

TEHNOLOGII JAVA

PENTRU DEZVOLTAREA APLICAȚIILOR

LUCRARE DE LABORATOR 20

JDBC – Accesul la baze de date

I. SCOPUL LUCRĂRII

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

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

II. NOȚIUNI TEORETICE

1. Exemple

Se va utiliza aceeași bază de date **StudentiDB** creată în cadrul laboratorului 18. În continuare vom prezenta un program Java (incomplet) care implementează o interfață grafică prin care utilizatorul poate să formuleze fraze SQL și să primească răspunsul.

```
import java.awt.*;
import java.awt.event.*;

public class DBQueryGUI extends Frame implements
    WindowListener, ActionListener {
    TextField sqlExec;
    Button executeButton;
    Label label1;
    static TextArea edit1;
    Label label2;
    Button quitButton;
    Button resetButton;
    SimpleSelect ob;

    public void init() {
        this.addWindowListener(this);
        LayoutManager layoutman = new BorderLayout();
        this.setLayout(layoutman);
        Panel p1 = new Panel();
        Panel p2 = new Panel();
        Panel p3 = new Panel();
        sqlExec = new TextField(50);
        edit1 = new TextArea(12, 80);
        label1 = new Label("Enter an SQL statement:");
```

```

label2 = new Label("Output");
executeButton = new Button("Execute");
quitButton = new Button("Quit");
resetButton = new Button("Reset");
p1.add(label1);
p1.add(sqlExec);
p2.add(executeButton);
p2.add(resetButton);
p2.add(quitButton);
p3.add(label2);
p3.add(edit1);
this.add("North", p1);
this.add("Center", p3);
this.add("South", p2);
executeButton.addActionListener(this);
quitButton.addActionListener(this);
resetButton.addActionListener(this);
}

public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();
    if (command.equals("Execute"))
        clickedExecuteButton();
    if (command.equals("Quit"))
        clickedQuitButton();
    if (command.equals("Reset"))
        sqlExec.setText("");
}

public void clickedExecuteButton() {
    edit1.setText("");
    edit1.repaint();
    if (ob == null) {
        try {
            ob = new SimpleSelect();
        } catch (ClassNotFoundException cnfe) {
            cnfe.printStackTrace();
            return;
        }
    }
    if (!ob.open()) {
        edit1.setText("Nu se poate stabili o conexiune cu baza de
                        date !");
        return;
    }
    String sqlString, result;

    sqlString = sqlExec.getText(); result = ob.queryDatabase(sqlString);
    if (result != null)
        edit1.setText(result);
    else
        edit1.setText("A aparut o eroare !");
}

public void clickedQuitButton() {
    if (ob != null)
        ob.close();
}

```

```

    System.exit(0);
}

public static void main(String argv[]) {
    DBQueryGUI sql = new DBQueryGUI();
    sql.init();
    sql.pack();
    sql.setTitle("JDBC/ODBC GUI");
    sql.show();
}

public void windowActivated(WindowEvent e) {}
public void windowClosed(WindowEvent e) {}
public void windowClosing(WindowEvent e) { System.exit(0);
}

public void windowDeactivated(WindowEvent e) {}
public void windowDeiconified(WindowEvent e) {}
public void windowIconified(WindowEvent e) {}
public void windowOpened(WindowEvent e) {}
}

```

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 fișierul cu extensia **.java**. Fișierul 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 tipărește **command** în căsuța de text și se apasă butonul OK. Printr-o schimbare de directoare și 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 tipărește la prompter **javac nume_fișier_sursă.java** și se apasă Enter. De ex., dacă fișierul se numește Test.java, se va scrie **javac Test.java**. În cazul în care programul conține erori acestea vor fi semnalate și afișate în fereastra MS-Dos, după care va apare iar prompter-ul. Dacă programul nu conține erori și compilarea se face cu succes, atunci va apare numai prompter-ul.
 5. Pentru rularea unei aplicații Java stand-alone, se lansează interpretorul Java. Se tipărește la prompter următoarea comandă **java nume_clasă_care_conține_main** și se apasă Enter. De ex., dacă clasa din program care conține metoda *main()* se numește Test, se va scrie **java Test**.
 6. Dacă programul Java este un applet, se editează fișierul **.html**. Se salvează în același director cu fișierul **.class** rezultat în urma compilării cu succes a fișierului sursă java. Apoi pentru rulare se poate utiliza **appletviewer nume.html**.
- Alternativ, după compilarea aplicației Java, fișierul **.class** împreună cu fișierul **.html** pot fi mutate în orice alt director (nu trebuie neapărat să fie în **c:\JBulider7\jdk1.3.1\bin**). Se încarcă fișierul **.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. Completați programul din laborator, implementând clasa **SimpleSelect**. Metoda principală a clasei este **queryDatabase()** care primește ca parametru un obiect de tipul String ce reprezintă o frază SQL și returnează un obiect SQL care reprezintă rezultatul sau o excepție.