Enumerations and Data Types

<<enumeration>>
VehicleType

CAR
MOTORCYCLE
TRUCK

<<enumeration>>
SpotType

Handicapped
Compact
Large
MotorBike
Electric

<<data type>>
Location

streetAddress:string
city:string
state:string
zipcode:string
country:string

<<enumeration>>
ParkingTicketStatus

Active
Paid
Lost

Classes

ParkingSpace

parkingLots:List<ParkingLot>
city:String
zipCode: int
spotType:SpotType
vehicle:VehicleType

findParkingLots(city,zipCode):List<ParkingLot>
findParkingLots():List<ParkingLot>
fetchSpots(parkingLotId):List<SpotType>
bookSpot(parkingLotId,ParkingTicket):<ParkingTicket>
cancelSpot():bool

ParkingLot

id:string
name:string
address:Location
isFull : boolean
total_floors:int

addParkingFloor():bool
getNewParkingTicket():ParkingTicket
isFull():bool

ParkingTicket

ticketNumber:string
parkingLotId:int
parkingLotName:string
issuedAt:datetime
startTime:datetime
endTime:datetime
payedAt:datetime
payedAmount:double
status:ParkingTicketStatus
barCode:string
penalty:double

Payment

creationDate:date
amount:double
status:PaymentStatus

initiateTransaction():bool

ParkingDisplayBoard

id:string
handicappedFreeSpot:HandicappedSpot
....

showEmptySpotNumber():void

ParkingFloor

id:int
number:int

updateDisplayBoard():void
addParkingSpot():void
assignVehicleToSpot():void
freeSpot():void

ParkingSpot

id:int
number:String
free:bool
type:ParkingSlotType

isFree():bool

Vehicle

licenseNumber:string
type:VehicleType

assignTicket():void

HandicappedSlot

......

Car

.....

Relationships:
- ParkingSpace — Has an → ParkingLot
- ParkingLot ← Has — ParkingTicket
- ParkingLot — Has an → ParkingFloor
- ParkingTicket → Payment
- ParkingDisplayBoard — Has an → ParkingFloor
- ParkingFloor — Has an → ParkingSpot
- HandicappedSlot — Extends → ParkingSpot
- ...... — Extends → ParkingSpot
- ParkingSpot — has → Vehicle
- ParkingSpot — has → ParkingTicket
- Car — Extends → Vehicle
- ..... → Vehicle