**Filename: Car.java**

public class Car extends Vehicle{

int NoOfDoors;

String Color;

static int NoOfCars = 0;

Car(){

NoOfCars+=1;

super.Type = "Car";

}

public void setIDPlate(String IDPlate) {

System.out.println(IDPlate);

}

public void setBrand(String Brand) {

System.out.println(Brand);

}

public void setTimeDate(String TimeDate) {

System.out.println(TimeDate);

}

public void setNoOfDoors(int NoOfDoors) {

this.NoOfDoors = NoOfDoors;

}

public void setColor(String Color) {

this.Color = Color;

}

public String toString() {

return (super.toString() +

"\nNo. of Doors:" + NoOfDoors +"\nColor: " + Color); }

}

**Filename :  Van.java**

public class Van extends Vehicle{

static int NoOfVans = 0;

int NoOfSeats;

Van(){

NoOfVans+=1;

super.Type = "Van"; }

public void setIDPlate(String IDPlate) {

super.IDPlate = IDPlate; }

public void setBrand(String Brand) {

super.Brand = Brand; }

public void setTimeDate(String TimeDate) {

super.DateTime = TimeDate; }

public void setNoOfSeats(int NoOfSeats) {

this.NoOfSeats = NoOfSeats; }

public String toString() {

return (super.toString() +"\nNo. of Seats: " + NoOfSeats);

} }

**Filename : ThreeWheeler.java**

public class ThreeWheeler extends Vehicle{

Boolean isTaxi;

static int NoOfThreeWheelers = 0;

ThreeWheeler(){

NoOfThreeWheelers+=1;

super.Type = "ThreeWheeler"; }

public void setIDPlate(String IDPlate) {

super.IDPlate = IDPlate; }

public void setBrand(String Brand) {

super.Brand = Brand; }

public void setTimeDate(String TimeDate) {

super.DateTime = TimeDate; }

public void isTaxi(boolean isTaxi) {

this.isTaxi = isTaxi; }

public String toString() {

return (super.toString() + "\nIs Taxi: " + isTaxi);

}

}

**Filename :  CarParkManager.java**

public interface CarParkManager {

final int totalVehicleSlots = 20;

boolean vehicleEntered(String vehicleType);

int vehicleLeft(String IDPlate);

int getNumEmptySlots();

int getNumOccupiedSlots(); }

**Filename :  Main.java**

public class Main {

public static void main(String[] args) {

SLIITCarParkManager CarManager = new SLIITCarParkManager();

CarManager.vehicleEntered("car");

CarManager.setInfo("AB-5390", "Tesla", "White", 4);

CarManager.vehicleEntered("car");

CarManager.setInfo("KL-3456", "BMW", “Asbh", 4); CarManager.vehicleEntered("threeWheeler"); CarManager.setInfo("ABB-6868", "Honda", false); CarManager.vehicleLeft("AB-5390"); CarManager.vehicleEntered("van");

CarManager.setInfo("UVX-9221", "Dolphine", 12); CarManager.vehicleEntered("car");

CarManager.setInfo("UVX-Sri 7890", "Corrolla", 10); CarManager.printAll();

}

}

**Filename :  SLIITCarParkManager.java**

import java.util.Date;

public class SLIITCarParkManager implements CarParkManager{ VehicleList vehicleList = new VehicleList();

Date dateTime = new Date();

public boolean vehicleEntered(String vehicleType) { if(getNumEmptySlots()>0) {

switch (vehicleType.toUpperCase()) {

case "CAR":

vehicleList.addCar();

System.out.println("-Entering a Vehicle-"); System.out.println(getNumEmptySlots()+ 1 +" parking slots are available for entered vehicle");

return true;

case "VAN":

vehicleList.addVan();

System.out.println("-Entering a Vehicle-"); System.out.println(getNumEmptySlots()+ 1 +" parking slots are available for entered vehicle ");

return true;

case "THREEWHEELER": vehicleList.addThreeWheeler(); System.out.println("-Entering a Vehicle-"); System.out.println(getNumEmptySlots()+ 1 +" parking slots are available for entered vehicle");

return true;

default:

System.out.println("Invalid input type..."); return false; }

} else {

System.out.println("There are No parking slots available for entered vehicle! \n");

return false;}

}

public void setInfo(String IDPlate, String Brand, String Color, int NoOfDoors) {

try {

vehicleList.carList[vehicleList.carSize()-1].setIDPlate(IDPlate); vehicleList.carList[vehicleList.carSize()-1].setBrand(Brand); vehicleList.carList[vehicleList.carSize()-1].setColor(Color); vehicleList.carList[vehicleList.carSize()-1].setNoOfDoors(NoOfDoors); vehicleList.carList[vehicleList.carSize()-1].setTimeDate(dateTime.toString()); System.out.println(vehicleList.carList[vehicleList.carSize()-1].toString()); System.out.println("\n"); }catch(ArrayIndexOutOfBoundsException e) {} }

public void setInfo(String IDPlate, String Brand, int NoOfSeats) { try {

vehicleList.vanList[vehicleList.vanSize()-1].setIDPlate(IDPlate); vehicleList.vanList[vehicleList.vanSize()-1].setBrand(Brand); vehicleList.vanList[vehicleList.vanSize()-1].setNoOfSeats(NoOfSeats); vehicleList.vanList[vehicleList.vanSize()-1].setTimeDate(dateTime.toString()); System.out.println(vehicleList.vanList[vehicleList.vanSize()-1].toString()); System.out.println("\n"); }catch(ArrayIndexOutOfBoundsException e) {}}

public void setInfo(String IDPlate, String Brand, boolean isTaxi) { try { vehicleList.threewheelerList[vehicleList.threewheelerSize()-1].setIDPlate(IDPlate); vehicleList.threewheelerList[vehicleList.threewheelerSize()-1].setBrand(Brand); vehicleList.threewheelerList[vehicleList.threewheelerSize()-1].isTaxi(isTaxi); vehicleList.threewheelerList[vehicleList.threewheelerSize()-1].setTimeDate(dateTime.toString()); System.out.println(vehicleList.threewheelerList[vehicleList.threewheelerSize()-1].toString());

System.out.println("\n"); }catch(ArrayIndexOutOfBoundsException e) {} }

public int vehicleLeft(String IDPlate) {

for(int i=0; i<vehicleList.carSize(); i++) { if(vehicleList.carList[i].IDPlate == IDPlate) { System.out.println("-Vehicle Leaving-"); System.out.println(vehicleList.carList[i].toString()); vehicleList.deleteCar(i); System.out.println("Number of Free Slots: "+getNumEmptySlots()); System.out.println("\n");

return getNumEmptySlots();}

}

for(int i=0; i<vehicleList.vanSize(); i++) { if(vehicleList.vanList[i].IDPlate == IDPlate) { System.out.println("-Vehicle Leaving-"); System.out.println(vehicleList.vanList[i].toString()); vehicleList.deleteVan(i); System.out.println("Free Slots : "+getNumEmptySlots()); System.out.println("\n");

return getNumEmptySlots();}

for(int i=0; i<vehicleList.threewheelerSize(); i++) { if(vehicleList.threewheelerList[i].IDPlate == IDPlate) { System.out.println("-Vehicle Leaving-"); System.out.println(vehicleList.threewheelerList[i].toString()); vehicleList.deleteThreeWheeler(i); System.out.println("Free Slots : "+getNumEmptySlots()); System.out.println("\n");

return getNumEmptySlots(); } } System.out.println("Invalid ID, please check and re-enter"); return getNumEmptySlots();}

public int getNumEmptySlots() {

return 20-Vehicle.NoOfVehicles; }

public int getNumOccupiedSlots() {

return Vehicle.NoOfVehicles; }

public void printAll() {

System.out.println("-Vehicle List-");

System.out.println("-------------------------------------------------------");

for(int i=0; i<vehicleList.carSize(); i++) { System.out.println(vehicleList.carList[i].toString()); System.out.println("-------------------------------------------------------"); }

for(int i=0; i<vehicleList.vanSize(); i++) { System.out.println(vehicleList.vanList[i].toString()); System.out.println("-------------------------------------------------------"); }

for(int i=0; i<vehicleList.threewheelerSize(); i++) { System.out.println(vehicleList.threewheelerList[i].toString()); System.out.println("-------------------------------------------------------");}

}

}

**Filename :  Vehicle.java**

public abstract class Vehicle {

String IDPlate;

String Type;

String Brand;

String DateTime;

static int NoOfVehicles=0;

Vehicle(){

NoOfVehicles+=1; }

abstract void setIDPlate(String IDPlate);

abstract void setBrand(String Brand);

abstract void setTimeDate(String TimeDate);

public String toString() {

return ("ID Plate: " + IDPlate + "\nType: " + Type + "\nBrand: " + Brand + "\nEntered Time: " + DateTime);

} }

**Filename : VehicleList.java**

public class VehicleList {

Car[] carList = new Car[20];

Van[] vanList = new Van[20];

ThreeWheeler[] threewheelerList = new ThreeWheeler[20];

void addCar() {

carList[Car.NoOfCars] = new Car(); }

void addVan() {

vanList[Van.NoOfVans] = new Van(); }

void addThreeWheeler() { threewheelerList[ThreeWheeler.NoOfThreeWheelers] = new ThreeWheeler(); }

void deleteCar(int i) {

for(int j=i; j<Car.NoOfCars-1; j++) {

carList[i] = carList[i+1]; } carList[Car.NoOfCars-1] = null;

Car.NoOfCars-=1;

Vehicle.NoOfVehicles-=1; }

void deleteVan(int i) {

for(int j=i; j<Van.NoOfVans-1; j++) {

vanList[i] = vanList[i+1]; } vanList[Van.NoOfVans-1] = null;

Van.NoOfVans-=1;

Vehicle.NoOfVehicles-=1; }

void deleteThreeWheeler(int i) {

for(int j=i; j<ThreeWheeler.NoOfThreeWheelers-1; j++) { threewheelerList[i] = threewheelerList[i+1]; } threewheelerList[ThreeWheeler.NoOfThreeWheelers-1] = null; ThreeWheeler.NoOfThreeWheelers-=1;

Vehicle.NoOfVehicles-=1; }

int carSize() {

return Car.NoOfCars; }

int vanSize() {

return Van.NoOfVans; }

int threewheelerSize() {

return ThreeWheeler.NoOfThreeWheelers;

} }

 The UML diagram

