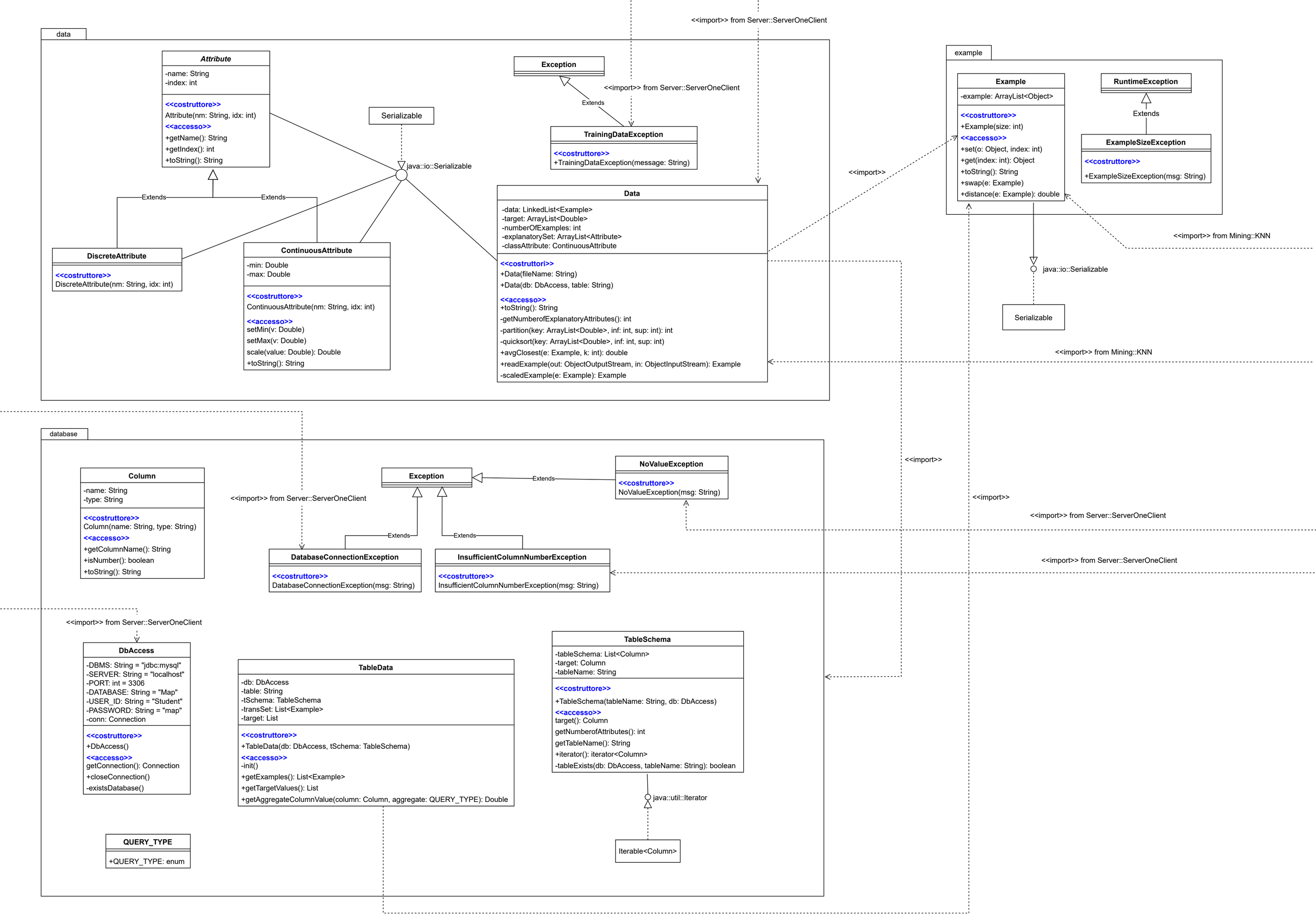
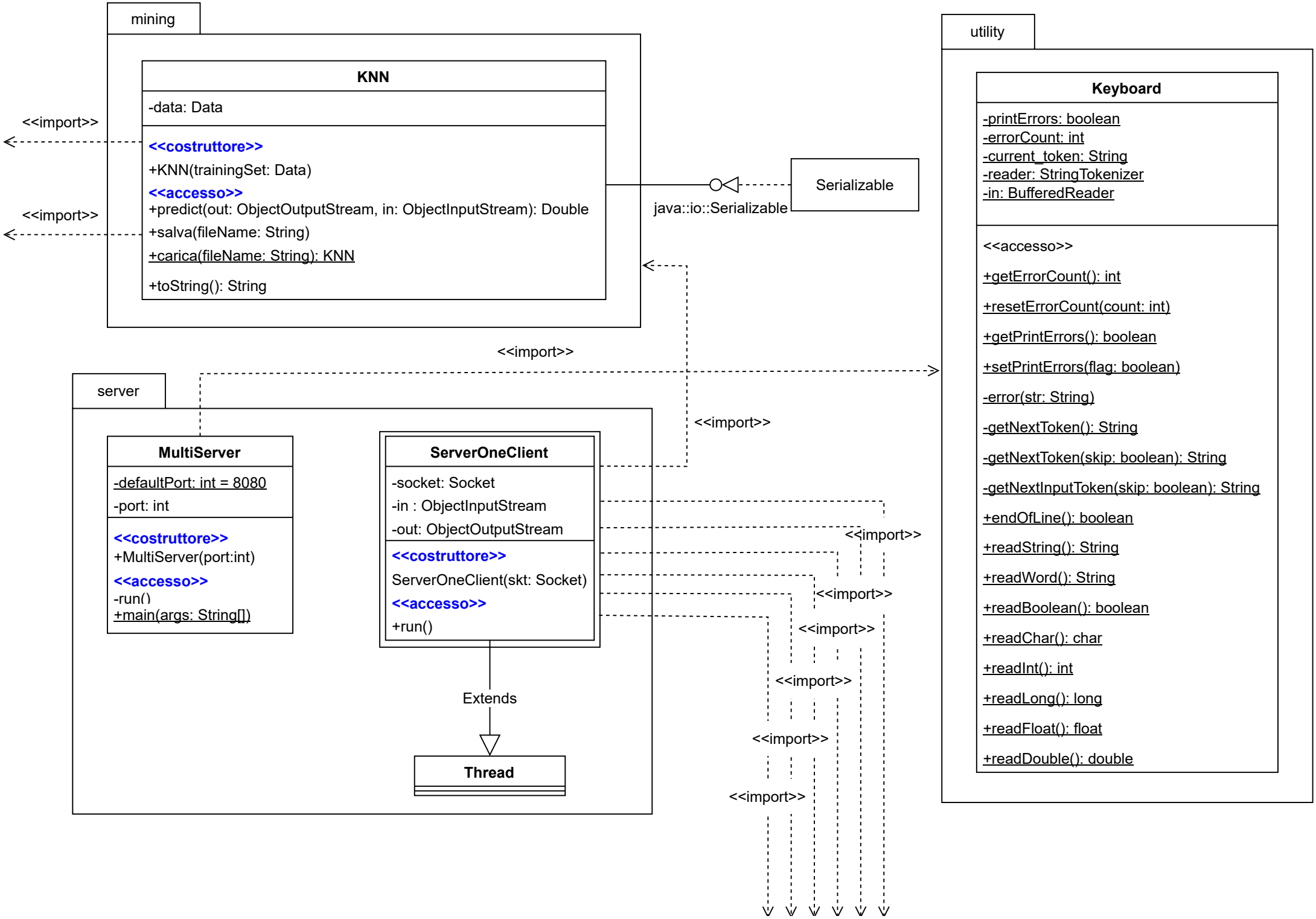

data
Attribute
-name: String
-index: int
<<costruttore>>
Attribute(nm: String, idx: int)
<<accesso>>
+getName(): String
+getIndex(): int
+toString(): String
Serializable
java::io::Serializable
Extends
Extends
DiscreteAttribute
<<costruttore>>
DiscreteAttribute(nm: String, idx: int)
ContinuousAttribute
-min: Double
-max: Double
<<costruttore>>
ContinuousAttribute(nm: String, idx: int)
<<accesso>>
setMin(v: Double)
setMax(v: Double)
scale(value: Double): Double
+toString(): String
Exception
Extends
TrainingDataException
<<costruttore>>
+TrainingDataException(message: String)
<<import>> from Server::ServerOneClient
<<import>> from Server::ServerOneClient
Data
-data: LinkedList<Example>
-target: ArrayList<Double>
-numberOfExamples: int
-explanatorySet: ArrayList<Attribute>
-classAttribute: ContinuousAttribute
<<costruttori>>
+Data(fileName: String)
+Data(db: DbAccess, table: String)
<<accesso>>
+toString(): String
-getNumberofExplanatoryAttributes(): int
-partition(key: ArrayList<Double>, inf: int, sup: int): int
-quicksort(key: ArrayList<Double>, inf: int, sup: int)
+avgClosest(e: Example, k: int): double
+readExample(out: ObjectOutputStream, in: ObjectInputStream): Example
-scaledExample(e: Example): Example
<<import>>
example
Example
-example: ArrayList<Object>
<<costruttore>>
+Example(size: int)
<<accesso>>
+set(o: Object, index: int)
+get(index: int): Object
+toString(): String
+swap(e: Example)
+distance(e: Example): double
RuntimeException
Extends
ExampleSizeException
<<costruttore>>
+ExampleSizeException(msg: String)
<<import>> from Mining::KNN
java::io::Serializable
Serializable
<<import>> from Mining::KNN
database
Column
-name: String
-type: String
<<costruttore>>
Column(name: String, type: String)
<<accesso>>
+getColumnName(): String
+isNumber(): boolean
+toString(): String
<<import>> from Server::ServerOneClient
Exception
Extends
NoValueException
<<costruttore>>
NoValueException(msg: String)
Extends
Extends
DatabaseConnectionException
<<costruttore>>
DatabaseConnectionException(msg: String)
InsufficientColumnNumberException
<<costruttore>>
InsufficientColumnNumberException(msg: String)
<<import>>
<<import>>
<<import>> from Server::ServerOneClient
<<import>> from Server::ServerOneClient
<<import>> from Server::ServerOneClient
DbAccess
-DBMS: String = "jdbc:mysql"
-SERVER: String = "localhost"
-PORT: int = 3306
-DATABASE: String = "Map"
-USER_ID: String = "Student"
-PASSWORD: String = "map"
-conn: Connection
<<costruttore>>
+DbAccess()
<<accesso>>
getConnection(): Connection
+closeConnection()
-existsDatabase()
TableData
-db: DbAccess
-table: String
-tSchema: TableSchema
-transSet: List<Example>
-target: List
<<costruttore>>
+TableData(db: DbAccess, tSchema: TableSchema)
<<accesso>>
-init()
+getExamples(): List<Example>
+getTargetValues(): List
+getAggregateColumnValue(column: Column, aggregate: QUERY_TYPE): Double
TableSchema
-tableSchema: List<Column>
-target: Column
-tableName: String
<<costruttore>>
+TableSchema(tableName: String, db: DbAccess)
<<accesso>>
target(): Column
getNumberofAttributes(): int
getTableName(): String
+iterator(): iterator<Column>
-tableExists(db: DbAccess, tableName: String): boolean
java::util::Iterator
Iterable<Column>
QUERY_TYPE
+QUERY_TYPE: enum


mining
KNN
-data: Data
<<costruttore>>
+KNN(trainingSet: Data)
<<accesso>>
+predict(out: ObjectOutputStream, in: ObjectInputStream): Double
+salva(fileName: String)
+carica(fileName: String): KNN
+toString(): String
<<import>>
<<import>>
Serializable
java::io::Serializable
utility
Keyboard
-printErrors: boolean
-errorCount: int
-current_token: String
-reader: StringTokenizer
-in: BufferedReader
<<accesso>>
+getErrorCount(): int
+resetErrorCount(count: int)
+getPrintErrors(): boolean
+setPrintErrors(flag: boolean)
-error(str: String)
-getNextToken(): String
-getNextToken(skip: boolean): String
-getNextInputToken(skip: boolean): String
+endOfLine(): boolean
+readString(): String
+readWord(): String
+readBoolean(): boolean
+readChar(): char
+readInt(): int
+readLong(): long
+readFloat(): float
+readDouble(): double
<<import>>
server
MultiServer
-defaultPort: int = 8080
-port: int
<<costruttore>>
+MultiServer(port:int)
<<accesso>>
-run()
+main(args: String[])
ServerOneClient
-socket: Socket
-in : ObjectInputStream
-out: ObjectOutputStream
<<costruttore>>
ServerOneClient(skt: Socket)
<<accesso>>
+run()
<<import>>
Extends
Thread
<<import>>
<<import>>
<<import>>
<<import>>
<<import>>