

Interfața Java API ODBC – JDBC

```
/* Program de exemplificare a interfetei API ODBC */
import java.net.URL;
import java.sql.*;
// executa un select SQL dintr-o sursa odbc
// url = jdbc:odbc:BazaTest
// utilizator = Anonim
// Parola = parola
class SimpleSelect{

    public static void main (String args[]) {
        // url-ul la care se gaseste sursa odbc pentru baza noastră de
        date
        // a fost creat din odbc manager - system dns
        // s-a ales calea spre baza de date
        // i s-a dat denumirea BazaMea
        // la advanced s-a dat numele utilizatorului și parola de
        conectare
        String url = "jdbc:odbc:BazaMea";
        // select SQL care se va executa
        String query = "SELECT * FROM tabel";
        // utilizatorul de conectare
        String utilizator = "Anonim";
        // parola de conectare
        String parola = "parola";
        try { // dacă este vreo problema la încarcarea driver
            // încarcă driverul de conexiune jdbc-odbc
            Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
            // Încearcă să se conecteze la un driver. Va fi încercat
            // fiecare driver înregistrat până se încarcă unul
            corespunzător
            // care poate procesa URL-ul
            Connection con = DriverManager.getConnection (
                url, utilizator, parola);
            // Dacă nu se poate conecta returnează o excepție. Astfel
            // dacă am ajuns aici suntem conectați cu succes
```

```

// Verifica si afiseaza mesaje de avertisment generate de
conexiune
    checkForWarning (con.getWarnings ());
    // ia obiectul DatabaseMetaData si afiseaza
    // cateva informatii despre conexiune
    DatabaseMetaData dma = con.getMetaData ();
    System.out.println ("\nConectat la " + dma.getURL ());
    System.out.println ("Driver      " +
        dma.getDriverName ());
    System.out.println ("Versiune      " +
        dma.getDriverVersion ());
    System.out.println ("");
    // Creaza un obiect instructiune SQL
    // care pot fi trimise la driver
    Statement stmt = con.createStatement ();
    // Trimit query, genereaza obiect rezultat - ResultSet
    ResultSet rs = stmt.executeQuery (query);
    // Afiseaza toate liniile si coloanele din multimea
resultata
    dispResultSet (rs);
    // distrugе rezultatul
    rs.close ();
    // inchide instructiunea de executat
    stmt.close ();
    // inchide conexiunea
    con.close ();
}
catch (SQLException ex) {
    // Exceptie SQL - afiseaza informatii
    System.out.println ("\n ____ Exceptie SQL interceptata !!
____\n");
    while (ex != null) {
        System.out.println ("Comanda SQL: " +
            ex.getSQLState ());
        System.out.println ("Mesaj: " +
            ex.getMessage ());
        System.out.println ("Producator: " +
            ex.getErrorCode ());
        ex = ex.getNextException ();
}

```

```

        System.out.println ("");
    }
}
catch (java.lang.Exception ex) {
    // altfel de exceptii le sare ...
    ex.printStackTrace ();
}
}

/*Verifica daca sunt avertismente - returneaza true daca gaseste vreunul */
private static boolean checkForWarning (SQLWarning warn)
    throws SQLException
{
    boolean rc = false;
    // daca a intalnit avertisment - afiseaza mesajul
    if (warn != null) {
        System.out.println ("\n__ Avertisment !! __\n");
        rc = true;
        while (warn != null) {
            System.out.println ("Comanda SQL: " +
                warn.getSQLState ());
            System.out.println ("Mesaj: " +
                warn.getMessage ());
            System.out.println ("Producator: " +
                warn.getErrorCode ());
            System.out.println ("");
            warn = warn.getNextWarning ();
        }
    }
    return rc;
}
/*afiseaza multimea rezultata - coloane si linii*/
private static void dispResultSet (ResultSet rs)
    throws SQLException
{
    int i;
    // preia ResultSetMetaData. Acestea vor fi folosite pentru
    // antetele de coloana
    ResultSetMetaData rsmd = rs.getMetaData ();
    // preia numarul de coloane

```

```
int numCols = rsmd.getColumnCount ();
// afiseaza antetele coloanelor
for (i=1; i<=numCols; i++) {
    if (i > 1) System.out.print(",");
    System.out.print(rsmd.getColumnLabel(i));
}
System.out.println("");
// afiseaza toate datele din tabel
boolean more = rs.next ();
while (more) {
    // trece de la un rand la altul afisand datele
    for (i=1; i<=numCols; i++) {
        if (i > 1) System.out.print(",");
        System.out.print(rs.getString(i));
    }
    System.out.println("");
    // incarca urmatoarul rand
    more = rs.next ();
}
}
```