10 Floor

Floor : 1000 slots

Slot : 1 car

Same slot -> Sa, Sb : bike parked

Realtor:

Parking lots - Huge

---------------------------------------------------

High Level class diagram

Code:

1. AllocateSlot(Vehicle type(car or bike), vehicle number)
2. DeallocateSlot(Vehicle number)

---------------------------------------------------

Vehicle : VehicleNumber, VehicleType, Ticket

ParkingSpace: Id, Address, LinkedList<ParkingFloor>

AllocateSlot(VehicleNumber)

DeallocateSlot(VehicleNumber)

isFree()

ParkingFloor: FloorNo, FloorStatus, LinkedList<ParkingSlot>,

isFree(VehicleType),

getOccupiedSlots(VehicleType)

ParkingSlot : Id(name+floor), Name, Floor, SlotStatus

isFree(VehicleType),

Ticket: VehicleNumber, EntryTime, ParkingSlot, ExitTime = NULL, PaymentInfo

Payment : VehicleNumber, Hours, Rates

SlotStatus: Occupied, Free, HalfOccupied

FloorStatus: Occupied, Free

Class ParkingSpace {

numSlotsOccupied;

maxSlotsOccupied;

Boolean isFree(VehicleType) {

if( VehicleType == “Bike” ) {

Return numSlotsOccupied < maxSlotsOccupied;

Else if( VehicleType == “Car” ) {

Return MAth.round( numSlotsOccupied) < maxSlotsOccuped;

Else {

//Invalid;

}

}

Public ParkingSlot AllocateSlot(Vehicle type(car or bike), vehicle number) {

if(isFree(VehicleType) ) {

EntryTime = time.now();

Ticket ticket = new Ticket(vehicleNumber, VehicleType, EntryTime );

for(ParkingFloor floor : ParkingFloors ) {

if( floor.isFree(VehicleType) ) {

if( VehicleType == “Bike” ) {

slots.stream().filter( slot -> slot.slotStatus() == “HalfOccupied”)

}

If slots.isEmpty()

slots.stream().filter( slot -> slot.slotStatus() == “Free”);

}

}

}

//Throw Full exception

}

}