TEHNOLOGII JAVA
PENTRU DEZVOLTAREA APLICAȚIILOR
LUCRARE DE LABORATOR 21

JDBC – Accesul la baze de date

I. SCOPUL LUCRĂRII

Lucrarea de față are rolul de a prezenta şi familiariza studentul cu modul de accesare si interogare a bazelor de date, lucrând în cadrul aplicațiilor Java.

La sfârşitul acestei lucrări, studentul va avea posibilitatea să scrie programe Java în care să utilizeze noțiunile învățate.

II. NOȚIUNI TEORETICE

Exemple

Se va utiliza aceeaşi bază de date Access – **StudentiDB,** creată în cadrul laboratorului 18. În continuare vom prezenta un program Java care prin intermediul unei interfețe grafice Java Swing permite utilizatorului, într-o manieră foarte simplă, să listeze toate înregistrările din baza de date şi să adauge noi înregistrări.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

class MyQuery extends JFrame implements ActionListener
{
private JSplitPane splitPaneV;
private JPanel panel1;
private JPanel panel2;
private JButton listButton;
private JButton insertButton;
private JButton closeButton;
private Connection con;
private Statement stmt;
static JTextArea textArea;
private String resultset ;

public MyQuery()
{
setTitle( "JDBC-Swing example" );
setSize( 500, 700 );
setBackground( Color.gray );
this.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE) ;
```

```
createPanel1();
createPanel2();

splitPaneV = new JSplitPane( JSplitPane.HORIZONTAL_SPLIT );
getContentPane().add( splitPaneV);
splitPaneV.setLeftComponent( panel1 );
splitPaneV.setRightComponent( panel2 );

resultset=new String();
String url = "jdbc:odbc:StudentiDB";
try {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    }
catch(java.lang.ClassNotFoundException e)
  {
     System.err.print("ClassNotFoundException: ");
     System.err.println(e.getMessage());
  }

try {
     con = DriverManager.getConnection(url,"", "");
     stmt = con.createStatement();
    }

    catch(SQLException ex)
     {
        System.err.println("SQLException:" + ex.getMessage());
     }
}

public void createPanel1()
{
panel1 = new JPanel();
panel1.setLayout( new GridLayout(3,1) );
listButton=new JButton( "List all");
panel1.add( listButton);
insertButton=new JButton( "Insert");
panel1.add( insertButton);
closeButton=new JButton( "Close");
panel1.add( closeButton);
listButton.addActionListener(this);
insertButton.addActionListener(this);
closeButton.addActionListener(this);
}

public void createPanel2()
{
panel2 = new JPanel();
panel2.setLayout( new BorderLayout() );
panel2.add( new JLabel( "Results display:" ), BorderLayout.NORTH );
textArea=new JTextArea();
JScrollPane scrollPane = new JScrollPane();
scrollPane.getViewport().add( textArea );
panel2.add( scrollPane, BorderLayout.CENTER );
```

```
textArea.setEditable(false);
}

public void actionPerformed(ActionEvent e) {

   if( e.getSource() == listButton )
 List();
   else if( e.getSource() == insertButton )
   {
        MyInsert insertFrame=new MyInsert(stmt);
        insertFrame.setVisible(true);

   }
   else if( e.getSource() == closeButton )
   {
        try {
       stmt.close();
              con.close();
     } catch (SQLException sqlx) {
  sqlx.printStackTrace();
     }
 System.exit(0);
   }
}

private void dispResultSet(ResultSet rs) {
 try {
  int i;
                resultset= resultset.substring(0,0);
  String sss = new String();
  ResultSetMetaData rsmd = rs.getMetaData();
  int numCols = rsmd.getColumnCount();
  for (i = 1; i <= numCols; i++) {
   if (i > 1)
    resultset = resultset.concat(",");
   sss = rsmd.getColumnLabel(i);
   resultset = resultset.concat(sss);
  }
         resultset =
               resultset.concat("\n-------------------------------\n");
  resultset = resultset.concat("\n");
  boolean more = rs.next();
  while (more) {
   for (i = 1; i <= numCols; i++) {
    if (i > 1)
     resultset = resultset.concat(",");
    sss = (rs.getString(i));
    if (sss != null)
     resultset = resultset.concat(sss);
   }
   resultset = resultset.concat("\n");
   more = rs.next();
  }
 } catch (Exception e) {
  resultset = null;
```

```
  e.printStackTrace();
               System.out.println("eroare ");
 }
}
public void List()
{
  try {
         String sir="select * from studenti";
         ResultSet rs=stmt.executeQuery(sir);
         dispResultSet(rs);
         rs.close();
         if (resultset!=null)  textArea.setText(resultset);
         else  textArea.setText("A aparut o eroare");
         textArea.repaint();

      }
  catch(SQLException ex)
     {
        System.err.println("SQLException:" + ex.getMessage());
     }
  catch (java.lang.Exception ex) {
  ex.printStackTrace();

 }
}
public static void main( String args[] )
{
MyQuery mainFrame = new MyQuery();
mainFrame.setVisible(true);
}
}


import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.text.*;
import javax.swing.event.*;
import java.sql.*;

class MyInsert extends JFrame
                      implements DocumentListener,ActionListener
{
private JTextField field1;
private JTextField field2;
private JTextField field3;
private JButton button1;
private JButton button2;
private JButton button3;
private JLabel label1;
private JLabel label2;
private JLabel label3;
private Statement stmt;

public MyInsert(Statement stmt)
```

```
{
super();
this.stmt=stmt;
setTitle( "Insert Frame" );
setSize( 300, 300 );
setBackground( Color.gray );
this.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE) ;
JPanel topPanel = new JPanel();
topPanel.setLayout(null );
getContentPane().add( topPanel );

field1 = new JTextField();
field1.setBounds( 20, 40, 260, 25 );
field1.setFocusAccelerator( 'n' );
topPanel.add( field1 );
label1 = new JLabel( "Nume:");
label1.setBounds( 20, 15, 260, 20 );
label1.setLabelFor(field1 );
label1.setDisplayedMnemonic( 'N' );
topPanel.add( label1 );

field2 = new JTextField();
field2.setBounds( 20, 90, 260, 25 );
field2.setFocusAccelerator( 'g' );
topPanel.add( field2 );
label2 = new JLabel( "Grupa:" );
label2.setDisplayedMnemonic( 'G' );
label2.setBounds( 20, 65, 260, 20 );
label2.setLabelFor(field2 );
topPanel.add( label2 );

field3 = new JTextField();
field3.setBounds( 20, 140, 260, 25 );
field3.setFocusAccelerator( 'm' );
topPanel.add(field3);
label3 = new JLabel( "Media:" );
label3.setDisplayedMnemonic( 'M' );
label3.setBounds( 20, 115, 260, 20 );
label3.setLabelFor(field3 );
topPanel.add( label3 );

button1 = new JButton( "Insert" );
button1.setBounds( 20, 180, 70, 25 );
button1.setEnabled(false );
topPanel.add( button1 );
button1.addActionListener(this);

button2 = new JButton( "Reset" );
button2.setBounds( 110, 180, 70, 25 );
topPanel.add( button2 );
button2.addActionListener(this);

button3 = new JButton( "Close" );
button3.setBounds( 200, 180, 80, 25 );
topPanel.add( button3 );
```

```
button3.addActionListener(this);

// Add a document listener to the last two fields
Document document = field2.getDocument();
document.addDocumentListener( this );
Document document1 = field3.getDocument();
document1.addDocumentListener( this );
}

public void actionPerformed( ActionEvent e )
{

if( e.getSource() == button1 )
  {
     try {
         String sir="insert into studenti values('"
                     +field1.getText()+"','"+field2.getText()+","
                     +field3.getText()+")";
         stmt.executeUpdate(sir);
         JOptionPane dialog = new JOptionPane();
         dialog.showConfirmDialog( this, "Insertion completed!",
                                   "Plain", JOptionPane.DEFAULT_OPTION,
                                   JOptionPane.PLAIN_MESSAGE, null );
         }
  catch(SQLException ex)
     {
        System.err.println("SQLException:" + ex.getMessage());
        JOptionPane dialog = new JOptionPane();
        dialog.showMessageDialog( this,
                                  "Error! Insertion not completed.",
                                  "Error", JOptionPane.ERROR_MESSAGE );
     }
  }
else if( e.getSource() == button2 )
{
   field1.setText("");
   field2.setText("");
   field3.setText("");
}
else if( e.getSource() == button3 )
{
  this.dispose();
}
else
{
// Get the source of the action event
JLabel label = (JLabel)e.getSource();
// Give the associated component the focus
Component fieldComponent = label.getLabelFor();
fieldComponent.requestFocus();
}
}

// Handle insertions into the text field
public void insertUpdate( DocumentEvent event )
```

```
{
String sString = field2.getText();
String sString1 = field3.getText();
try {
int iValue = Integer.parseInt( sString );
double dValue=Double.parseDouble(sString1);
button1.setEnabled( true );
}
catch( NumberFormatException e )
{
button1.setEnabled( false );
}
}

// Handle deletions from the text field
public void removeUpdate( DocumentEvent event )
{
// Prevent the user from entering a blank field
if( field1.getText().length() == 0 || field2.getText().length() == 0
    || field3.getText().length() == 0)
button1.setEnabled( false );
else
{
insertUpdate( event );
}
}

public void changedUpdate( DocumentEvent event )
{}
}
```

III. MODUL DE LUCRU

1. Se editează codul sursă al programului Java folosind un editor de text disponibil(de ex., se poate utiliza Notepad).
2. Se salvează fişierul cu extensia **.java**. Fişierul trebuie salvat la următoarea locație: **c:\JBulider7\jdk1.3.1\bin**
3. Compilarea aplicației Java se va face din linia de comandă. Se poate proceda astfel. Se deschide o fereastră MS-Dos: Start ->Run, se tipăreşte **command** în căsuța de text şi se apasă butonul OK. Printr-o schimbare de directoare şi subdirectoare se trece la locația: **c:\JBulider7\jdk1.3.1\bin.** Sau, se lansează WindowsCommander. Se trece la locația **c:\JBulider7\jdk1.3.1\bin.** Se deschide o fereastră MS-Dos: Commander ->Run Dos.
4. Pentru compilare, se tipăreşte la prompter **javac nume_fişier_sursă.java** şi se apasă Enter. De ex., dacă fişierul se numeşte Test.java, se va scrie **javac Test.java**. În cazul în care programul conține erori acestea vor fi semnalate şi afişate în fereastra MS-Dos, după care va apare iar prompter-ul. Dacă programul nu conține erori şi compilarea se face cu succes, atunci va apare numai prompter-ul.

5. Pentru rularea unei aplicații Java stand-alone, se lansează interpretorul Java. Se tipăreşte la prompter următoarea comandă **java nume_clasă_care_conține_main** şi se apasă Enter. De ex., dacă clasa din program care conține metoda *main( )* se numeşte Test, se va scrie **java Test**.
6. Dacă programul Java este un applet, se editează fişierul .**html**. Se salvează în acelaşi director cu fişierul **.class** rezultat în urma compilării cu succes a fişierului sursă java. Apoi pentru rulare se poate utiliza **appletviewer nume.html.**
Alternativ, după compilarea aplicației Java, fişierul **.class** împreună cu fişierul **.html** pot fi mutate în orice alt director (nu trebuie neapărat să fie în **c:\JBulider7\jdk1.3.1\bin**). Se încarcă fişierul **.html** într-un browser Web (ex., Internet Explorer).

IV. TEMĂ

1. Se vor parcurge toate exemplele prezentate în platforma de laborator testându-se practic.
2. Modificați programul din laborator, optimizând interfața, astfel ca afişarea informațiilor din baza de date să se facă într-un tabel (JTable).