[1] includes Garage - name: String [1] EntryKiosk - garageStatus: GarageStatus [1] Customer - kioskNumber: String [1] - parkingLots: EMap [1..*] - entryGate: boolean [1] - totalOccupiedSpaces: int [1] 🔁 - name: String [1] managedBy - totalUnoccupiesSpaces: int [1] - phoneNumber: String [1] + generateTicketandOpenEntryGate(in assignedParkingLot: int, in customer: Customer): EResourceSet [1] includes 📮 - emailld: String [1] [1] + enterGarage(in customer: Customer): Ticket + openEntryGate() 📮 - vehicleNumber: String [1] + closeEntryGate() 🐞 + validateTicketForExitingGarage(in ticketReferenceNumber: int, in vehicleNumber: Stri... [1] + generateReport(in choice: String, in start: EDate): EResourceSet # + Operation1() + payParkingFeeByCash(in ticket: Ticket, in amount: float): float + payParkingFeeByCard(in ticket: Ticket, in cardNumber: long, in expiryDate: EDate): b... [1] (+ findBusiestHourOfMonth(in month: EDate): int [1] + activateSensor(in choiceOfGate: String) storesTicketsIn [1] + authorizeAdmin(in userName: String, in password: String): boolean has a 💮 + stimulateTime(in testingChoice: String, in cal: EDate, in ticketNum: int) [1] Admin [1] Ticket houses / = - userName: String [1] **ExitKiosk** - assignedParkingLot: int [1] = - password: String [1] - kioskNumber: String [1] Register = - ticketReferenceNumber: int [1] - exitGate: boolean [1] - timeOfEntry: EDate [1] 🐡 + authorizeAdmin(in userName: String, in password: String): boolean - timeOfExit: EDate [0..1] + stimulateTimeForEntry(in cal: EDate, in ticketNum: int) 🐡 + addTicketToGeneratedTickets(in ticket: Ticket): boolean 🐡 + validateTicket(in ticketRefNumber: int, in vehicleNumber: String): Ticket - parkingRate: float [1] [1] accessTo + stimulateTimeForExit(in cal: EDate, in ticketNum: int) 🐡 + generateReport(in start: EDate, in choice: String): EResourceSet + openExitGate() - /parkingFee: float [0..1] + findBusiestHourOfTheMonth(in startOfMonth: EDate): int + closeExitGate() - ticketNumber: int [1] - ticketStatus: TicketStatus [1] + calculateTotalParkingFee() [*] «Enumeration» [1] records GarageStatus «Enumeration» TicketStatus ■ full available pending paid Payment 🌼 + makePaymentByCash(in parkingFee: float, in amount: float): float 🏶 + makePaymentByCard(in cardNumber: long, in expiryDate: EDate): boolean 🐡 + isValid(in number: long): boolean 🐡 + sumOfEvenPlaces(in number: long): int 🐡 + sumOfOddPlaces(in number: long): int + getDigit(in number: int): int 🌼 + hasRequiredBalance(in cardNumber: long): boolean