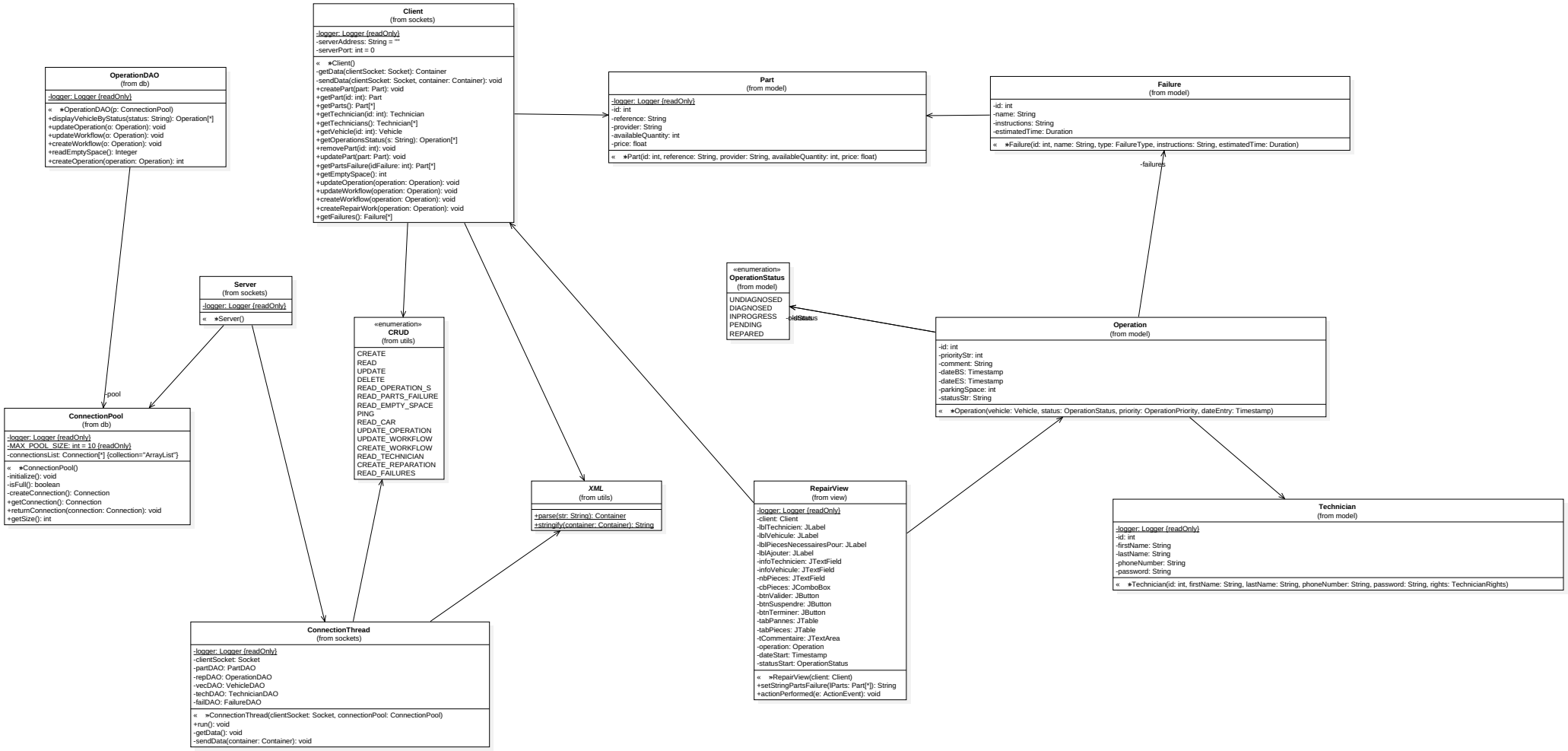
Model::Main


Client
(from sockets)
-logger: Logger {readOnly}
-serverAddress: String = ""
-serverPort: int = 0
«constructor»+Client()
-getData(clientSocket: Socket): Container
-sendData(clientSocket: Socket, container: Container): void
+createPart(part: Part): void
+getPart(id: int): Part
+getParts(): Part[*]
+getTechnician(id: int): Technician
+getTechnicians(): Technician[*]
+getVehicle(id: int): Vehicle
+getOperationsStatus(s: String): Operation[*]
+removePart(id: int): void
+updatePart(part: Part): void
+getPartsFailure(idFailure: int): Part[*]
+getEmptySpace(): int
+updateOperation(operation: Operation): void
+updateWorkflow(operation: Operation): void
+createWorkflow(operation: Operation): void
+createRepairWork(operation: Operation): void
+getFailures(): Failure[*]
OperationDAO
(from db)
-logger: Logger {readOnly}
«constructor»+OperationDAO(p: ConnectionPool)
+displayVehicleByStatus(status: String): Operation[*]
+updateOperation(o: Operation): void
+updateWorkflow(o: Operation): void
+createWorkflow(o: Operation): void
+readEmptySpace(): Integer
+createOperation(operation: Operation): int
Part
(from model)
-logger: Logger {readOnly}
-id: int
-reference: String
-provider: String
-availableQuantity: int
-price: float
«constructor»+Part(id: int, reference: String, provider: String, availableQuantity: int, price: float)
Failure
(from model)
-id: int
-name: String
-instructions: String
-estimatedTime: Duration
«constructor»+Failure(id: int, name: String, type: FailureType, instructions: String, estimatedTime: Duration)
-failures
Server
(from sockets)
-logger: Logger {readOnly}
«constructor»+Server()
«enumeration»
OperationStatus
(from model)
UNDIAGNOSED
DIAGNOSED
INPROGRESS
PENDING
REPARED
-idStatus
Operation
(from model)
-id: int
-priorityStr: int
-comment: String
-dateBS: Timestamp
-dateES: Timestamp
-parkingSpace: int
-statusStr: String
«constructor»+Operation(vehicle: Vehicle, status: OperationStatus, priority: OperationPriority, dateEntry: Timestamp)
-pool
ConnectionPool
(from db)
-logger: Logger {readOnly}
-MAX_POOL_SIZE: int = 10 {readOnly}
-connectionsList: Connection[*] {collection="ArrayList"}
«constructor»+ConnectionPool()
-initialize(): void
-isFull(): boolean
-createConnection(): Connection
+getConnection(): Connection
+returnConnection(connection: Connection): void
+getSize(): int
«enumeration»
CRUD
(from utils)
CREATE
READ
UPDATE
DELETE
READ_OPERATION_S
READ_PARTS_FAILURE
READ_EMPTY_SPACE
PING
READ_CAR
UPDATE_OPERATION
UPDATE_WORKFLOW
CREATE_WORKFLOW
READ_TECHNICIAN
CREATE_REPARATION
READ_FAILURES
XML
(from utils)
+parse(str: String): Container
+stringify(container: Container): String
RepairView
(from view)
-logger: Logger {readOnly}
-client: Client
-lblTechnicien: JLabel
-lblVehicule: JLabel
-lblPiecesNecessairesPour: JLabel
-lblAjouter: JLabel
-infoTechnicien: JTextField
-infoVehicule: JTextField
-nbPieces: JTextField
-cbPieces: JComboBox
-btnValider: JButton
-btnSuspendre: JButton
-btnTerminer: JButton
-tabPannes: JTable
-tabPieces: JTable
-tCommentaire: JTextArea
-operation: Operation
-dateStart: Timestamp
-statusStart: OperationStatus
«constructor»+RepairView(client: Client)
+setStringPartsFailure(lParts: Part[*]): String
+actionPerformed(e: ActionEvent): void
Technician
(from model)
-logger: Logger {readOnly}
-id: int
-firstName: String
-lastName: String
-phoneNumber: String
-password: String
«constructor»+Technician(id: int, firstName: String, lastName: String, phoneNumber: String, password: String, rights: TechnicianRights)
ConnectionThread
(from sockets)
-logger: Logger {readOnly}
-clientSocket: Socket
-partDAO: PartDAO
-repDAO: OperationDAO
-vecDAO: VehicleDAO
-techDAO: TechnicianDAO
-failDAO: FailureDAO
«constructor»+ConnectionThread(clientSocket: Socket, connectionPool: ConnectionPool)
+run(): void
-getData(): void
-sendData(container: Container): void