

## JDBC

Fallucchi Francesca

[fallucchi@info.uniroma2.it](mailto:fallucchi@info.uniroma2.it)

<http://art.unroma2.it/fallucchi>

s.a. 2010/2011

## JDBC

- API per far interagire applicazioni Java con basi di dati
- API per l'accesso a basi di dati, in modo indipendente dalla specifica tecnologia
- JDBC è una interfaccia, realizzata da classi chiamate driver

l'interfaccia è standard

i driver contengono le specificità dei singoli DBMS (o di altre fonti informative)

## Il funzionamento di JDBC

- Caricamento del driver
- Apertura della connessione alla base di dati
- Richiesta di esecuzione di istruzioni SQL
- Elaborazione dei risultati delle istruzioni SQL

## Connessione Java a un DB MySQL

- L'interfaccia JDBC è contenuta nel package `java.sql`
  - `import java.sql.*;`
- `Class.forName(String driver):`
  - È un metodo statico
  - Permette di caricare la classe (che è la stringa presa in input) e ritornare l'istanza di classe
  - Il driver deve essere caricato
  - `Class.forName("com.mysql.jdbc.Driver ")`

## Connessione Java a un DB MySQL

- **Connessione:** interfaccia del package `java.sql`
  - permette di specificare la connessione con un database specifico come MySQL
  - permette l'esecuzione dei costrutti SQL
  - costituisce un collegamento attivo fra Java e la base di dati
  - oggetto di tipo `Connection`
    - `Connection con = null;`

## Connessione Java a un DB MySQL

- **DriverManager:** classe del package `java.sql`
  - Controlla un set di drivers JDBC
  - Ogni driver può essere registrato in essa
- `getConnection(String url, String userName, String password)`
  - Metodo che stabilisce la connessione a una certa url e ha tre argomenti
    - url dove è memorizzato o creato il database
    - `userName` di MySQL
    - `Password` di MySQL

## Connessione Java a un DB MySQL

```
String url = "jdbc:mysql://localhost:3306/";
String dbName = "esempioJDBC";
String userName = "root";
String password = "root";
con=DriverManager.getConnection(url+dbName,
    userName,password);
```

## Connessione Java a un DB MySQL

- **con.close():**  
Metodo usato per terminare la connessione al database.  
Permette di liberare tutte le risorse occupate dal database
- **printStackTrace():**  
Metodo usato per mostrare I messaggi di errore  
Se la connessione non è stabilita viene sollevata un'eccezione e stampato il messaggio

## Connessione Java a un DB MySQL

```
import java.sql.*;
public class MysqlConnect{
    public static void main(String[] args) {
        System.out.println("MySQL Connect Example.");
        Connection conn = null;
        String url = "jdbc:mysql://localhost:3306/";
        String dbName = "esempioJDBC";
        String driver = "com.mysql.jdbc.Driver";
        String userName = "root";
        String password = "root";
        try {
            Class.forName(driver).newInstance();
            conn =
            DriverManager.getConnection(url+dbName,userName,password);
            System.out.println("Connected to the database");
            conn.close();
            System.out.println("Disconnected from databas");
        } catch (Exception e) { e.printStackTrace();} }}
```

## Statement

- Un'interfaccia i cui oggetti consentono di inviare, tramite una connessione, istruzioni SQL e di ricevere i risultati forniti
- Un oggetto di tipo Statement viene creato con il metodo createStatement di Connection con i seguenti metodi:  
executeUpdate  
executeQuery  
execute  
executeBatch

## ResultSet

- I risultati delle interrogazioni sono forniti in oggetti di tipo ResultSet (interfaccia definita in java.sql)  
sequenza di ennuple su cui si può "navigare"
  - in avanti
  - indietro
  - con accesso direttodalla ennupla "corrente" si possono estrarre i valori degli attributi

## ResultSet

- Metodi principali:  
**next()**  
**getXXX(posizione)**
  - getString(3);
  - getInt(2);**getXXX(nomeAttributo)**
  - getString("Cognome");
  - getInt("Codice");

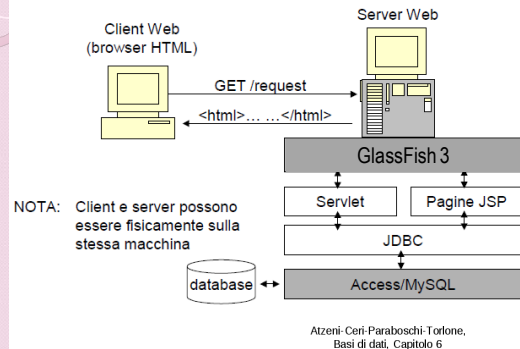
## Esecuzione dell'interrogazione ed elaborazione del risultato

- Esecuzione dell'interrogazione

```
Statement query = con.createStatement();
ResultSet result =
    query.executeQuery("select * from Corsi");
```
- Elaborazione del risultato

```
while (result.next()){
    String nomeCorso =
        result.getString("NomeCorso");
    System.out.println(nomeCorso);
}
```

## Applicazioni Web



## Servlet

- Sono particolari classi Java che vengono eseguite in server web opportunamente predisposti
- Sono viste come risorse web standard infatti hanno una URL che le caratterizza
- Sono scritte in codice Java e per questo si può far uso di tutte le librerie e le utility java come JDBC che permette la connessione a DBMS

## Funzionamento Servlet

- Occorre disporre di un server che fornisce il supporto all'attivazione ed esecuzione delle servlet
- Modalità request/response
  - L'utente fa richiesta della servlet mediante la URL associata
  - Il server attiva ed esegue la servlet e restituisce il risultato come contenuto della risorsa