/\*\*

\* Roberto Andino

\* Version 1

\* Program is done in Java programming language

\*

\* This is the tester class with the main method to run the program, it creates

\* car and parkingLot objects, a queue for a waiting list, and a stack for a

\* parking lot, the stack that acts a street to park cars, is located in the

\* parkingLot class.

\*

\* I could not include a loop to check for incorrect user input because of the

\* input the user is supposed to type in to retrieve or exit the program.

\*

\*/

package parkinglot;

import java.util.\*;

public class ParkingLotTester {

public static void main(String[] args) {

//last in first out - Parking Lot

Stack<Integer> parkingLot = new Stack<>();

//first in first out - Wait List

Queue<Integer> waitList = new PriorityQueue<>();

//for input

Scanner input = new Scanner(System.in);

//car object

car cars = new car();

//parking lot object

parkingLot lot = new parkingLot();

//for while loop and user input

int tag = 1;

//storing cars in parking lot stack

for(int i = 0; i < 11; i++)

{

parkingLot.push(cars.getRandPlate());

}

System.out.println("-------Welcome to Premium Parking------\n\n");

//to display cars in parking lot and waitlist

lot.displayLot(parkingLot, waitList);

while(tag != 0)

{

System.out.print("\n--- To park your vehicle type in 5 digit licence plate ---\n" +

"--- To retrieve your vehicle type in - followed by 5 digit license"

+ " plate ---\n" + "--- To exit type in 0 ---\n--- Type: ");

tag = input.nextInt();

//parking new car in parking lot

if(tag > 0)

{

//calling method parkCar to check if car can be parked in parking lot

lot.parkCar(cars, waitList, parkingLot, tag);

//to display cars in parking lot and waitlist

lot.displayLot(parkingLot, waitList);

}

else if(tag < 0) //retrieving car from parking lot and check if there

//is space to park car from waitList

{

//turning - number to +

int posTag = Math.abs(tag);

//calling retrieve method from parkingLot class to retrieve car

lot.retrieveCar(parkingLot, posTag);

//calling waitList method from parkinglot class to check if car

//on waitlist can be parked inside

lot.waitList(waitList, parkingLot, tag);

//to display cars in parking lot and waitlist

lot.displayLot(parkingLot, waitList);

}

}

System.out.println("\n\n----GOODBYE----\n\n");

}

}

//end of tester class

/\*\*

\* Roberto Andino

\* Version 1

\* Program is done in Java programming language

\*

\* This is the parking lot class that simulates a parking lot The class has

\* a method to display the cars in the parking lot and wait list, a method to

\* retrieve car specified by user, and a method to add a car to the parking lot.

\*

\*/

package parkinglot;

import java.util.\*;

public class parkingLot {

//constructor

public parkingLot()

{

}

//method to check if car can be parked in parking lot

public void parkCar(car cars, Queue <Integer> waitList, Stack <Integer> parkingLot, int car)

{

int randDecal = cars.getDecal();

//check decal number is between 1-15

if(randDecal < 16 && randDecal > 1)

{

if(parkingLot.size() < 16) //parking car in garage if there is space

{

System.out.println("\nYou have the correct decal: " + randDecal);

System.out.println("Your car with license plate: " + car +

" has succesfully been parked\n");

//creating car

cars.setPlate(car);

//parking car in parking lot

parkingLot.push(cars.getPlate());

}else //if there no space to park the car with correct decal, car is added to waitList

{

System.out.println("\nYou have the correct decal: " + randDecal

+ "\nUnfortunately the parking lot is currently full \n"

+"You car will be added to a waiting list\n");

//adding car to waitlist

waitList.add(car);

}

}else //if decal is not between 1-15, car if not parked

{

System.out.println("\nYou do not have the correct decal\n" +

"Your decal # must be between 1-15\n" +

"Your decal is: " + randDecal + "\n");

}

}

//method to retrieve car from parking lot

public void retrieveCar(Stack <Integer> parkingLot, int car)

{

Stack<Integer> street = new Stack<>(); //stack to temporaly park cars

//in the street

int parkingLotSize = parkingLot.size(); //getting stack size

//if else statement to check if car is in parking lot

if(parkingLot.contains(car))

{

//for loop to check and remove specied car

for(int i = 0; i < parkingLotSize; i++)

{

//check if item on top of stack == item to be removed

if(parkingLot.peek() == car)

{

parkingLot.pop();

}

else //if car does not = car user is looking for, the car on top

//is temporaly parked on the street

{

street.push(parkingLot.peek());

parkingLot.pop();

}

}

int streetSize = street.size(); //getting stack size

//moving cars back in the parkinglot

for(int x = 0; x < streetSize; x++)

{

parkingLot.push(street.peek());

street.pop();

}

System.out.println("\n\nCar with Licence plate " + car + " has been removed\n");

}else

{

System.out.println("\nCar with license plate: " + car + " is not in"

+ " parking lot\n");

}

}

//method to move car from waitlist to parking lot if there is an open space

public void waitList(Queue <Integer> waitList, Stack <Integer> parkingLot, int car)

{

if(parkingLot.size() < 16) //parking car in garage if there is space

{

if(waitList.size() > 0)

{

System.out.println("Car with License plate: " + waitList.peek() +

" on wait List has been parked in parking lot\n");

parkingLot.push(waitList.element());

waitList.remove();

}

}

}

//method to display cars in parking lot

public void displayLot(Stack <Integer> parkingLot, Queue <Integer> waitList)

{

System.out.println("Current cars in the parking lot: ");

for(int i = 1; i < parkingLot.size() ; i++)

{

System.out.println("Lot " + i + ": " + parkingLot.get(i));

}

System.out.println("\nCars on WaitList: ");

if(waitList.size() <= 0)

{

System.out.println("\nNo cars on wait list\n");

}

else

{

if(waitList.size() == 1)

{

System.out.println(waitList.peek());

}

else

{

System.out.println(waitList);

}

}

}

}

//end of parkinglot class

/\*\*

\* Roberto Andino

\* Version 1

\* Program is done in Java programming language

\*

\* This is the car class to simulate a car with a random decal and random plate,

\* it had a method to create a random decal, a method to create a random plate,

\* and two methods to set and return a plate set by the user.

\*

\*/

package parkinglot;

import java.util.Random;

public class car {

private int plate = 0;

private int decal = 0;

//constructor

public car()

{

}

//getter methods

//returns random plate number

public int getRandPlate()

{

Random rand = new Random();

//generate random plate number beween 10000-20000

int ranPlate = rand.nextInt(20000) + 10000 ;

return ranPlate;

}

//returns random decal number

public int getDecal()

{

//to create random number

Random rand = new Random();

//generate random number between 1-20

decal = rand.nextInt(20) + 1;

return decal;

}

//return plate set by user

public int getPlate()

{

return plate;

}

//settet methods

//setting plate set by user

public void setPlate(int newPlate)

{

plate = newPlate;

}

}

//end of car class