

LogManager
-Instance: LogManager
-cacheOfLoggers: Map<String, Logger>
-LogManager()
+getInstance(): LogManager
+logAll(tag:String, string:String): void
+writeLog(type:LogType,tag:String,string:String): void
+loadFromXml(xml:String): void
+loadFromFile(file:String): void
-parseFromReader(reader:Reader): void

Logger
-type: LogManager.Type
-filter: Filter
-handler: Handler
-loggers: List<Logger>
+setType(type:LogManager.Type): void
+toString(): String
+log(type:LogManager.Type,string:String): void
+getFilter(): Filter
+setFilter(filter:Filter): void
+getHandler(): Handler
+setHandler(handler:Handler): void
+setLogger(logger:Logger): void

Filter
-regex: String
-accepting: boolean
-p: Pattern
+setAccepting(accepting:boolean): void
+getRegex(): String
+setRegex(regex:String): void
+filter_message(message:String): boolean

<<interface>> Handler
+writeMessage(message:String): void

<<Handler>> FileHandler
-fileName: String
+writeMessage(message:String): void
+getFileName(): String
+setFileName(fileName:String): void

<<Handler>> BaseHandler
-baseName: String
-userName: String
-password: String
-insert: String
+writeMessage(message:String): void
+getBaseName(): String
+setBaseName(baseName:String): void
+getUserName(): String
+setUserName(userName:String): void
+getPassword(): String
+setPassword(password:String): void
+getInsert(): String
+setInsert(insert:String): void

DataBaseQueryExecutor
+executeQuery(baseName:String, user:String, password:String, query:String): boolean

