

# GUIDA ALL'USO: RoundRobinDB - libreria

Per utilizzare la libreria del RoundRobinDB all'interno di un vostro personale progetto potete procedere in tre modi:

→ copiate l'archivio *RoundRobinDB.jar* all'interno della vostra cartella di lavoro (ossia la cartella in cui avete salvato i file sorgenti .java che fanno uso della libreria del RoundRobinDB). Ora per compilare ed eseguire il vostro progetto dovrete digitare i seguenti comandi:

```
javac -classpath ./RoundRobinDB.jar; esempio.java  
java -classpath ./RoundRobinDB.jar; esempio
```

sostituendo ovviamente al nome "esempio" il nome del vostro file o del vostro package.

→ copiate l'archivio *RoundRobinDB.jar* nella cartella corrispondente al CLASSPATH di java e procedete alla normale compilazione ed esecuzione del vostro progetto.

→ CONSIGLIATO!!! copiate l'archivio *RoundRobinDB.jar* nella cartella `..\jre\lib\ext` e procedete alla normale compilazione ed esecuzione del vostro progetto (nel mio caso tale cartella si trova al seguente percorso `C:\Programmi\Java\jdk1.6.0_06\jre\lib\ext`).

Per l'utilizzo della libreria vi consiglio vivamente di leggere con attenzione le javadoc della libreria ed eventualmente guardarvi i test da me effettuati.

Ecco un paio di esempi di uso della libreria:

```
RecordAttribute[] RS = new RecordAttribute[1];  
RS[0] = new RecordAttribute("Lettera", 10);  
  
DBStructure DBS = null;  
try{DBS = new DBStructure("lettere", 5, 2, 12, 1, RS);  
}catch(IllegalArgumentException e){System.out.println("ERRORE  
DBS"); return;}  
  
RoundRobinDB RRD = null;  
try{RRD = new RoundRobinDB(DBS);  
}catch(IOException e){System.out.println("ERRORE RRD"); return;}
```

Con queste codice viene creato un *RoundRobinDB* chiamato "lettere" che aspetta l'inserimento di un valore ogni 5 secondi con una tolleranza di 2 secondi.

Il database principale è costituito da 12 slots.

Non vi sono database aggregati.

E' possibile inserire un solo record per volta costituito da un unico attributo chiamato "Lettera" la cui dimensione massima non deve superare i 10 bytes.

```

RecordAttribute[] RS = new RecordAttribute[3];
RS[0] = new RecordAttribute("IP", 50);
RS[1] = new RecordAttribute("DownloadKbps", 30);
RS[2] = new RecordAttribute("UploadKbps", 30);

ExampleAggregationFunction aggregationFunction = new
ExampleAggregationFunction();

DBAggregatedStructure[] DBASSs = new DBAggregatedStructure[2];
DBASSs[0] = new DBAggregatedStructure("IPByDay", "IPbyHour", 12,
3600, aggregationFunction, 24);
DBASSs[1] = new DBAggregatedStructure("IPByWeek", "IPByDay", 24,
86400, aggregationFunction, 7);

DBStructure DBS = null;
try{DBS = new DBStructure("IPbyHour", 300, 120, 12, DBASSs, 2, RS);
}catch(IllegalArgumentException e){System.out.println("ERRORE
DBS"); return;}

RoundRobinDB RRD = null;
try{RRD = new RoundRobinDB(DBS);
}catch(IOException e){System.out.println("ERRORE RRD"); return;}

```

Con queste codice viene creato un *RoundRobinDB* chiamato "IPByHour" che aspetta l'inserimento di un valore ogni 300 secondi (5 minuti) con una tolleranza di 120 secondi (2 minuti).

Il database principale è costituito da 12 slots.

Vi sono 2 database aggregati.

Uno si chiama "IPByDay" e aggrega gli ultimi 12 slots del database "IPByHour" ogni 3600 secondi (1 ora) utilizzando la classe *ExampleAggregationFunction* come funzione di aggregazione.

Questo database è costituito da 24 slots.

L'altro database aggregato si chiama "IPByWeek" e aggrega gli ultimi 24 slots del database "IPByDay" ogni 86400 secondi (24 ore) utilizzando la classe *ExampleAggregationFunction* come funzione di aggregazione.

Questo database è costituito da 7 slots.

E' possibile inserire 2 record per volta ciascuno costituito da 3 attributi chiamati :

- "IP" la cui dimensione massima non deve superare i 50 bytes.
- "DownloadKbps" la cui dimensione massima non deve superare i 30 bytes.
- "UploadKbps" la cui dimensione massima non deve superare i 30 bytes.

**IMPORTANTE!!!** → ricordatevi sempre di lanciare il metodo *close()* quando terminate di utilizzare il *RoundRobinDB* altrimenti il database potrebbe risultare danneggiato, corrotto o inutilizzabile.