Mysqli

In php per effettuare una connessione a un database mySql server è necessario utilizzare la classe mysqli. Il metodo vero e proprio per il collegamento è mysqli\_connect e necessita come parametri l’indirizzo del server, il nome dell’utente, la sua password e il nome del database.

Per eseguire una query si deve utilizzare il metodo mysqli\_query, con la seguente sintassi:

$result = $mysqli->query(sql);

Per estrarre i dati da una query si adopera un array associativo, numerico o entrambi, creato dal metodo mysqli\_fetch\_array. La sua sintassi è la seguente:

$row=$result->fetch\_array(mysqli\_num/mysqli\_assoc/mysqli\_both)

Per contare I record estratti dalla query è possibile utilizzare la funzione mysqli\_num\_rows, data la seguente sintassi:

$cnt\_rows = $result->num\_rows

Nel caso di un errore, per sapere di che tipo di errore si tratta, si ricorre alla funzione mysqli\_error che restituisce in formato stringa il testo dell’errore generato dalla precedente operazione sql. Sintassi:

$error=$mysqli->error

Per effettuare un login è dunque necessario effettuare prima la connessione al database, eseguire la query per controllare se l’utente è presente e poi contare il numero di record, se è uguale a 1 allora l’utente è autenticato, in qualunque altro caso c’è stato un errore.

Classi per un socket in java

Per la creazione di un socket in java è necessario utilizzare 2 classi:

* Java.net.\*, serve per la realizzazione di applicazioni client/server
* Java.io.\*, serve per gestire i flussi di input

Per la connessione a un socket si utilizza la sintassi:

connessione = new socket(indirizzo\_server, porta)

Per leggere i dati inviati dal server sono necessari tre passaggi, bisogna prima creare un InputStream che è uguale a connessione.getInputStream(), per poi passarlo come parametro a un InputStreamReader() per passarlo nuovamente come parametro a un BufferedReader(). La sintassi completa può quindi essere:

informazione=BufferedReader(InputStramReader(connessione.GetInputStream()));

Per inviare i dati sono necessari solo due passaggi, bisogna prima creare un OutPutStream attraverso OutPutStream = connessione.getOutPutStream() da passare poi come parametro a un PrintWriter. La sintassi sarà quindi:

informazione = PrintWriter(connessione.GetOutPutStream);

Per la creazione di un server socket si utilizza la classe ServeSocket nel seguente modo:

ServerSocket = new ServerSocket(porta);

La porta deve essere maggiore a 1023.

Si utilizza il metodo accept() per bloccare il programma finché viene intercettata una richiesta di connessione. Il metodo accept restituisce un socket che gestirà la connessione fino alla fine delle operazioni. È necessario inserire tutte le istruzioni all’interno di un costrutto try-catch, al fine di gestire tutti i possibili errori che possono saltare fuori nell’esecuzione.

Interfacce

Le interfacce sono le componenti che si frappongono tra la macchina e l’utente. Una buona interfaccia permette all’utente di essere utile, che non deve assolutamente adattarsi a essa. È fondamentale inoltre creare dei sistemi per rimediare a dei possibili errori commessi, chiedere ad esempio conferma per l’esecuzione di azioni pericolose e fornire dei metodi di annullamento. I principali elementi per comporre un’interfaccia sono:

* Form
* Finestre
* Icone
* Menu

Per realizzare delle interfacce è necessario analizzare come l’uomo rappresenta la propria conoscenza e può avvenire in due modi:

* Reti semantiche, forma attraverso la quale l’uomo associa qualcosa a delle parole, ad esempio alla parola “mordere” si può associare un cane.
* Reti basate sulla conoscenza, si acquisisce mediante la ripetizione di azioni svolte, ad esempio si crea il concetto di “mordere” attraverso una serie di esperienze basate su questo tipo di azioni.