Esercitazione 4 – Serializzazione, DBAccess

Rendere gli oggetti istanza della classe HierachicalClusterMiner serializzabili e modificare la classe `HierachicalClusterMiner` *aggiungendo i metodi:*

```
public static HierachicalClusterMiner loaHierachicalClusterMiner(String fileName)
throws FileNotFoundException,IOException,ClassNotFoundException{…}

public void salva(String fileName)throws FileNotFoundException, IOException {…}
```

per il salvataggio e caricamento di una istanza di `HierachicalClusterMiner` *serializzata*

Modificare il metodo main della classe MainTest per permettere all'utente di caricare un oggetto HierachicalClusterMiner precedentemente serializzato oppure scoprirne uno nuovo (che deve essere serializzato in un archivio con percorso scelto dall'utente).

Definire il package database

Aggiungere le classi DBAccess e TableSchema fornite dal docente

Definire la classe TableData come segue:

***Attributi**:*

```
private DbAccess db;
```

***Metodi**:*

```
TableData(DbAccess db)
```

Comportamento: inizializza l'attributo db

```
List<Example> getDistinctTransazioni(String table) throws SQLException,
EmptySetException,MissingNumberExceptio{…}
```

*Comportamento: interroga la tabella con nome table nel database e restituisce la lista di Example memorizzata nella tabella. Solleva e propoga una istanza di: SLQException in caso di errore nella interrogazione, EmptySetException in caso di tabella vuota , MissingNumberExceptio**n** in presenza di attributi non numerici.*

Definire la classe Data rimuovendo il costruttore Data() e aggiungendo il costruttore:

```
Data(String tableName) throws NoDataException{…}
```

Che avvalora l'oggetto istanza di Data leggendo i suoi esempi dalla tabella con nome tableName nel database. Il costruttore può leggere qualunque tabella memorizzata ne database.

Per il test, creare e popolare la tabella già usata fino ad ora nel database mysql. Il programma, tuttavia, deve funzionare con qualunque tabella che contenga attributi numerici

```
CREATE TABLE mapdb.exampleTab(
    X1 float,
    X2 float,
    X3 float
);

insert into mapdb.exampleTab values(1,2,0);
insert into mapdb.exampleTab values(0,1,-1);
insert into mapdb.exampleTab values(1,3,5);
insert into mapdb.exampleTab values(1,3,4);
insert into mapdb.exampleTab values(2,2,0);

commit;
```

Modificare il metodo main della classe MainTest al fine di usare il nuovo costruttore di Data. Tale operazione deve essere ripetuta fino a quando non si riesce ad istanziare un oggetto valido di Data.

N.B. gestire il caso in cui il dendrogramma caricato (come oggetto precedentemente serializzato) ha una profondità superiore al numero di esempi della tabella caricata da database. Questa situazione può causare errore in `String toString(Data data)` *di Dendrogram*