

Esercitazione 4

JDBC

Obiettivi dell'esercitazione

- Familiarizzare con l'organizzazione dell'ambiente di lavoro per la realizzazione di applicazioni Java
- Utilizzare i costrutti di base della libreria JDBC per
 1. la gestione delle connessioni con il DBMS
 2. l'esecuzione di comandi di interrogazione

Esercitazione 4

Task1: Impostare l'ambiente di lavoro per un'applicazione Java che accede ad un DBMS (MySQL 5.0) mediante JDBC

Scompattare l'archivio [esercitazioneJDBC.zip](#) in c:\

Verranno così create una serie di cartelle che contengono il materiale relativo all'applicazione da realizzare.

La struttura delle directory create è la seguente:

```
esercitazioneJDBC/  
    |-----bin/  
    |-----lib/  
    |-----sql/  
    |-----src/
```

Il codice sorgente Java (da completare) è posto nella cartella src.

Lo script sql per la creazione della base di dati che vogliamo utilizzare per l'esercitazione è posto nella cartella sql.

La cartella lib/ contiene il driver JDBC di MySQL (mysql-connector-java-5.0.4-bin.jar) scaricabile dal sito <http://dev.mysql.com/downloads/connector/j/5.0.html>.

Infine, la cartella bin/ verrà usata per redirezionarvi il codice binario dell'applicazione.

Esercitazione 4

Task2: Connettersi al DBMS MySQL:

1. Aprire il programma QueryBrowser (Start-->MySQL-->QueryBrowser)
2. Nella casella server inserire: **192.168.51.254**
3. Nella casella user inserire: **prog10xx** (sostituire **xx** con il numero assegnato)
4. Nella casella password inserire: **prog10**
5. Premere il tasto "IGNORE" nella schermata successiva

Task3: Aprire e modificare lo script sql denominato "vacanze.sql" posto nella cartella "sql" in modo che crei un nuovo database denominato "VIAGGI"+"proprio numero di matricola".

Per eseguire lo script andare su **MENU-->File-->OpenScript** e successivamente premere su **"Execute"**

Verificare la corretta inizializzazione eseguendo i comandi:

```
SELECT * FROM USR_VIAGGI.VACANZA;
```

```
SELECT * FROM USR_VIAGGI.VILLEGGIANTE;
```

Esercitazione 4

Task4 : definire il connection manager per la base dati appena creata

Aprire il file `src/ConnectionManager.java` con un editor di testo (consigliato Notepad++).

Completare il metodo

```
public static Connection getConnection()
```

che deve restituire oggetti di tipo `Connection` che consentono l'accesso alla base dati creata nel task precedente tramite.

Per **compilare l'applicazione di esempio**, aprire un Prompt dei comandi, posizionarsi nella cartella contenente il progetto (`c:\esercitazioneJDBC`), ed eseguire il comando

```
javac -sourcepath src/ -d bin/ src/*.java
```

Esercitazione 4

In Java, la compilazione viene eseguita con il comando:

```
javac -sourcepath <source-path> -d <output-path> <source-file-path-names>
```

Dove: *<source-path>* è il percorso contenente i sorgenti,
<output-path> è il percorso dove porre i file oggetto e
<source-file-path-names> sono i nomi dei file sorgente da compilare
comprensivi di percorso.

Qualora il codice dovesse dipendere da librerie esterne, queste possono essere aggiunte ridefinendo il percorso di ricerca con l'opzione *-cp <class-path>* o in alternativa impostando opportunamente la variabile di ambiente CLASSPATH....

Esercitazione 4

... Nel nostro esempio, il codice sorgente si trova nella sotto-directory `src/` mentre l'output della compilazione (file `.class`) viene redirezionato nella sotto-directory `bin/`.

Questa scelta è in linea con l'organizzazione dell'ambiente di lavoro impostata all'inizio dell'esercitazione. In generale, organizzare l'ambiente di lavoro secondo lo schema proposto agevola la manutenzione dell'applicazione e la sua analisi da parte di chi non ha partecipato alla realizzazione dell'applicazione. Sebbene il compilatore sia in grado di gestire i time-stamp dei sorgenti e del codice oggetto per minimizzare il numero di oggetti da ricompilare, a volte può essere opportuno forzare la ricompilazione dell'intera applicazione. Per fare ciò è necessario **cancellare completamente** il contenuto della cartella `bin/`.

In ambiente Windows ciò può essere ottenuto con il comando

```
del /S bin
```

eseguito avendo come directory quella contenente il progetto e confermando via via le operazioni di cancellazione delle singole sotto-directory.

Esercitazione 4

Task5: eseguire l'applicazione

Aprire un Prompt dei comandi e posizionatevi nella cartella contenente il progetto.
Eseguire l'applicazione con il comando:

```
java -cp bin/;lib/mysql-connector-java-5.0.4-bin.jar EsempioJDBCA
```

A questo punto chiedete all'applicazione di visualizzare le vacanze disponibili (cliccando e sul bottone corrispondente)

Esercitazione 4

Nota: In Java, l'esecuzione dell'applicazione viene attivata specificando il nome della classe pubblica contenente il metodo con la segnatura:

```
public static void main(String[] args)
```

detta anche *main-class*. Il codice delle classi (e delle risorse) necessarie all'esecuzione dell'applicazione devono essere raggiungibili mediante il *class-path*, che specifica un elenco di directory e di *assembly* (file con estensione .jar) contenenti gli elementi in questione. Il percorso predefinito comprende le librerie standard, la directory corrente e quanto specificato nella variabile di ambiente CLASSPATH.

Nel caso dell'esempio, invece, le classi dell'applicazione sono poste nella cartella bin/, mentre il driver JDBC è contenuto nell'assembly mysql-connector-java-5.0.4-bin.jar posto nella cartella lib/.

Per specificare il nuovo percorso di ricerca si utilizza l'opzione `-cp <class-path>`.

Esercitazione 4

Task6: implementare una funzione mediante JDBC

Implementare il metodo

```
EsempioJDBC.visualizzaPartecipanti(String descVacanza)
```

in modo che venga visualizzato nel campo di testo output l'elenco dei villeggianti che hanno partecipato alla vacanza di cui è fornita la descrizione.

Per i dettagli dell'implementazione ed in particolare per come interagire con il componente che realizza il campo di testo analizzare il metodo `visualizzaVacanze()`. Utilizzare sia il costrutto `Statement` che `PreparedStatement`.