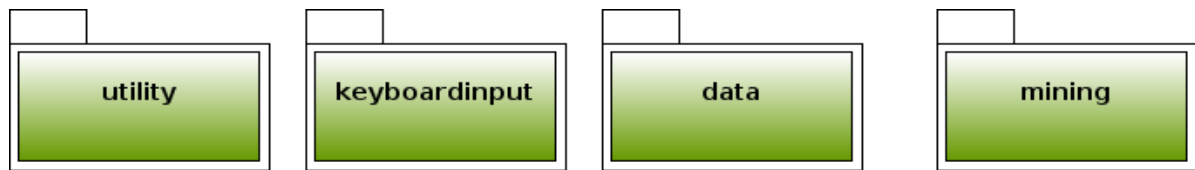
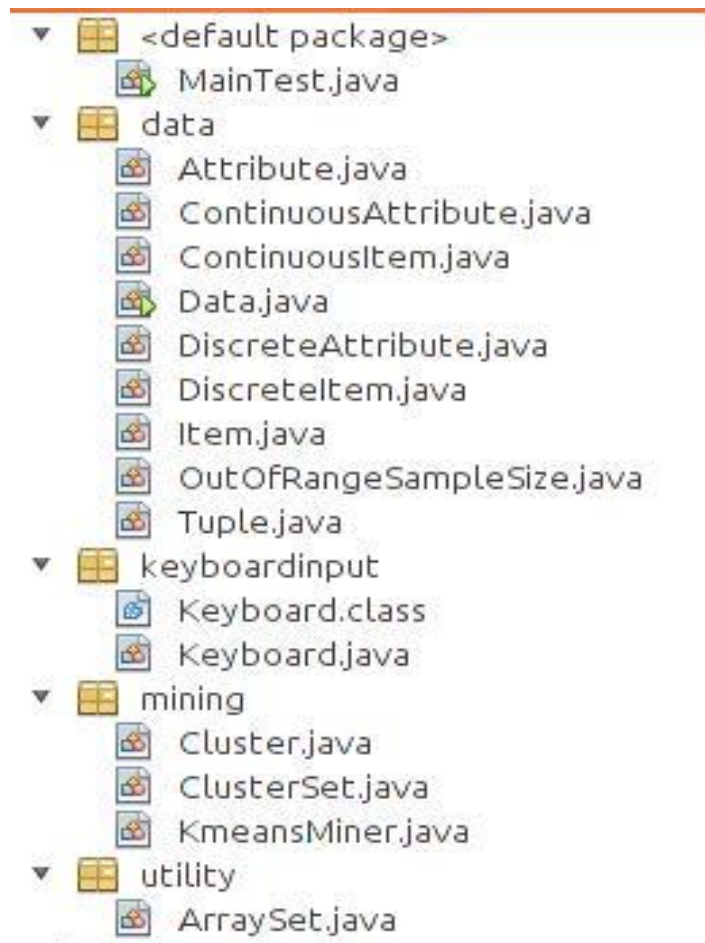


Esercitazione 3 – Package, Input da tastiera ed Eccezioni.



MainTest
Attributes
Operations
public void main(String args[0..*])

Definire i package utility, keyboardinput, data e mining spostare le classi nel package di appartenenza secondo lo schema riportato di segue. Ove necessario, modificare **appropriatamente** anche i qualificatori di visibilità.



■ Aggiungere la classe *Keyboard* (fornita dal docente) che colleziona i metodi di classe per l'acquisizione dell'input da tastiera

■ Modificare *MainTest* in modo da stabilire una interazione con l'utente per acquisire da tastiera il numero intero k di cluster da scoprire e dare la possibilità all'utente di decidere di ripetere l'esecuzione del k -means anche con valori di k differenti.

■ Modificare la classe *Data* con l'aggiunta del metodo:

```
private int countDistinctTuples()
```

Input:

Output: Numero di transazioni distinte memorizzate nella matrice data

Comportamento: Conta il numero di transazioni distinte memorizzate in data (fare uso del metodo boolean `compare(int,int)`)

Esempio:

a,b,a,c

a,b,d,d

a,b,a,c

a,b,d,c

Le transazioni distinte sono tre (evidenziate in giallo)

■ Modificare la classe *Data* con l'aggiunta del membro dato:

```
private int distinctTuples;
```

■ Modificare l'implementazione del costruttore della classe *Data* aggiungendo l'inizializzazione:

```
public Data() {
```

```
...
```

```
distinctTuples=countDistinctTuples();
```

}

■ Definire una classe `OutOfRangeException` per modellare una **eccezione controllata** da considerare qualora il numero k di cluster inserito da tastiera è maggiore rispetto al numero di centroidi generabili dall'insieme di transazioni.

In tale caso l'oggetto eccezione va creato e sollevato nella implementazione del metodo `sampling(...)`.

```
public int[] sampling(int k) throws OutOfRangeSampleSize // da Modificare
```

Comportamento: Se $k \leq 0$ or $k > \text{distinctTuples}$ allora si crea ed espelle un oggetto istanza di `OutOfRangeException`. Altrimenti si procede con implementazione definite in precedenza

L'oggetto eccezione deve essere propagato fino al main (seguendo lo stack delle chiamate). Il gestore di tale eccezione deve essere opportunamente definito nel main della classe `MainTest`.

Esempi di output:

run:

0:sunny,hot,high,weak,no

1:sunny,hot,high,strong,no

2:overcast,hot,high,weak,yes

3:rain,mild,high,weak,yes

4:rain,cool,normal,weak,yes

5:rain,cool,normal,strong,no

6:overcast,cool,normal,strong,yes

7:sunny,mild,high,weak,no

8:sunny,cool,normal,weak,yes

9:rain,mild,normal,weak,yes

10:sunny,mild,normal,strong,yes

11:overcast,mild,high,strong,yes

12:overcast,hot,normal,weak,yes

13:rain,mild,high,strong,no

Inserisci k:3

Numero di Iterazione:3

0:Centroid=(overcast hot high weak yes)

Examples:

[overcast hot high weak yes] dist=0.0

[overcast hot normal weak yes] dist=1.0

AvgDistance=0.5

1:Centroid=(rain cool normal weak yes)

Examples:

[rain mild high weak yes] dist=2.0

[rain cool normal weak yes] dist=0.0

[rain cool normal strong no] dist=2.0

[overcast cool normal strong yes] dist=2.0

[sunny cool normal weak yes] dist=1.0

[rain mild normal weak yes] dist=1.0

AvgDistance=1.3333333333333333

2:Centroid=(sunny mild high strong no)

Examples:

[sunny hot high weak no] dist=2.0

[sunny hot high strong no] dist=1.0

[sunny mild high weak no] dist=1.0

[sunny mild normal strong yes] dist=2.0

[overcast mild high strong yes] dist=2.0

[rain mild high strong no] dist=1.0

AvgDistance=1.5

Vuoi ripetere l'esecuzione?(y/n)