+ map : Map<Id ,Transportation>

+getCurrld():long

+ select All (Truck Mapper, Driver Mapper) : List < Transportation >

+ select (Truck Mapper, Driver Mapper): Transportation

+insert(long, String, String, String, int, long, long):void

+ add Transportatiob (int, String, String, String, int, int, int): void

+getTransportationsByArea(TruckMapper,DriverMapper,Area):List<Transportations>

+addTransportation(Transportation):void

Mapper - dname : String - currBranchID: int # connect() : Connection + updateIntInt(id: int, value: int, column: String, idn: String, tableName: String) : void + updateIntBoolean(id: int, value: boolean, column: String, idn: String, tableName: String): void + updateIntString(id: int, value: String, column: String, idn: String, tableName: String): void + updateIntString(id: int, value: String, column: String, idn: String, tableName: String): void + createTables() : void Extends <<Singleton>> ShiftMapper <<Singleton>> <<Singleton>> ConstraintMapper EmployeeMapper <<singleton>> TruckMapper - constraints : Map<Integer ,Constraint> - shifts : Map<Integer ,Shift> - employees : Map<Integer ,Employee> + map : Map<ld ,Truck> - defaults : Map<String ,Map<String,Integer>> - needToUpdateEmps : boolean + getNextCID(): int +selectAll():List<Truck> + insertNewShift(s: Shift): void + insertConstConstraint(c :ConstConstraint) : void + get(eid : int): Employee +select(long):Truck + insertEmpToShift(sid: int. eid : int, role: String) : void + insertTempConstraint(c :TempConstraint) : void + containsKey(eid: int) : boolean +getTrucks():List<Truck> + selectShiftsFromUntil(from: LocalDate, until: LocalDate) :List<Shift> + deleteConstraint(cid :int) : void + insert(emp: Employye, driver: boolean): boolean +addTruck(long,int,String,int,int) + selectAllConstraints() : Map<Integer,Constraint> + getShift(sid: int) : Shift + getAllBranches() : List<Integer> +getTruck(long):Truck + getConstraint(eid: int, date : LocalDate) : List<Constraint> + getShiftsOfEID(eid: int) : List<Shift> + delete(fireEID: int) : Employee + getConstraint(eid: int) : Constraint + getShiftsByDate(date: LocalDate, shiftType: String) : Shift + updateName(eid: int, newName: String) : void + getConstraintsOfEID(eid: int) : List<Constraint> + updateWasSelfMake(sid: int, wasSelfMake: boolean) : void + updateSalary(eid: int, newSalary: int) : void + deleteEmpFromShift(sid: int, eid: int, roleOfRemoved: String) : void + updateReason(cid: int, newReason: String) : void + updateBankAccountNum(eid: int, newAccountNum: int) : void + updateShiftType(cid: int, newType: String) : void + updateAmountRole(sid: int, newAmount: int, role : String) : void + updateBankBranch(eid: int, newBranch: int) : void <<singleton>> TransportationMapper + hasDefaultShifts(): boolean + updateBankID(eid: int, newBankID: int) : void + insertDefaults(defaults: Map<String ,Map<String,Integer>>): void + updateEducationFund(eid: int, newEducationFund: int) : void + getNextSID() : int + updateDaysOff(eid: int, newDaysOff: int) : void + getDefaults(): Map<String ,Map<String,Integer>> + updateSickDays(eid: int, newSickDays: int) : void + loadEmployeesInBranch() : List<Employee> + addRole(eid: int, role: String) : void + get Transportation (long, Truck Mapper, Driver Mapper): Transportation+ get Transportations (Truck Mapper, Driver Mapper) : List < Transportation >