

Manuale utente - QTClustering - Base

Credits software e documentazione: **Marco Pio Cascella**

Indice

- 1. Introduzione
- 2. Guida di installazione
- 3. Esecuzione del sistema
- 4. Manuale d'uso
- 5. Architettura di sistema UML

1. Introduzione

Il progetto QTClustering si svolge durante l'anno accademico 2024-2025 nel corso di "Metodi avanzati di programmazione". L'obiettivo principale di questo progetto è quello di creare un sistema client-server capace di implementare il clustering quality threshold. ([Per maggiori informazioni riguardo al clustering e in particolare a questo tipo di clustering](#))

Come anticipato precedentemente, il nostro sistema si basa su due principali componenti:

- **Server:** Componente formata dalle principali classi che regolano e governano l'algoritmo di clustering.
- **Client:** Componente che permette all'utente di poter interagire con il lato server dell'applicazione e di poter utilizzare i servizi di clustering (e anche altri servizi).

2. Guida di installazione

Per installare il software QTClustering, è necessario seguire i seguenti passaggi:

1. Scaricare e installare il Java Development Kit (JDK):

- Scaricare il JDK versione 24.0.2 dal sito ufficiale di [Oracle](#)
- Dopo averlo scaricato, procediamo con l'installazione tramite terminale di Windows:
 - Aprire il **Prompt dei comandi come amministratore** e digitare il seguente comando:

```
curl -o jdk-24_windows-x64_bin.exe  
https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.exe
```

- Una volta completato il download, installare l'eseguibile del JDK eseguendo il comando:

```
jdk-24_windows-x64_bin.exe
```

- Questo avvierà il processo di installazione del JDK.

Al termine dell'installazione, verificare che sia andata a buon fine digitando il seguente comando:

```
java -version
```

- L'output dovrebbe restituire la versione di Java appena installata (24).
Un esempio di output atteso è:

```
java version "24.0.2" 2025-07-15
Java(TM) SE Runtime Environment (build 24.0.2+7-70)
Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+7-70, mixed mode,
sharing)
```

- Per aggiungere il JDK al **PATH** delle variabili di sistema, digitare il comando:

```
setx PATH "%PATH%;C:\Program Files\Java\jdk-24\bin"
```

- Se si desidera verificare l'aggiunta del percorso al PATH, utilizzare il comando:

```
echo %PATH%
```

2. Scaricare ed installare MySQL

Un'altra componente fondamentale per il corretto funzionamento del software è il download e la configurazione di **MySQL Community Server** scaricabile al seguente [link](#) (Si consiglia di scaricare la versione 8.0.40, l'utilizzo di versioni inferiori o superiori non garantisce la compatibilità con il software), l'installazione avverrà successivamente mediante strumento Wizard.

Scrivendo il seguente comando all'interno del terminale di Windows è possibile verificare se l'installazione è avvenuta correttamente:

```
mysql --version
```

Un esempio di output è il seguente:

```
mysql Ver 8.0.40 for Win64 on x86_64 (MySQL Community Server - GPL)
```

3. Esecuzione del sistema

All'interno della directory del progetto, più precisamente all'interno della cartella **"File per l'avvio"** sono presenti quattro file batch (Bisogna eseguirli in ordine per poter utilizzare correttamente il software):

- **inizializzazione_db.bat**: Facendo doppio click sul file bat, la prima cosa che verrà chiesta all'utente è quella di inserire la password dell'user root, configurata al momento dell'installazione di MySQL.

La funzionalità principale di questo file batch è quella di automatizzare la creazione del database **MapDB** e dell'utente **MapUser**, identificato dalla password **"map"**, (UTILE PER IL SECONDO FILE BATCH) richiamando uno script sql (**"inizializzazione_db.sql"** presente all'interno della cartella **"File per l'avvio"**), di seguito il contenuto del file batch:

```
@echo off

"C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql" -u root -p <
"inizializzazione_db.sql"

IF ERRORLEVEL 1 (
    echo Errore durante l'inizializzazione del database. Controllare il file di
log di MySQL per maggiori informazioni.
) ELSE (
    echo Inizializzazione del Database MapDB e dell'utente MapUSER riuscita.
)

pause
```

- **tabella_campione.bat**: Il seguente file batch automatizza la creazione di una tabella di esempio per poter testare il software, facendo doppio click sul file bat, la prima cosa che verrà richiesta all'utente è quella di inserire la password, che in questo caso sarà **"map"** come specificato anche nel punto precedente. Una volta inserita la password, a video verrà mostrato un messaggio che comunica all'utente se la creazione della tabella di esempio è avvenuta con successo.

Di seguito è riportato il contenuto del file batch:

```
@echo off
setlocal

REM Percorso dello script
set "SCRIPT_DIR=%~dp0"

REM Percorso completo di mysql.exe
set "MYSQL_PATH=C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe"

REM Messaggio iniziale
echo Avvio esecuzione SQL per la tabella di esempio...

REM Esegui il file SQL usando ridirezione e password interattiva
"%MYSQL_PATH%" -u MapUser -p < "%SCRIPT_DIR%tabella_campione.sql"

REM Controlla l'errore
```

```
if %ERRORLEVEL% neq 0 (
    echo.
    echo ERRORE: Impossibile eseguire lo script SQL.
    echo Verifica la password e che l'utente MapUser abbia accesso al database
MapDB.
    pause
    exit /b 1
) else (
    echo.
    echo SUCCESSO: Lo script SQL e' stato eseguito correttamente!
)

pause
```

(I CONTENUTI DEI FILE .SQL SONO VISUALIZZABILI E SI TROVANO ALL'INTERNO DELLA CARTELLA "**File per l'avvio**")

- **Server_setup_base.bat**: Questo file batch automatizza l'avvio del server, questo il suo contenuto:

```
@echo off
echo Avvio del Server QT...
java -jar "..\Jar\server_start_base.jar"
pause
```

il file .bat dunque esegue il file .jar "**server_start_base.jar**" presente nella cartella "**Jar**" della cartella di progetto.

Facendo doppio click sul file, il terminale verrà aperto mostrando la seguente scritta in caso di avvio corretto e senza errori del server:

```
Avvio del Server QT...
Started ServerSocket[addr=0.0.0.0/0.0.0.0,localport=8080]
```

- **"Client_setup_base.bat"**: Questo file batch automatizza l'avvio del client, questo il suo contenuto:

```
@echo off
java -jar "..\Jar\app_start_base.jar" localhost 8080
pause
```

il file .bat dunque esegue il file .jar "**app_start_base.jar**" presente nella cartella "**Jar**" della cartella di progetto, prendendo come argomenti anche "**localhost 8080**" per stabilire la corretta connessione al server.

Facendo doppio click sul file, il terminale verrà aperto mostrando la seguente scritta in caso di avvio corretto e di successo nella connessione al server:

```

addr = localhost/127.0.0.1
Socket[addr=localhost/127.0.0.1,port=8080,localport=51557]
(1) Load clusters from file
(2) Load data from db
(1/2):

```

4. Manuale d'uso

Una volta avviato correttamente l'applicazione, l'utente avrà di fronte a sé la seguente schermata:

```

addr = localhost/127.0.0.1
Socket[addr=localhost/127.0.0.1,port=8080,localport=51557]
(1) Load clusters from file
(2) Load data from db
(1/2):|

```

- Scelta **(1) Load clusters from file**: Scegliendo l'opzione (1), l'utente potrà visualizzare il contenuto di un file precedentemente salvato con nome della tabella del database + raggio (vedremo successivamente grazie all'opzione (2)), mostrando a schermo però solo i centroidi:

```

addr = localhost/127.0.0.1
Socket[addr=localhost/127.0.0.1,port=8080,localport=51618]
(1) Load clusters from file
(2) Load data from db
(1/2):1
Table Name:playTennis
Radius:1
Centroid=(overcast, 30.0, high, weak, yes), Centroid=(rain, 0.0, normal, weak, yes), Centroid=(sunny, 30.3, high, weak, no)
would you choose a new operation from menu?(y/n)

```

L'applicazione successivamente chiederà all'utente **"would you choose a new operation from menu? (y/n)"**, permettendo in base alla risposta di poter scegliere nuovamente se utilizzare l'opzione (1) o (2), oppure se chiudere l'applicazione.

- Scelta **(2) Load data from db**: Scegliendo l'opzione (2), l'utente potrà inserire il nome della tabella presa in considerazione e il raggio di clustering, visualizzando successivamente il risultato del clustering, successivamente i rispettivi centroidi verranno salvati in un file .dat chiamato nome tabella + raggio + .dat (Per esempio playTennis1.dat):

```
addr = localhost/127.0.0.1
Socket[addr=localhost/127.0.0.1,port=8080,localport=51612]
(1) Load clusters from file
(2) Load data from db
(1/2):2
Table name:playTennis
Radius:2
Number of Clusters:3
0:Centroid=(sunny 30.3 high weak no )
Examples:
[sunny 30.3 high weak no ] dist=0.0
[sunny 30.3 high strong no ] dist=1.0
[sunny 13.0 high weak no ] dist=0.5709571100859968

AvgDistance=0.5236523700286656
1:Centroid=(overcast 12.5 high strong yes )
Examples:
[overcast 30.0 high weak yes ] dist=1.5775577703182049
[overcast 0.1 normal strong yes ] dist=1.4092409343968995
[sunny 12.5 normal strong yes ] dist=2.0
[overcast 12.5 high strong yes ] dist=0.0
[rain 12.5 high strong no ] dist=2.0

AvgDistance=1.397359740943021
2:Centroid=(rain 0.0 normal weak yes )
Examples:
[rain 13.0 high weak yes ] dist=1.4290429150935235
[rain 0.0 normal weak yes ] dist=0.0
[rain 0.0 normal strong no ] dist=2.0
[sunny 0.1 normal weak yes ] dist=1.003300330116104
[rain 12.0 normal weak yes ] dist=0.3960396139324834
[overcast 29.21 normal weak yes ] dist=1.9640264269139867

AvgDistance=1.132068214342683

Would you repeat?(y/n)|
```

L'applicazione successivamente chiederà all'utente **"Would you repeat?(y/n)"** dove in caso di risposta affermativa, chiederà un nuovo raggio (il clustering sarà eseguito sempre sulla stessa tabella), in caso di risposta negativa invece, l'applicazione chiederà **"would you choose a new operation from menu? (y/n)"** che permetterà all'utente di scegliere se utilizzare un'altra funzionalità dell'applicazione, oppure di chiuderla.

Input non validi:

- Scelta (1): In caso di scelta (1), inserendo il nome di un file non esistente, verrà mostrata a schermo una frase di errore che sottolinea all'utente la non esistenza del file:

```
(1) Load clusters from file
(2) Load data from db
(1/2):1
Table Name:playTennis
Radius:5
File non trovato: playTennis5.dat (Impossibile trovare il file specificato)
would you choose a new operation from menu?(y/n)|
```

- Scelta (2): In caso di scelta (2), inserendo il nome di una tabella inesistente o errata, verrà mostrata a schermo la seguente frase di errore:

```
(1) Load clusters from file
(2) Load data from db
(1/2):2
Table name:playFootball
Errore nel caricamento del database: Errore SQL nel recupero della tabella: null
Table name:|
```

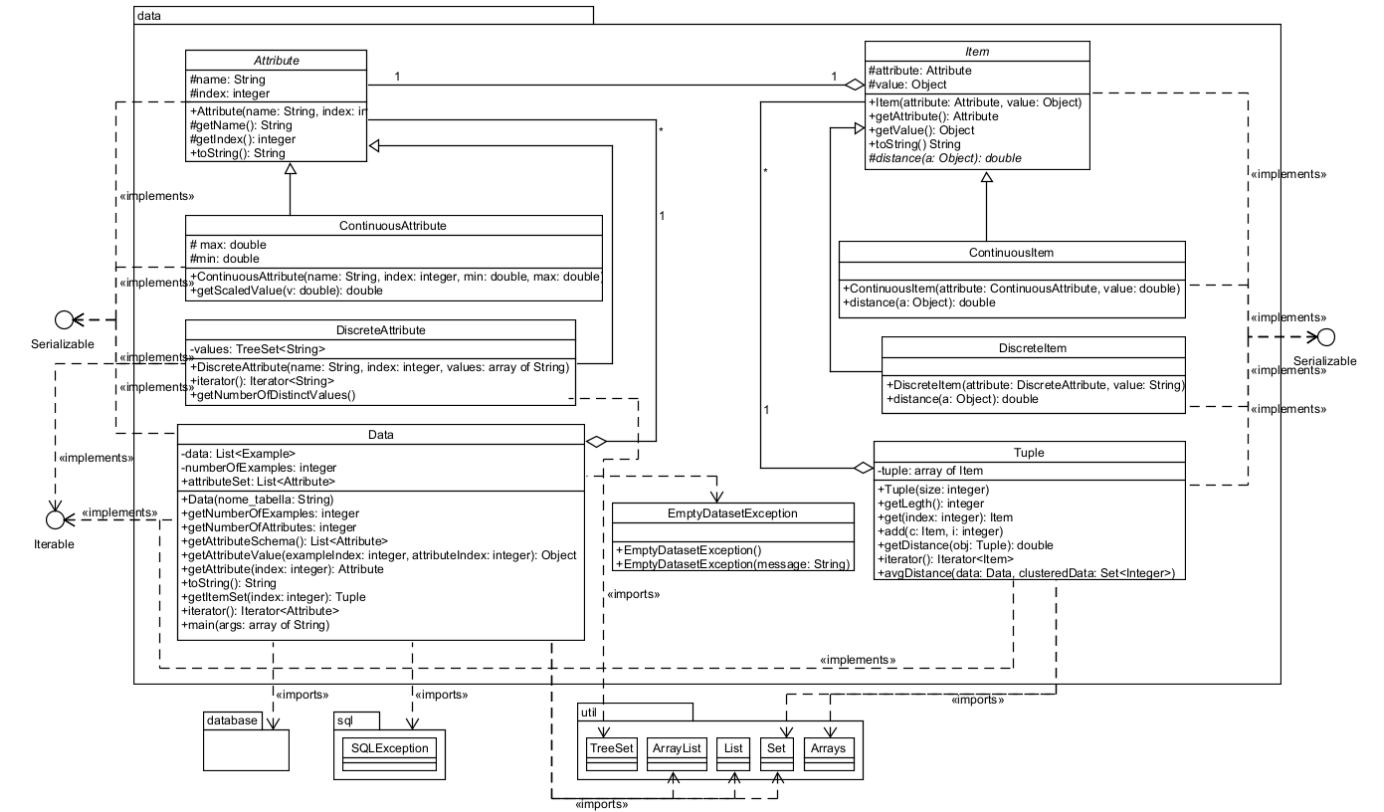
inserendo invece un raggio non valido:

```
Table name:playTennis
Radius:6
Raggio di clustering non valido: null
Would you repeat?(y/n)|
```

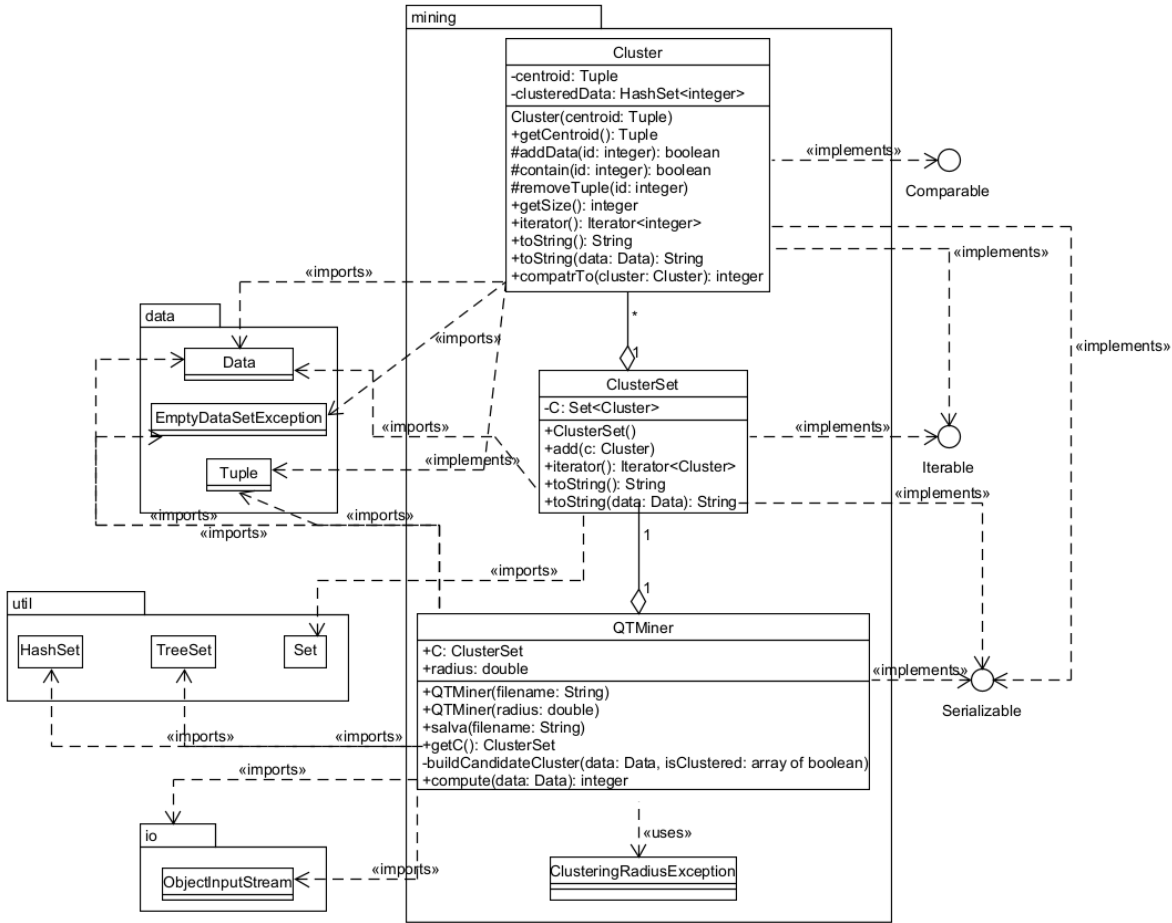
5. Architettura di sistema UML

In questa sezione descriviamo l'architettura di sistema mediante diagrammi UML, utili per descrivere modelli software con un approccio Object Oriented, per maggiori informazioni riguardo eventuali omissioni di package e info utili alla consultazione dei diagrammi è possibile leggere il file "**UML_info**" presente nella cartella UML della directory del progetto.

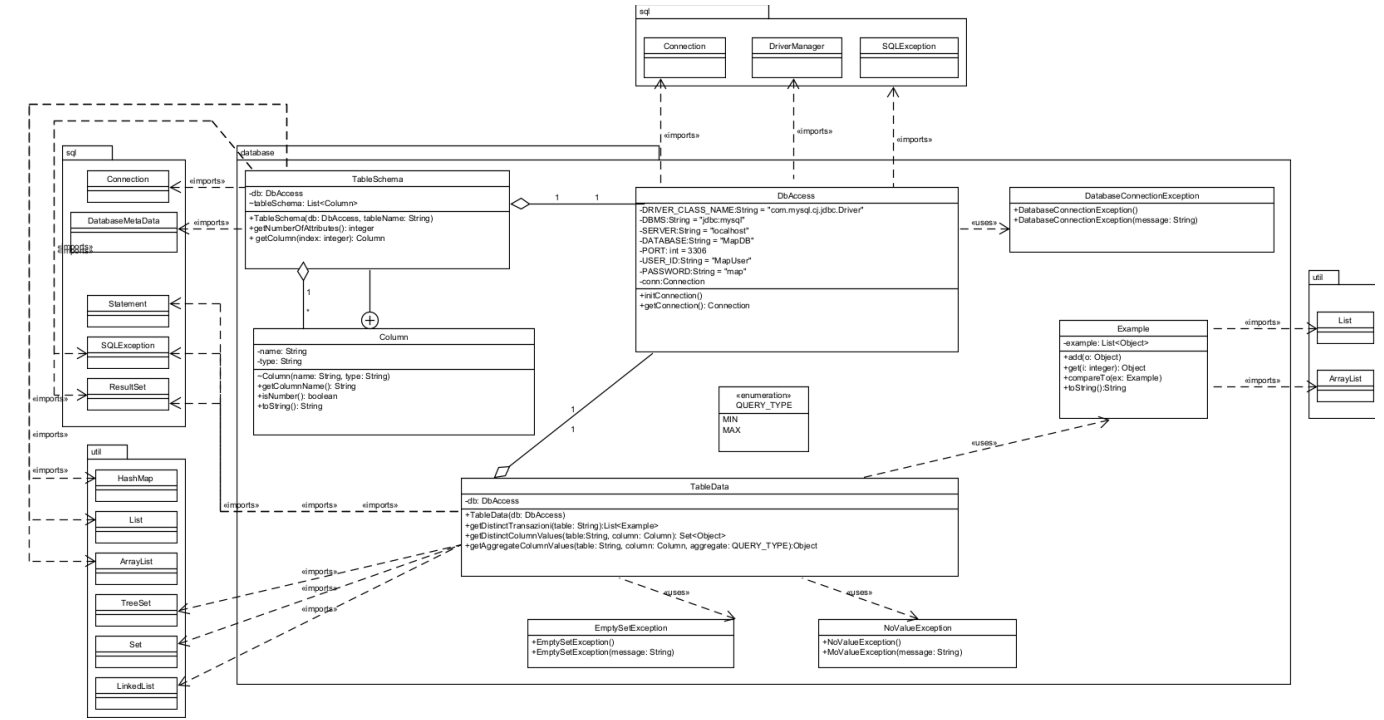
- **Package data**



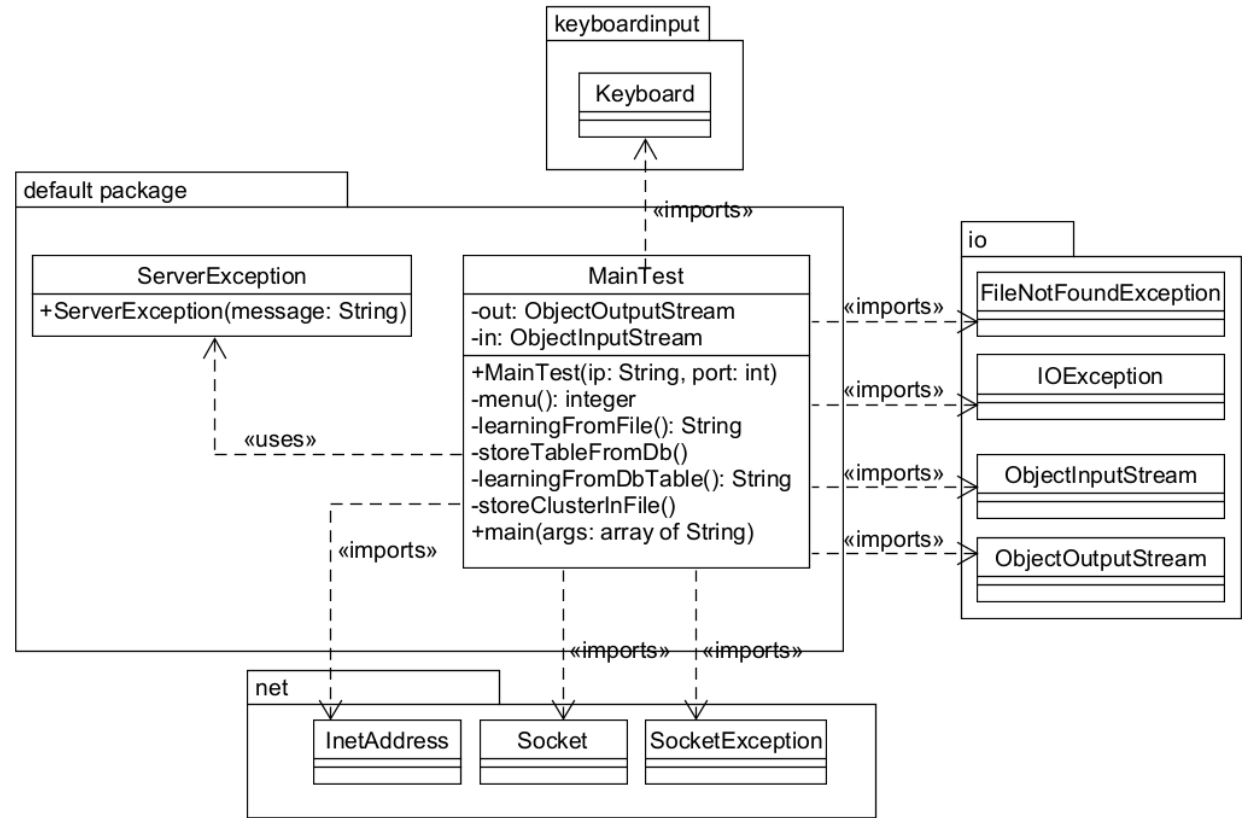
• Package mining



• Package database



• Package default_package



• Package server

