of time like first truck arrives after 2 secs and then second truck arrives after 5 secs when the first truck leaves, meanwhile I have kept a timer which keeps storing the data after every 10 seconds(just for demonstration the interval has been kept this short) into a file named TruckReceipts.txt. After the arrival of a truck, its toll due and details are shown on the screen and the options for display data , reset and none are given, one of them which the user must enter to proceed(you may enter none if you don't want to anything). At the end of the program the details of receipts stored in the booth from date 01-12-2020 to 03-12-2020 have been retrieved from the file and shown. The runtime for the timer has been kept 6 mins(for demonstration) which is enough for execution of given trucks and their arrival times, so the user can either wait till 6 mins are over or can stop the execution, if he/she wants to end it before.

Details of individual files-

Truck.java- This is an interface having functions for Truck objects.

Tollbooth.java- This is an interface which has abstract functions-

```
public double calculateToll(Truck truck);  
public void displayData();  
public void cashRemoval();  
public void storeReceipt();  
public void datewiseCollection(String d1,String d2);
```

BarCode.java- This is a class representing the barcode to be scanned which has truckid and axles as **final** (so they cannot be modified) **private** data members and the getters and setters as member functions.

Bill.java- This class is representing a bill, which contains the weight of the truck.

TruckType_1.java- This is a class implementing Truck interface representing a **make** of truck, consisting of a Barcode and a Bill as its data members and a static member id(initialised to 0) which autoincrements by 2 and becomes the truckid of the new truck.

TruckType_2.java- This is a class implementing the Truck interface a **make** of truck, consisting of a Barcode and a Bill as its data members and a static member id(initialised to 1) which autoincrements by 2 and becomes the truckid of the new truck.

NagpurTollBooth.java- This is a class implementing the Tollbooth interface, which implements all the functions of the interface.

Receipt.java- This is a class representing a receipt, which consists of truckid, receiptid, amount(toll paid), date(on which this receipt was generated), time(at which this receipt was generated).

StoreReceipts.java- This is a class extending Thread class whose run function starts a timer for storing the receipts after every 10 seconds into a text file(TruckReceipts.txt). So this keeps running along with main.

ExecuteTollBooth.java- This is the driver class which consists of the main function, this declares an object of NagpurTollBooth class, starts the StoreReceipts thread, so from the inputTrucks text file a row is taken and the main thread sleeps until the truck arrives(for ex, for first truck it sleeps for 2 secs), when the truck arrives thread wakes up and a message is displayed and its toll is calculated. Three options are shown for displaying data, resetting(cash removal) and resuming, **display data** displays the no. of trucks and total receipt collection since last collection and **resetting option** resets the variables. The receipts(having amount, date, time and truckid) are being stored into the file parallelly. This cycle happens until all the trucks have

arrived. Then at last receipts collected from date 01-12-202 to 03-12-2020 are shown(some details were put into the text file manually for demonstration).

Text Files-

inputTrucks.txt- This text file contains some rows of data having (milliseconds(for truck arrival), truckid, total weight, axles) in each row for different trucks.

TruckReceipts.txt- This text file contains the stored details. This is where details are stored into.

How to execute?

Change the file paths first in storeReceipt(), datewiseCollection() and in ExecuteTollBooth.java file.

Run **javac ExecuteTollBooth.java**, and then **java ExecuteTollBooth** .