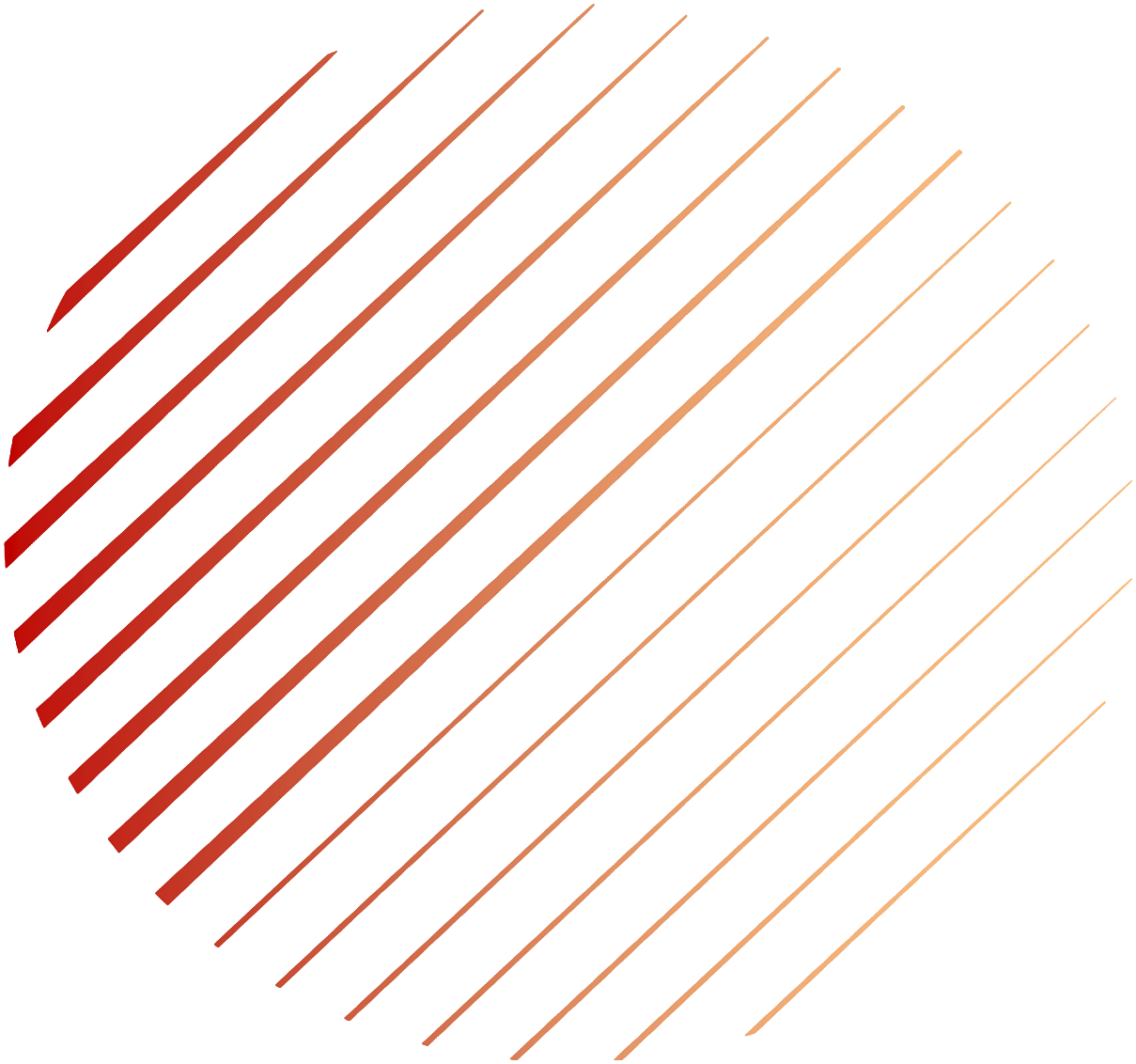
H-clus clustering

Caso di studio di Metodi Avanzati di Programmazione
AA 2023-2024



Realizzato Da
Ivan Digioia 716685
i.digioia3@studenti.uniba.it

SOMMARIO

1. INTRODUZIONE

1.1 L’algoritmo H-Clus

H-Clus è un algoritmo di clustering gerarchico progettato per creare una struttura ad albero (dendrogramma) che rappresenta la gerarchia dei cluster nei dati. A differenza degli algoritmi di clustering basati su partizioni, come K-Means, H-Clus non richiede di specificare a priori il numero di cluster e permette di visualizzare la struttura gerarchica delle relazioni tra i dati.

Origine e Caratteristiche

H-Clus è stato sviluppato per gestire dati complessi, con lo scopo di individuare relazioni gerarchiche tra le osservazioni. Questo algoritmo costruisce una gerarchia di cluster, rappresentando i dati come un albero dove ogni nodo corrisponde a un cluster e i nodi foglia rappresentano i singoli dati. H-Clus può essere utilizzato sia in modalità agglomerativa (bottom-up) che divisiva (top-down), a seconda del contesto e degli obiettivi dell'analisi.

1.2 Funzionamento dell’algoritmo

L'algoritmo H-Clus si basa principalmente sull'approccio gerarchico agglomerativo, che funziona attraverso le seguenti fasi:

1. **Inizializzazione dei Cluster**: Ogni elemento del dataset inizia come un singolo cluster individuale. La distanza tra ciascun elemento viene calcolata utilizzando metriche come la distanza euclidea, di Manhattan o altre metriche di similarità (nel progetto sarà implementata solo la distanza euclidea).

2. **Fusione dei Cluster**: In ogni iterazione, H-Clus unisce i due cluster più vicini sulla base della distanza minima. Questo processo continua fino a quando tutti gli elementi sono raggruppati in un unico cluster globale. Durante la fusione, l'algoritmo aggiorna le distanze tra i cluster utilizzando metodi come il single o average link distance.

3. **Creazione del Dendrogramma**: Durante il processo di fusione, H-Clus costruisce un dendrogramma, un grafico che rappresenta la gerarchia di tutti i cluster. Gli utenti possono scegliere il livello di taglio dell’albero per identificare il numero di cluster ottimale in base alle esigenze dell'analisi.

4. **Determinazione dei Cluster Finali**: Una volta completato il dendrogramma, l'utente può selezionare il livello appropriato di profondità dell'albero per suddividere i dati in gruppi distinti.

1.3 Limiti

Complessità Computazionale: La costruzione del dendrogramma può essere computazionalmente intensiva per dataset molto grandi, rendendo H-Clus meno adatto per big data rispetto a metodi più scalabili.

Sensibilità alle Scelte di Distanza e Linkage: I risultati dell'H-Clus possono variare significativamente a seconda della scelta delle metriche di distanza e del metodo di linkage

2. INTRODUZIONE AL PROGETTO

2.1 Descrizione del progetto

Il software realizzato utilizza l'algoritmo H-Clus, descritto nella sezione precedente, esso elabora dati da una tabella presente in un database di tipo MySQL.
Il progetto, risultato di esercitazioni, consiste in un'applicazione di tipo Client/Server.

Il server si occupa di ricevere le richieste di un client, il quale può effettuare le seguenti operazioni:
-Generare un dendrogramma partendo dai dati del database e dagli inserimenti dell'utente come la profondità e la scelta tra single o average link distance e memorizza il risultato in un file.
-Caricare da un file il dendrogramma memorizzato, si preferiscono I file in estensione '.dat' ma andrà bene un qualsiasi file come un '.txt'.

In entrambi i casi, il client dovrà specificare nei criteri di ricerca:
-la profondità per suddividere i dati.
-se operare il single o average link distance.
-il nome del file su cui salvare o caricare I dati.

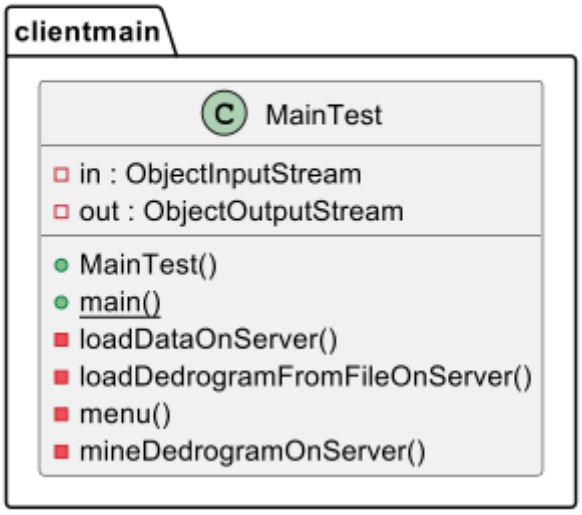
Nella sezione 3 sono riportati anche i diagrammi UML per il client e per il server. Inoltre, nella cartella "Javadoc" è stata allegata la Javadoc creata direttamente dall'IDE di sviluppo (IntelliJ). Nella sezione 5 del documento sono riportati esempi di esecuzione.

3. DIAGRAMMI UML

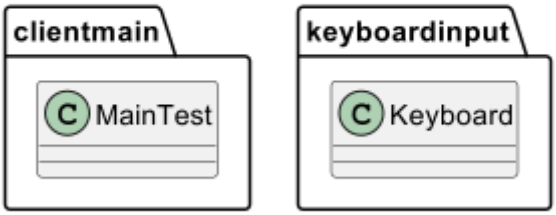
Segue la realizzazione dei diagrammi per la versione Base del HclusCleint e HClusServer

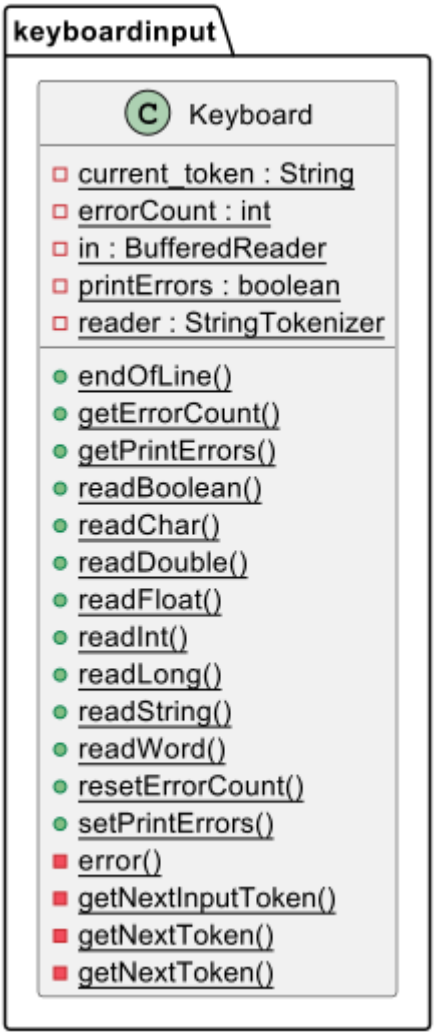
3.1 Client UML

CLIENTMAIN's Class Diagram

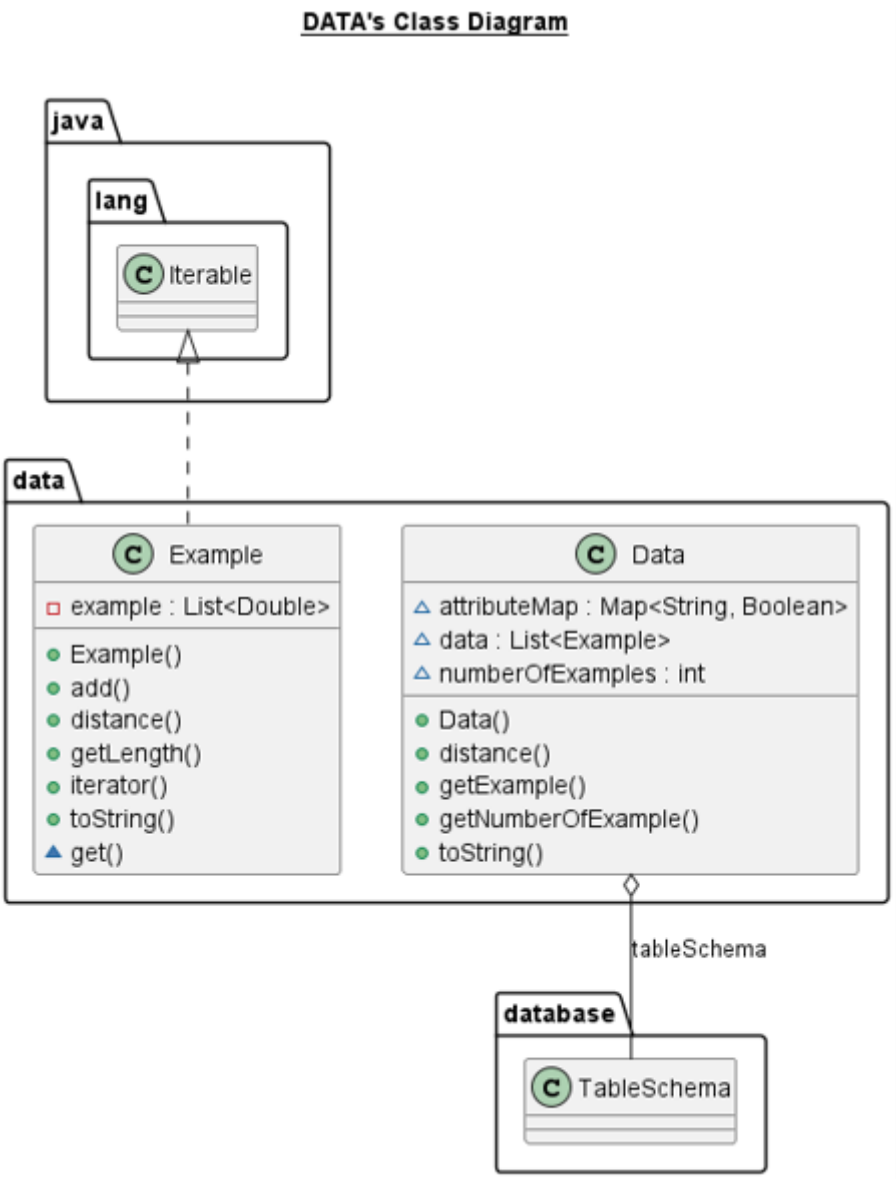


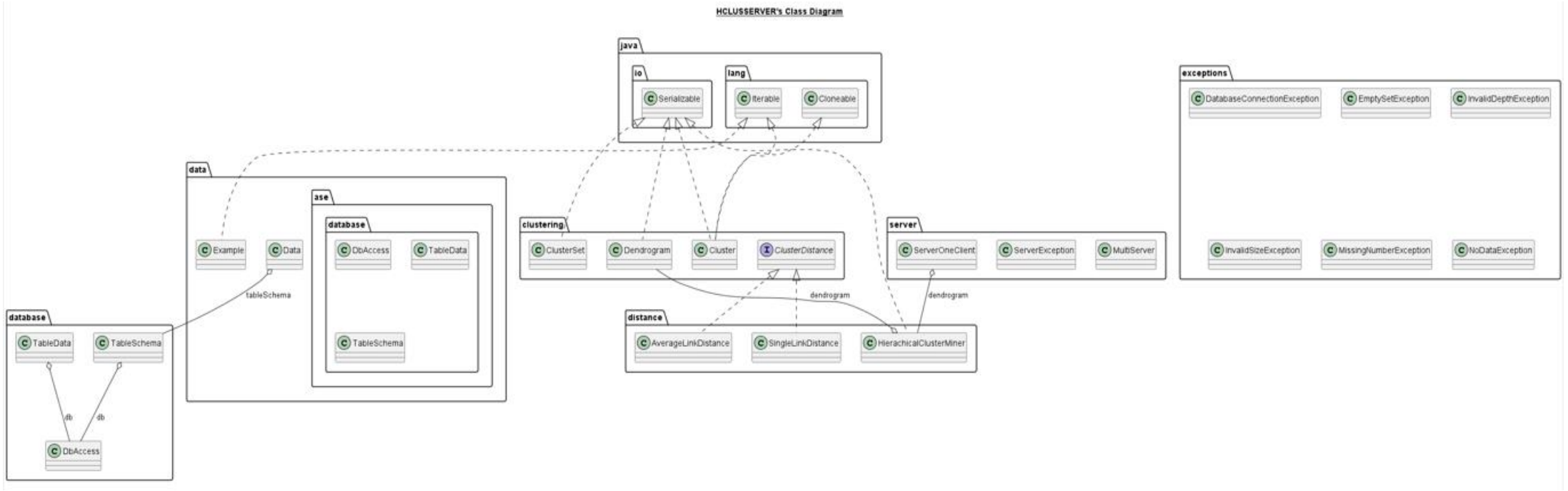
HCLUSCLIENT's Class Diagram



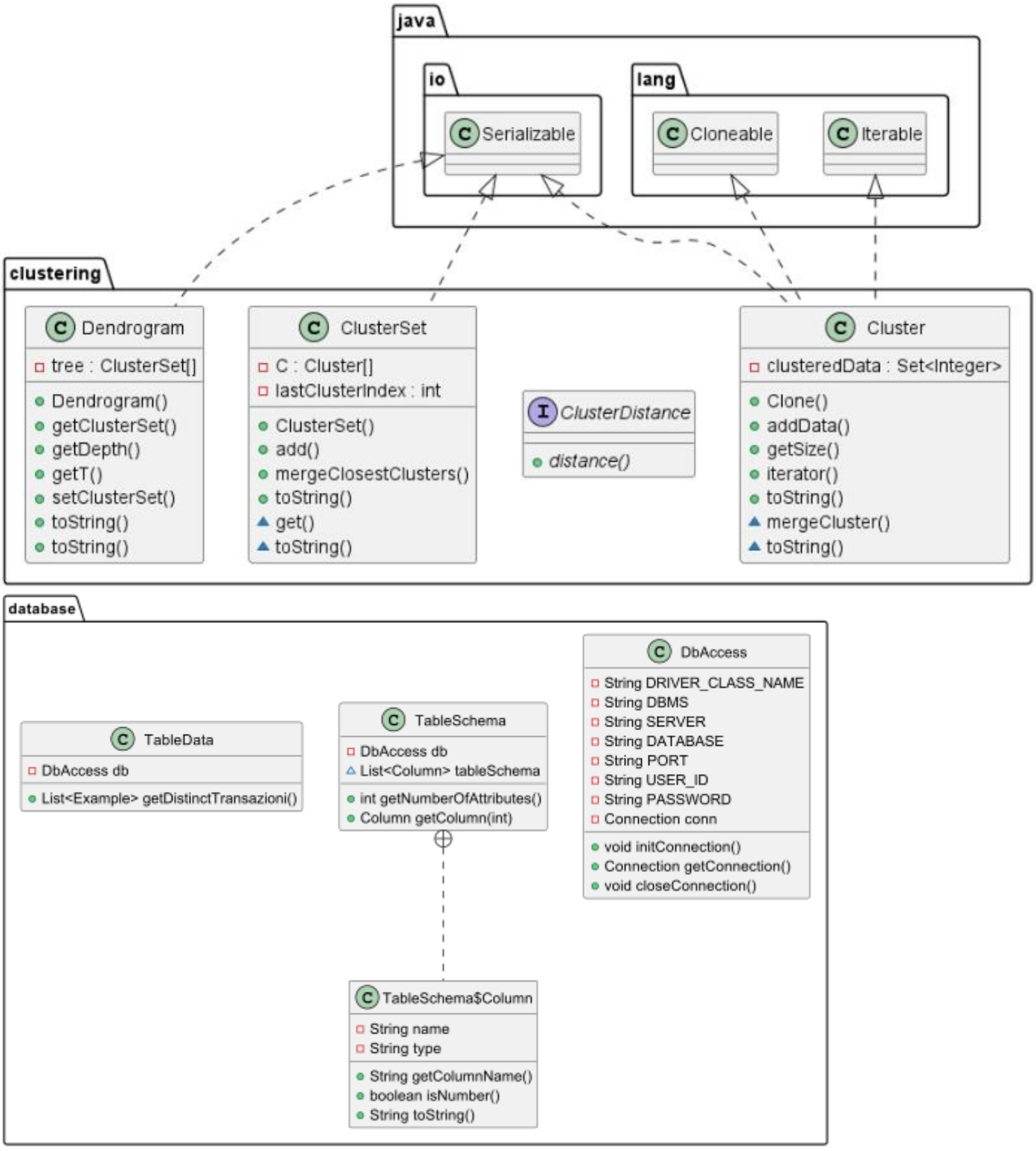


3.2 Server UML

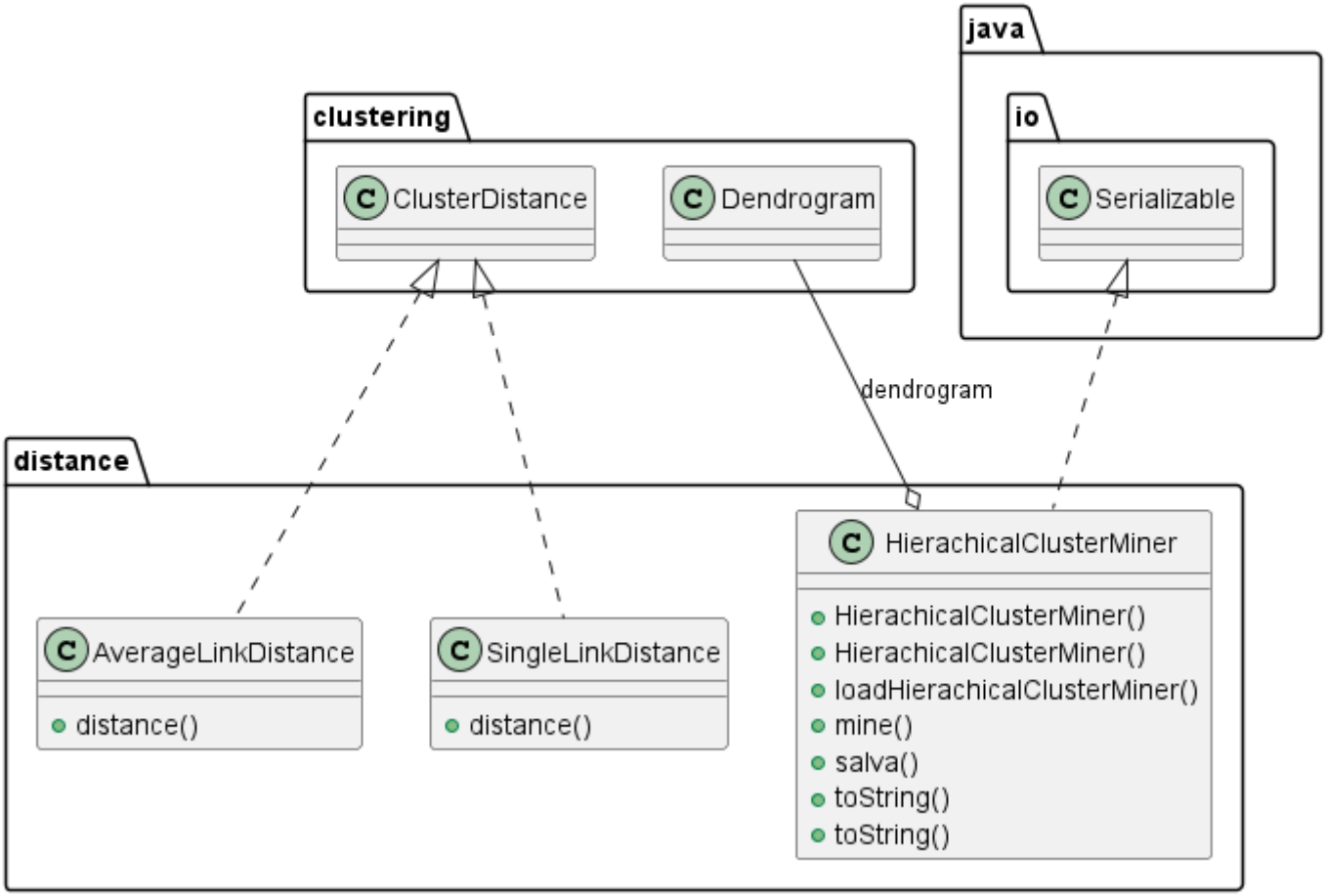



HCLUSSERVER's Class Diagram
java
io
Serializable
lang
Iterable
Cloneable
exceptions
DatabaseConnectionException
EmptySetException
InvalidDepthException
InvalidSizeException
MissingNumberException
NoDataException
data
Example
Data
tableSchema
ase
database
DbAccess
TableData
TableSchema
clustering
ClusterSet
Dendrogram
Cluster
ClusterDistance
dendrogram
server
ServerOneClient
ServerException
MultiServer
dendrogram
distance
AverageLinkDistance
SingleLinkDistance
HierachicalClusterMiner
database
TableData
TableSchema
db
db
DbAccess

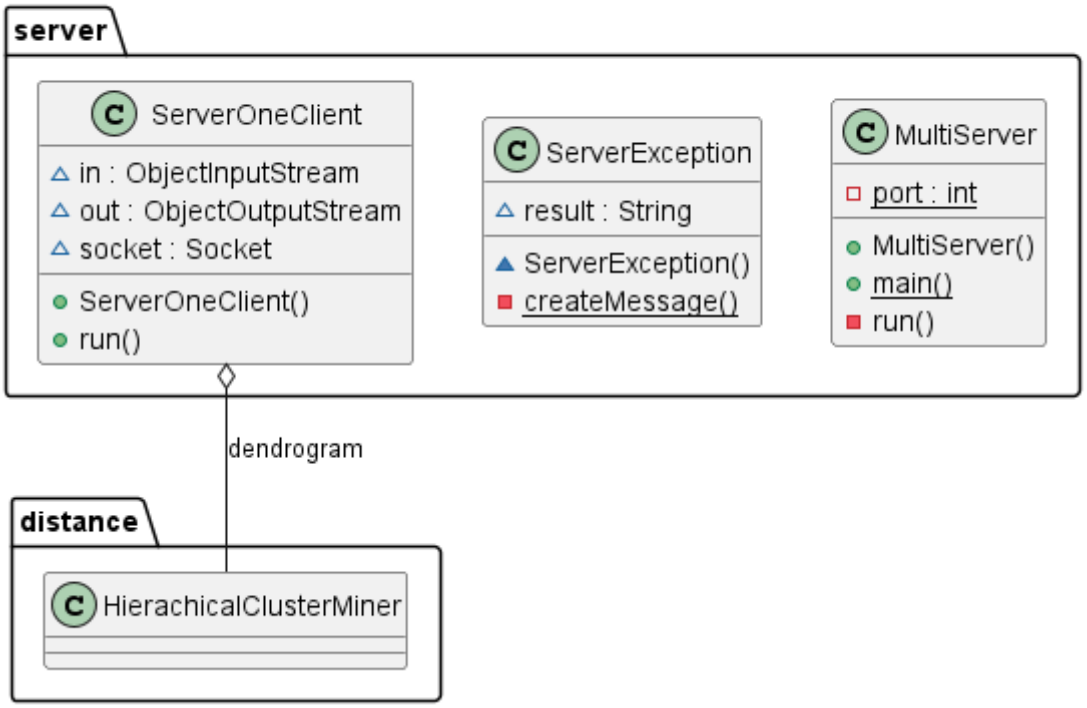
CLUSTERING's Class Diagram

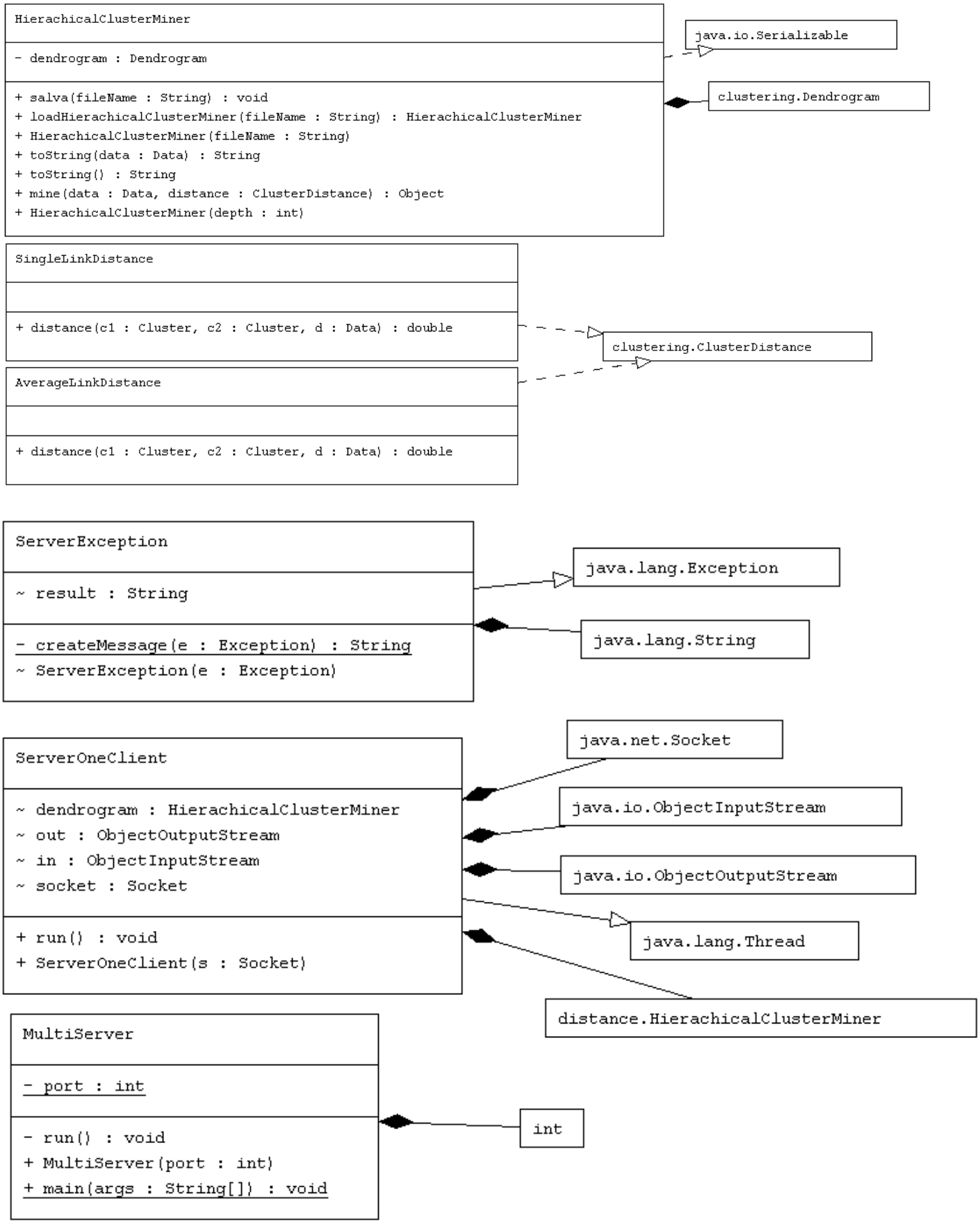


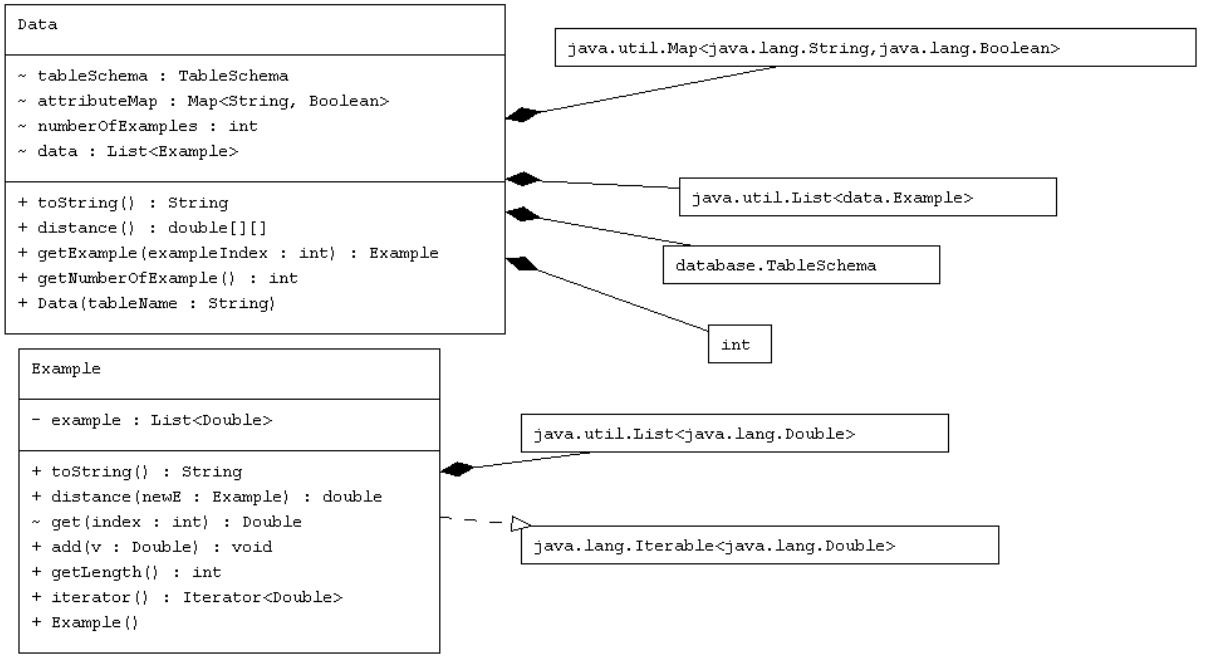
DISTANCE's Class Diagram

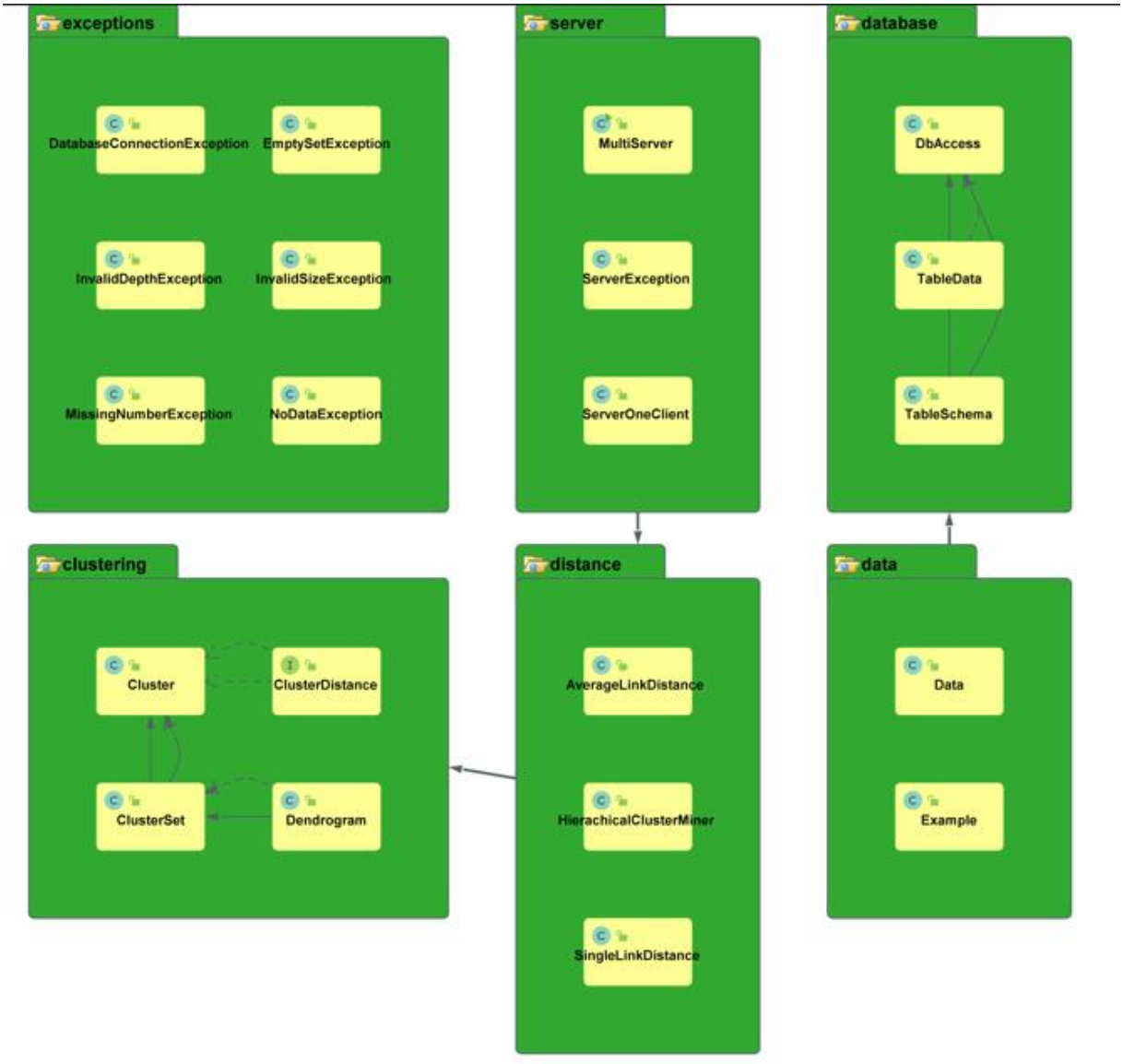


SERVER's Class Diagram




HierachicalClusterMiner
- dendrogram : Dendrogram
+ salva(fileName : String) : void
+ loadHierachicalClusterMiner(fileName : String) : HierachicalClusterMiner
+ HierachicalClusterMiner(fileName : String)
+ toString(data : Data) : String
+ toString() : String
+ mine(data : Data, distance : ClusterDistance) : Object
+ HierachicalClusterMiner(depth : int)
java.io.Serializable
clustering.Dendrogram
SingleLinkDistance
+ distance(c1 : Cluster, c2 : Cluster, d : Data) : double
AverageLinkDistance
+ distance(c1 : Cluster, c2 : Cluster, d : Data) : double
clustering.ClusterDistance
ServerException
~ result : String
- createMessage(e : Exception) : String
~ ServerException(e : Exception)
java.lang.Exception
java.lang.String
ServerOneClient
~ dendrogram : HierachicalClusterMiner
~ out : ObjectOutputStream
~ in : ObjectInputStream
~ socket : Socket
+ run() : void
+ ServerOneClient(s : Socket)
java.net.Socket
java.io.ObjectInputStream
java.io.ObjectOutputStream
java.lang.Thread
distance.HierachicalClusterMiner
MultiServer
- port : int
- run() : void
+ MultiServer(port : int)
+ main(args : String[]) : void
int


Data
~ tableSchema : TableSchema
~ attributeMap : Map<String, Boolean>
~ numberOfExamples : int
~ data : List<Example>
+ toString() : String
+ distance() : double[][]
+ getExample(exampleIndex : int) : Example
+ getNumberOfExample() : int
+ Data(tableName : String)
java.util.Map<java.lang.String,java.lang.Boolean>
java.util.List<data.Example>
database.TableSchema
int
Example
- example : List<Double>
+ toString() : String
+ distance(newE : Example) : double
~ get(index : int) : Double
+ add(v : Double) : void
+ getLength() : int
+ iterator() : Iterator<Double>
+ Example()
java.util.List<java.lang.Double>
java.lang.Iterable<java.lang.Double>


exceptions
DatabaseConnectionException
EmptySetException
InvalidDepthException
InvalidSizeException
MissingNumberException
NoDataException
server
MultiServer
ServerException
ServerOneClient
database
DbAccess
TableData
TableSchema
clustering
Cluster
ClusterDistance
ClusterSet
Dendrogram
distance
AverageLinkDistance
HierachicalClusterMiner
SingleLinkDistance
data
Data
Example

4. GUIDA ALL' INSTALLAZIONE

4.1 Installazione Server

Per il corretto funzionamento del progetto lato server è necessario:

- Spostare l'intera cartella del progetto sul desktop;
- Installare MySQL 8.0;
- Installare Java Runtime Environment (JRE) versione 20;
- Avviare il server MySQL;
- Eseguire lo script MySQL presente nella cartella "SQL Connector". Tale script inizializza il database con tabelle di esempio.[i]

Per avviare il server è possibile aprire il file "*Eseguibile Server.bat*" contenuto nella cartella *"Eseguibile/Base"*. Alternativamente, è possibile avviare il server tramite riga di comando indicando (parendo dalla cartella in cui si trova il file *Eseguibile/Base/HclusServer.jar*):

- La directory in cui è contenuto il java.exe (se non è contenuto nel PATH) - Il comando -jar che indica di avviare un file .jar

La riga sarà simile a:

C:\$PathTo$\java.exe -jar HclusServer.jar

4.1 Installazione Client

Per il corretto funzionamento del progetto lato client è necessario:

- Installare Java Runtime Environment (JRE) versione 20;
- Avviare il server[ii]

Per avviare il client è possibile aprire il file *Eseguibile Client.bat* contenuto nella cartella *"Eseguibile/Base"*. Alternativamente, è possibile avviare il client tramite riga di comando indicando (partendo dalla cartella in cui si trova il file *Eseguibile/Base/HclusClient.jar*):

- La directory in cui è contenuto il java.exe (se non è contenuto nel PATH)
- Il comando -jar che indica di avviare un file .jar
- L'indirizzo IP a cui è collegato il server (di default 127.0.0.1) - La porta su cui è in ascolto il server (di default 8080)

La riga sarà simile a:

C:\$pathTo$\java.exe -jar HclusClient.jar 127.0.0.1 8080

5. GUIDA UTENTE

Nella cartella principale del progetto è presente una sottocartella *"File memorizzati"*, nella quale verranno salvati (e caricati) in file. In essa sono presenti già dei file a scopo di esempio
La tabella di esempio presenti nello script MySQL si chiamana *"exampletab"*.

5.1 Guida base alla interazione da console

Nella cartella *"Eseguibile"* eseguire il file *"Eseguibile generale Base.bat"*. Si apriranno due distinte schermate a linea di comando: una per il server e una per il client

1) Avvio server:

```
Started: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=8080]
```

2) Avvio Client:

```
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=51152]
Nome tabella:
```

3) Carica Dendrogramma da File

```
Nome tabella:
exampletab
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:1
Inserire il nome dell'archivio (comprensivo di estensione):
example3.dat
level0:
cluster0:<1.0 2.0 0.0>
cluster1:<0.0 1.0 -1.0>
cluster2:<1.0 3.0 5.0>
cluster3:<1.0 3.0 4.0>
cluster4:<2.0 2.0 0.0>

level1:
cluster0:<1.0 2.0 0.0><2.0 2.0 0.0>
cluster1:<0.0 1.0 -1.0>
cluster2:<1.0 3.0 5.0>
cluster3:<1.0 3.0 4.0>

level2:
cluster0:<1.0 2.0 0.0><2.0 2.0 0.0>
cluster1:<0.0 1.0 -1.0>
cluster2:<1.0 3.0 5.0><1.0 3.0 4.0>
```

4) Apprendi Dendrogramma da database

```
Nome tabella:
exampletab
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:2
Introdurre la profondita' del dendrogramma
3
```

```
Distanza: single-link (1), average-link (2):
1
level0:
cluster0:<1.0 2.0 0.0>
cluster1:<0.0 1.0 -1.0>
cluster2:<1.0 3.0 5.0>
cluster3:<1.0 3.0 4.0>
cluster4:<2.0 2.0 0.0>

level1:
cluster0:<1.0 2.0 0.0><2.0 2.0 0.0>
cluster1:<0.0 1.0 -1.0>
cluster2:<1.0 3.0 5.0>
cluster3:<1.0 3.0 4.0>

level2:
cluster0:<1.0 2.0 0.0><2.0 2.0 0.0>
cluster1:<0.0 1.0 -1.0>
cluster2:<1.0 3.0 5.0><1.0 3.0 4.0>


Inserire il nome dell'archivio (comprensivo di estensione):
example3.dat
```

5) Casi particolari:

```
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=52223]
Nome tabella:
example
[404] La tabella example non esiste
Nome tabella:
```

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:A
Errore: input non valido, per favore inserisci un numero intero.
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:
```

Il main fornito all'interno della quinta esercitazione terminava il software una volta finito di eseguire il caricamento del dendrogramma da file o con l'apprendimento dal database, per rimanere fedele alla versione del main fornita non sono stati applicati cambiamenti, ma sono stati riscritti i metodi main() e menu() sotto commento nel caso si volesse avere una versione del main ciclica che non termini alla fine di ogni operazione di caricamento o apprendimento, ma soltanto sotto richiesta dell'utente.

NOTE

[i] In alternativa si può aprire con un editor di testo e copiare il contenuto nella shell MySQL [ii] Per passare dalla versione base a quella estesa o viceversa, assicurarsi di utilizzare la giusta versione del server (\Base\). Se necessario, chiudere il server esteso prima di aprire il server base