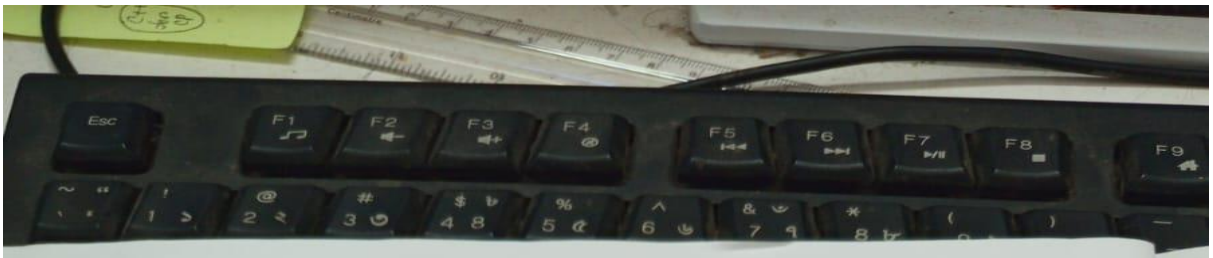

```
public class ParkingAgent extends Thread {
    private final ParkingQueue pool;
    private final String agentName;
    public ParkingAgent (String agentname, ParkingQueue pool){
        this.agentName = agentName;
        this.pool = pool;
    }
    @Override
    public void run(){
      while (true){
        RegisterParking request = pool.getParkingRequest();
        if (request != NULL){
          System.out.println(agentName + " is processing car"
                              + request.getCarId());
          request.process();
        }
      }
    }
}
```

```
public class RegistarParking {
    private static int count = 1;
    private final int carID;
    public RegistarParking() {
        this.carID = count++;
    }
    public int getCarID() {
        return carID;
    }
    public void process() {
        System.out.println("Car " + carID + " is being parked.");
        try {
            Thread.sleep(1000);
        }
        catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
        System.out.println("Car " + carID + " has been parked.\n");
    }
}
```

```
import java.util.LinkedList;
import java.util.Queue;
public class ParkingQueue{
  private final Queue<RegistanParking> queue = new
                                        LinkedList<>();

  public synchronized void addParkingRequest
                        (RegistanParking request ) {
    queue.offer (request);
    notifyAll ();
  }
public synchronized Registanparking getParkingRequest()
    while (queue.isEmpty ()){
      try{
          wait ();
        } catch (InterruptedException e){
            Thread.currentThread().interrupt ();
            return null;
          }
    }
    return queue.poll ();
  }
}
```

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        ParkingQueue pool = new ParkingQueue();

        new ParkingAgent("Agent-1", pool).start();
        new ParkingAgent("Agent-2", pool).start();
        new ParkingAgent("Agent-3", pool).start();

        Scanner sc = new Scanner(System.in);
        int requestCount = 0;
        while(requestCount < 10){
            System.out.println("\n Press Enter to simulate car
                                arrival(type -1 to exit) : ");
            String input = sc.nextLine();
            if(input.equals("-1")){
                System.out.println("Exiting...");
                break;
            }
            RegistarParking parkingRequest = new RegistarParking();
            pool.addParkingRequest(ParkingRequest);
            System.out.println("Car" + parkingRequest.getCarID() +
                               "has arrived for parking.\n");
            requestCount++;
        }
        Scanner.close();
    }
}
```