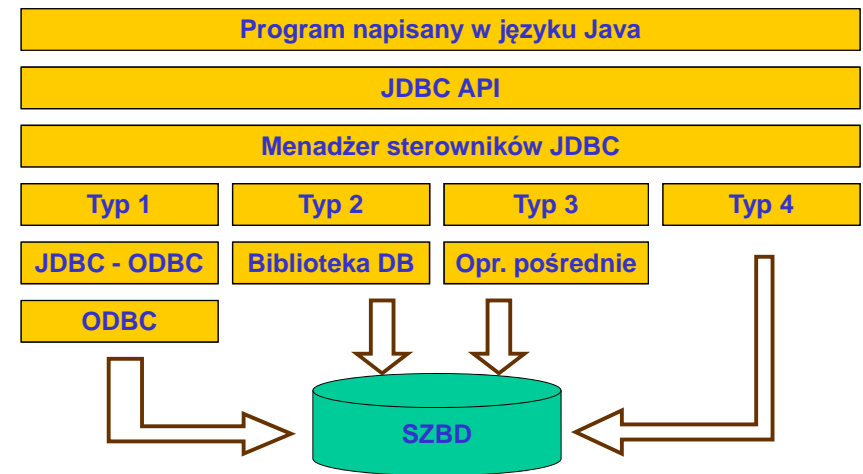


Rozdział 21 JDBC

Wprowadzanie do JDBC, kursory zwykłe, kursory przewijane, obsługa błędów, przetwarzanie transakcyjne, wywoływanie kodu składowanego



Typy sterowników



Historia

- JDBC 1.1 – JDK 1.1, rok 1997,
- JDBC 2.1 – JDK 1.2, rok 1999,
- JDBC 3.0 – JDK 1.4, rok 2001,
- JDBC 4.0 – Java SE 6, rok 2006,
- JDBC 4.1 – Java SE 7, rok 2011,
- JDBC 4.2 – Java SE 8.



Podstawowe kroki aplikacji

1. dołączenie do źródła danych
2. określenie parametrów połączenia
3. przygotowanie i wykonanie polecenia
4. pobranie wyników
5. odłączenie od źródła danych



Nawiązywanie i zamykanie połączeń

```
① import java.sql.*;

② Class.forName("oracle.jdbc.driver.OracleDriver");

③ Connection con = DriverManager.getConnection(
    "jdbc:oracle:thin:@dblab.cs.put.poznan.pl:1521:dblab11g",
    "scott", "tiger");

④ con.close();
```



Wykonywanie zapytań

```
① Statement stmt = con.createStatement() ;

② ResultSet rs=stmt.executeQuery(
    "SELECT nazwisko,placa_pod FROM pracownicy");

③ while (rs.next()) {
    String nazwisko=rs.getString("NAZWISKO");
    float placa=rs.getFloat(2);
    System.out.println(nazwisko+" "+placa);
}

④ rs.close();
    stmt.close();
```



Wykonywanie aktualizacji danych

```
Statement stmt = con.createStatement() ;
int changes;

① changes=stmt.executeUpdate(
    "INSERT INTO pracownicy(id_prac,nazwisko)+"
    "VALUES(300,'Zieliński')");
System.out.println("Wstawiono "+changes+" krotek.");

② changes=stmt.executeUpdate("UPDATE pracownicy " +
    "SET placa_pod=placa_pod*1.5");
System.out.println("Zmodyfikowano "+changes+" krotek.");

③ changes=stmt.executeUpdate(
    "DELETE FROM pracownicy WHERE id_prac=300");
System.out.println("Usunięto "+changes+" krotek.");

stmt.close();
```



Zarządzanie transakcjami (1)

```
private static int ilePracownikow(Connection con)
    throws SQLException {
    int result;
    Statement stmt=con.createStatement();
    ResultSet rs=stmt.executeQuery(
        "SELECT COUNT(*) FROM pracownicy");
    rs.next();
    result=rs.getInt(1);
    rs.close();
    stmt.close();
    return result;
}
```



Zarządzanie transakcjami (2)

```
1 con.setAutoCommit(false);  
Statement stmt=con.createStatement();  
2 System.out.println(ilePracownikow(con)); // 14  
  
changes=stmt.executeUpdate(  
    "DELETE FROM pracownicy WHERE id_prac=150");  
3 System.out.println("Usunieto "+changes+" krotek.");  
con.rollback();  
System.out.println(ilePracownikow(con)); //14  
  
changes=stmt.executeUpdate(  
    "DELETE FROM pracownicy WHERE id_prac=150");  
4 System.out.println("Usunieto "+changes+" krotek.");  
con.commit();  
System.out.println(ilePracownikow(con)); //13
```

Prekompilowane polecenia (1)

```
1 PreparedStatement stmt = con.prepareStatement(  
    "SELECT nazwisko FROM pracownicy WHERE id_prac=?");  
  
ResultSet rs;  
stmt.setInt(1,140); rs=stmt.executeQuery();  
2 while (rs.next()) {  
    System.out.println(rs.getString("NAZWISKO"));  
}  
rs.close();  
  
stmt.setInt(1,170); rs=stmt.executeQuery();  
3 while (rs.next()) {  
    System.out.println(rs.getString("NAZWISKO"));  
}  
rs.close();  
stmt.close();
```

Prekompilowane polecenia (2)

```
1 PreparedStatement stmt = con.prepareStatement(  
    "UPDATE pracownicy SET placa_pod= ?, etat = ?  
    WHERE id_prac= ?");  
  
2 stmt.setFloat(1,2000);  
stmt.setString(2,"PROFESOR");  
stmt.setInt(3,140);  
int changes =stmt.executeUpdate();  
  
System.out.println("Zmodyfikowano "+changes+ " krotek");  
  
3 stmt.close();
```

Aktualizowanie wsadowe (1)

```
1 PreparedStatement stmt = con.prepareStatement(  
    "UPDATE pracownicy SET placa_pod=placa_pod * ?  
    WHERE id_prac = ? ");  
  
2 stmt.setFloat(1,new Float(0.9));  
stmt.setInt(2,130);  
stmt.addBatch();  
  
stmt.setFloat(1,new Float(1.5));  
stmt.setInt(2,150);  
stmt.addBatch();  
  
3 int []changes =stmt.executeBatch();  
System.out.println("Zmodyfikowano"+changes[0]+", "  
    +changes[1]+" krotek");
```

Aktualizowanie wsadowe (2)

```
1 Statement stmt = con.createStatement();

stmt.addBatch("update pracownicy set placa_pod=1000" +
"where nazwisko = 'MORZY'");

2 stmt.addBatch("delete from pracownicy where " +
"nazwisko = 'HAPKE'");

3 int []changes =stmt.executeBatch();
...
```

Zwiększanie przesyłanej liczby krotek

```
1 Statement stmt = con.createStatement();
((OracleStatement)stmt).setRowPrefetch(10);

ResultSet rs=stmt.executeQuery(
"SELECT nazwisko FROM pracownicy");

2 while (rs.next()) System.out.println(rs.getString(1));

rs.close();
stmt.close();
```

Przewijalne zbiory wyników

```
1 Statement stmt = con.createStatement(
ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY);

ResultSet rs = stmt.executeQuery(
"SELECT nazwisko FROM pracownicy");

2 rs.absolute(1); System.out.println(rs.getString(1));
rs.absolute(-2); System.out.println(rs.getString(1));
rs.relative(-1); System.out.println(rs.getString(1));
rs.relative(2); System.out.println(rs.getString(1));

3 rs.afterLast();
while(rs.previous())System.out.println(rs.getString(1));
```

Modyfikowalne zbiory wyników

```
1 Statement stmt = con.createStatement(
ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);

2 ResultSet rs = stmt.executeQuery(
"SELECT id_prac,nazwisko,placa_pod FROM pracownicy");

rs.last();

3 rs.updateString(2,rs.getString(2)+"123");
rs.updateFloat("PLACA_POD", new Float(rs.getFloat(3)*0.9));
rs.updateRow();// albo rs.cancelRowUpdates();

rs.moveToInsertRow();
rs.updateInt(1,310);

4 rs.updateString(2,"Zieliński");
rs.updateFloat(3,new Float(1234.5));
rs.insertRow();
```

Wyjątki

```
Statement stmt;  
try {  
    stmt=con.createStatement();  
    stmt.executeUpdate(  
        "DELETE FROM pracownicy WHERE nazwisko='WEGLARZ'");  
} catch (SQLException e) {  
    ① System.out.println("Stan SQL: "+e.getSQLState());  
    ② System.out.println("Kod błędu: "+e.getErrorCode());  
    ③ System.out.println("Komunikat: "+e.getMessage());  
}  
stmt.close();
```



Procedury składowane

```
CallableStatement stmt = con.prepareCall(  
    ① "{call WstawZespol(?,?,?)}");  
  
    ② stmt.setInt(1,60);  
    stmt.setString(2, "NOWY ZESPÓŁ");  
    stmt.setString(3, "PIOTROWO 3A");  
  
    ③ stmt.execute();  
  
    ④ stmt.close();
```



Funkcje składowane

```
    ① CallableStatement stmt = con.prepareCall(  
        "{? = call PoliczPracownikow(?)}"");  
  
    ② stmt.setString(2, "ALGORYTMY");  
  
    ③ stmt.registerOutParameter(1, Types.INTEGER);  
  
    ④ stmt.execute();  
  
    ⑤ int vLiczbaPracownikow = stmt.getInt(1);  
  
    ⑥ stmt.close();
```

