```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.CallableStatement;
import java.sql.Types;
// OJO: no importar com.mysql.jdbc.*

public class principal {
    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();  //carga driver
            conn = DriverManager.getConnection("jdbc:mysql://localhost/" +
                    "classicmodels?" + "user=root&password=amador");
            stmt = conn.createStatement();
            rs = stmt.executeQuery("SELECT * FROM products"); // para select
            if (stmt.execute ("SELECT * FROM products")) {  // para cq sentencia
                rs = stmt.getResultSet();
            }

            String nombreProd = "";
            while (rs.next()) {
                nombreProd +=  (rs.getString("productName") + "\n");
            }
            // System.out.println (nombreProd);


            CallableStatement cStmt = conn.prepareCall("{call mayorDeudor(?, ?)}");
            // si hubiera parámetros de entrada (IN) se les asigna valor así:
            // (el primer argumento tiene índice 1)
            // cStmt.setString(1, "abcdefg"); // existe setInt, setLong, etc.
            // cStmt.setString("inputParameter", "abcdefg"); // alternativa

            // también hay que registrar los parámetros de salida (OUT o INOUT):
            // (el primer argumento tiene índice 1)
            cStmt.registerOutParameter(1, Types.INTEGER); // idCliente
            cStmt.registerOutParameter("maxdebt", Types.FLOAT);
            //(OUT idCli INT(11), OUT maxdebt FLOAT)
            // alternativa: cStmt.registerOutParameter("inOutParam", Types.INTEGER);

            boolean hadResults = cStmt.execute();
            System.out.println ("resultados procedimiento: " + hadResults);
            /* SI EL PROCEDIMIENTO DEVUELVE RESULTSET:
            while (hadResults) {
                rs = cStmt.getResultSet();
                //  PROCESADO DE LOS RESULT SETS
                hadResults = cStmt.getMoreResults();
            }
            */
            int idCliente = cStmt.getInt(1); // alternativa 1: con índices
            float maxDeuda = cStmt.getFloat("maxdebt"); // alternativa 2: nombre
            System.out.println ("MAYOR DEUDOR: idCliente: " + idCliente +
                                " Deuda: " + maxDeuda);
```

```
            // función almacenada: deuda (id INT) RETURNS float
            cStmt = conn.prepareCall("{? = call deuda(?)}");
            cStmt.registerOutParameter(1, Types.FLOAT);
            cStmt.setInt(2, idCliente); // existe setInt, setLong, etc.
            hadResults = cStmt.execute();
            System.out.println ("resultados función: " + hadResults);
            float deuda = cStmt.getFloat(1);
            System.out.println ("deuda del cliente " + idCliente + ": " + deuda);

        } catch (SQLException ex) {
        // handle any errors
              System.out.println("SQLException: " + ex.getMessage());
              System.out.println("SQLState: " + ex.getSQLState());
              System.out.println("VendorError: " + ex.getErrorCode());
        } catch (Exception e) {
            System.out.println("SQLException: " + e.getMessage());
        }
        finally {  // liberación de recursos en orden inversa a us creación
            if (rs != null) {
                try {
                    rs.close();
                } catch (SQLException sqlEx) { } // ignorada
                rs = null;
            }
            if (stmt != null) {
                try {
                    stmt.close();
                } catch (SQLException sqlEx) { } // ignorada
                stmt = null;
            }
            if (conn != null) {
                try {
                    conn.close();
                } catch (SQLException sqlEx) { } // ignorada
                conn = null;
            }
        }
    }
}
```