

Programación Orientada a Objetos

Práctica 7: JDBC



Alumno: Vargas Romero Erick Efraín

Profesor: Tecla Parra Roberto

Fecha: 16-05-2017

Grupo: 2CM4

**Introducción**

JDBC hace que las tareas con bases de datos sean simples. JDBC nos permite realizar operaciones sobre bases de datos, esto es independiente del sistema operativo donde se ejecute la base de datos o bien de la base de datos a la cual se accede, utilizando el dialecto SQL.

**Desarrollo**

Para esta práctica se ha implementado solamente una clase, la cual implementa la interfaz ActionListener

public class practica7 **implements** ActionListener**{**

Lo primero que encontramos son nuestras variables de instancia además hay que hacer hincapié que utilizamos swing para el manejo de la interfaz, siendo así para la interfaz utilizamos objetos JLabel, JTextField JButton y JFrame.

Posteriormente tenemos dos variables de tipo String la primera llamada URL la cual tiene la dirección a la cual se hace la conexión a nuestra base de datos donde tenemos primeramente el controlador, luego la IP o bien escribimos localhost y finalmente el puerto. Nuestra otra variable tipo String es el nombre de la base de datos a la cual nos conectaremos y finalmente un objeto tipo Connection

public JLabel TXTNom**,** TXTRaz**,** TXTEd**,** TXTGen**;**

public JTextField TXTWNom**,** TXTWRaz**,** TXTWEd**,** TXTWGen**;**

public JButton BTNCon**,** BTNIns**;**

public JFrame jframe**;**

public String URL **=** "jdbc:mysql://localhost:3306/"**,**

dbname **=** "dbperros"**;**

public Connection con**;**

Posteriormente tenemos el constructor de nuestra clase, en la cual creamos todos los objetos que son utilizados en la interfaz además de volver a los dos botones que se han creado fuentes de eventos esto utilizando el método addActionListener()

Posteriormente desactivamos el JButton llamado BTNIns con el cual insertamos un registro a la base de datos finalmente añadimos todos los elementos creados a nuestro JFrame y hacemos visible este último.

public practica7**(){**

jframe **=** **new** JFrame**(**"Añadir perros"**);**

jframe**.**setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

jframe**.**setLayout**(new** GridLayout**(**5**,**2**));**

jframe**.**setSize**(**300**,**300**);**

TXTNom **=** **new** JLabel**(**"Nombre"**);**

TXTRaz **=** **new** JLabel**(**"Raza"**);**

TXTEd **=** **new** JLabel**(**"Edad"**);**

TXTGen **=** **new** JLabel**(**"Genero"**);**

TXTWNom **=** **new** JTextField**(**30**);**

TXTWRaz **=** **new** JTextField**(**30**);**

TXTWEd **=** **new** JTextField**(**30**);**

TXTWGen **=** **new** JTextField**(**30**);**

BTNCon **=** **new** JButton**(**"Conectar"**);**

BTNIns **=** **new** JButton**(**"Insertar"**);**

BTNCon**.**addActionListener**(this);**

BTNIns**.**addActionListener**(this);**

BTNIns**.**setEnabled**(false);**

jframe**.**add**(**TXTNom**);**

jframe**.**add**(**TXTWNom**);**

jframe**.**add**(**TXTRaz**);**

jframe**.**add**(**TXTWRaz**);**

jframe**.**add**(**TXTEd**);**

jframe**.**add**(**TXTWEd**);**

jframe**.**add**(**TXTGen**);**

jframe**.**add**(**TXTWGen**);**

jframe**.**add**(**BTNCon**);**

jframe**.**add**(**BTNIns**);**

jframe**.**setVisible**(true);**

**}**

Dentro de nuestra clase practica7 tenemos un método llamado connect que nos retorna un objeto tipo Connection dentro de este método tenemos un bloque try, dentro de el tenemos primeramente que obtener el controlador para JDBC, posteriormente nuestro objeto Connection se le obtiene la dirección de la base de datos a la cual queremos acceder, esto concatenando la cadena URL, la cadena dbname luego añadiendo el nombre del usuario y contraseña para acceder a nuestra base de datos, una vez hecho esto si se ha realizado satisfactoriamente activamos el JButton llamado BTNIns previamente desactivado, en caso de que si haya existido algún error en el bloque catch imprimimos en consola la excepción que se ha atrapado, finalmente retornamos el objeto tipo Connection.

public Connection connect**(){**

Connection connection **=** **null;**

**try{**

Class**.**forName**(**"com.mysql.jdbc.Driver"**);**

connection **=** DriverManager**.**getConnection**(**URL **+** dbname**,** "root"**,** ""**);**

System**.**out**.**println**(**"Connected successfully"**);**

BTNIns**.**setEnabled**(true);**

**}** **catch** **(**Exception e**){**

e**.**printStackTrace**();**

**}**

**return** connection**;**

**}**

Ahora tenemos el método insertNewDog() el cual recibe como parámetro un objeto tipo Connection, en este método obtenemos lo que este dentro de nuestros JTextField, posteriormente verificamos si están o no vacíos y si la edad es mayor a cero si hay algún campo vacío o la edad es menor a cero entonces se envía un mensaje de error, sino entra a un bloque try, en el cual creamos un objeto tipo Statement, creamos un Statement a base de nuestro objeto Connection y ejecutamos una query, la cual nos servirá para insertar datos en la base de datos esto con los datos que se han obtenido de nuestros JTextField, si se ha realizado la inserción satisfactoriamente cerramos el statement, si se ha lanzado una excepción el bloque try mostrará la excepción en consola y enviará un JOptionPane con un mensaje de error

public void insertNewDog**(**Connection conex**){**

String Nombre**,** Raza**,** Genero**;**

int Edad**;**

Nombre **=** TXTWNom**.**getText**().**toString**().**trim**();**

Raza **=** TXTWRaz**.**getText**().**toString**().**trim**();**

Genero **=** TXTWGen**.**getText**().**toString**().**trim**();**

Edad **=** Integer**.**parseInt**(**TXTWEd**.**getText**().**toString**().**trim**());**

**if(!(**Nombre**.**equals**(**""**)** **&&** Raza**.**equals**(**""**)** **&&** Genero**.**equals**(**""**)** **&&** Edad **>** 0**)){**

**try{**

Statement st **=** conex**.**createStatement**();**

st**.**executeUpdate**(**"INSERT INTO tblperros (Nombre, Raza, Edad, Genero) VALUES ('"**+**Nombre**+**"', '"**+**Raza**+**"', '"**+**Edad**+**"', '"**+**Genero**+**"')"**);**

conex**.**close**();**

st**.**close**();**

JOptionPane**.**showMessageDialog**(null,** "Se ha insertado a un nuevo perro"**);**

**}** **catch(**SQLException e**)** **{**

e**.**printStackTrace**();**

JOptionPane**.**showMessageDialog**(null,** "Ha ocurrido un error añadiendo al nuevo perro"**);**

**}**

**}** **else** **{**

JOptionPane**.**showMessageDialog**(null,** "Hay algún campo vacio"**);**

**}**

**}**

Ahora tenemos el método actionPerformed el cual esta contenido en la interfaz ActionListner, en el cual obtendremos la fuente del evento que ha ocurrido y compararemos con los JButton que se han creado posteriormente, si al comparar es el JButton llamado BTNCon se iguala nuestra variable de instancia tipo Connection llamada con a lo que retorne el método connect() y si la fuente es el JButton BTNIns se invoca al método insertNewDog() y se le pasa como parámetro nuestra variable de instancia tipo Connection llamada con.

public void actionPerformed**(**ActionEvent e**){**

JButton button **=** **(**JButton**)**e**.**getSource**();**

**if(**button **==** BTNCon**)**

con **=** connect**();**

**if(**button **==** BTNIns**)**

insertNewDog**(**con**);**

**}**

Finalmente, solo resta ejecutar nuestra aplicación esto en el método main

public static void main**(**String**[]** args**){**

**new** practica7**();**

**}**

**Conclusión:**

Podemos concluir que realmente es sencillo implementar alguna aplicación que haga utilización de bases de datos usando JDBC ya que nos permite realizar conexiones de una manera muy sencilla, además de que la manipulación de los datos en nuestra base de datos es bastante sencilla, solamente debemos conocer las instrucciones necesarias para poder hacer una manipulación de manera satisfactoria.