Adatbázisok webalkalmazásokban

ANTAL Margit

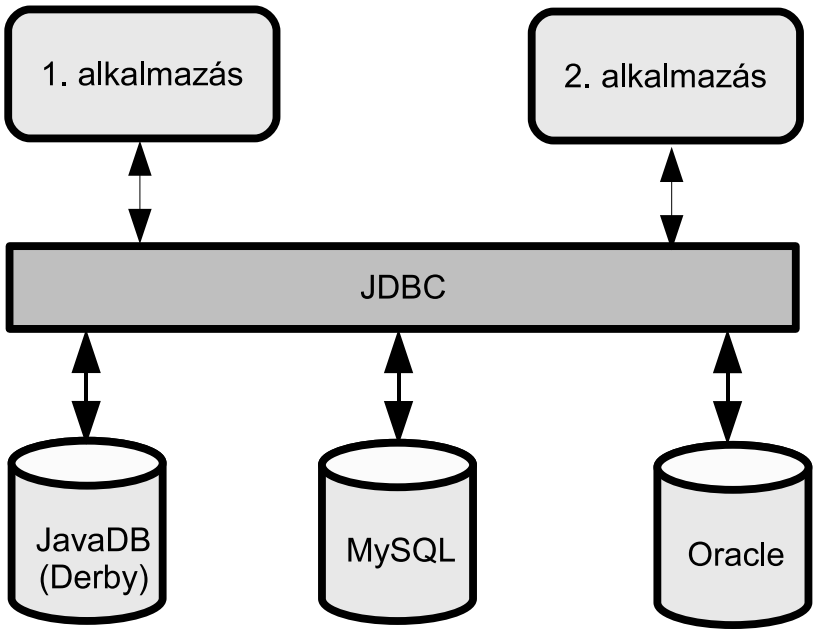
Sapientia - EMTE, Pannon Forrás ,,Egységes erdélyi felnőttképzés a Kárpát-medencei hálózatban”

2010

Adatbázisok webalkalmazásokban

- A JDBC API
- A Data Access Object tervezési minta
- Adatforrás - DataSource

JDBC architektúra



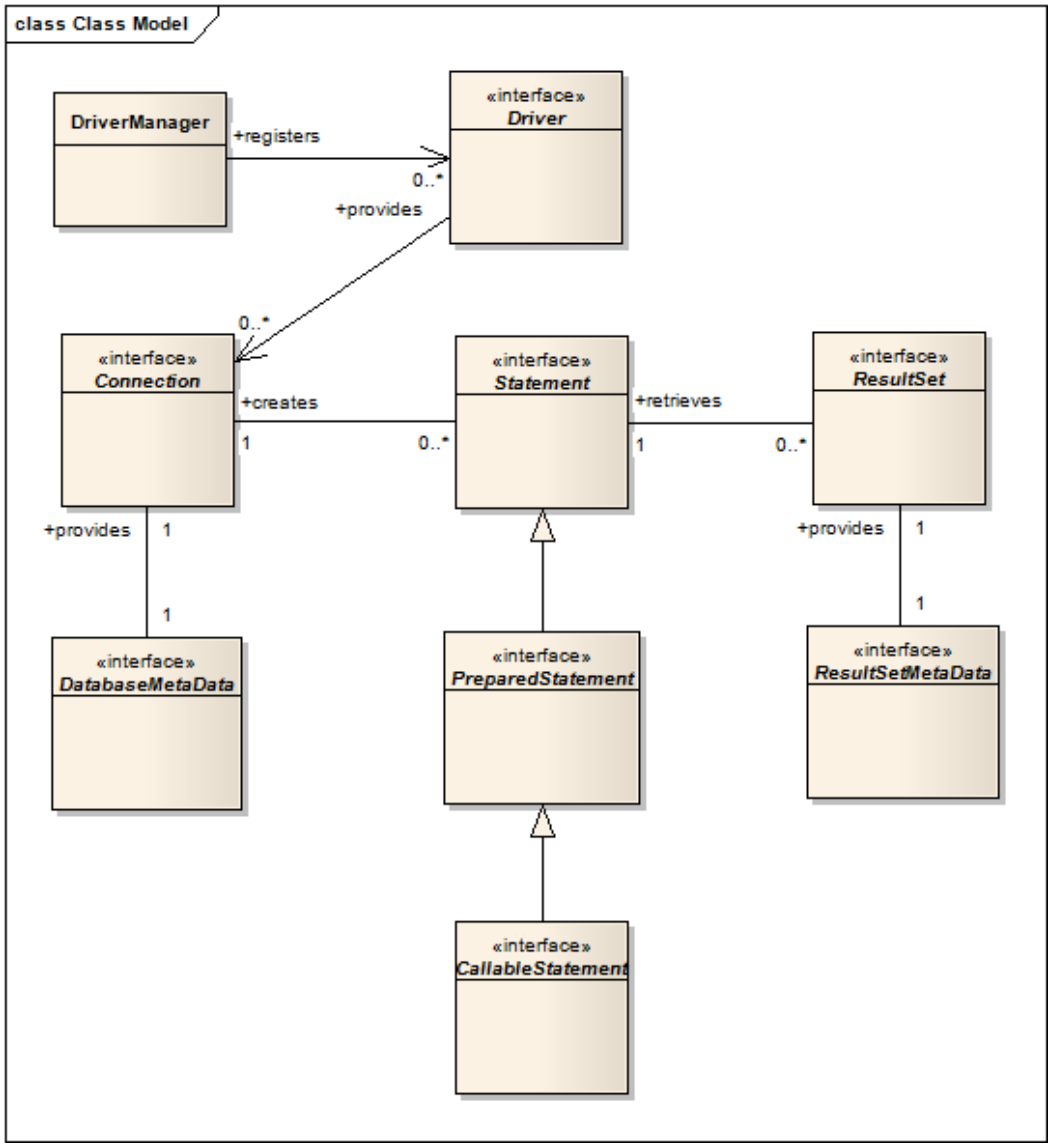
A JDBC API

- `java.sql` -adatbázisok eléréséhez szükséges alapvető osztályokat tartalmazza
- `javax.sql` -haladó szintű osztályokat tartalmaz (pl. `DataSource`)

Osztályozás

- adatbázis-kapcsolat megteremtését segítő típusok
- SQL utasítások végrehajtását segítő típusok
- SQL lekérdezések eredményeinek feldolgozását segítő típusok

JDBC API, alapvető osztályok



JDBC meghajtóprogramok

Példák

- Derby (Java DB): `derbyclient.jar`
- MySql : `mysql-connector-java-5.1.7-bin.jar`

JDBC adatbáziskezelés

1. adatbázis-meghajtó betöltése (driver regisztráció)
2. adatbázis-kapcsolat kiépítése
3. egy SQL művelet végrehajtásához szükséges Statement példány létrehozása:
   - Statement
   - PreparedStatement
   - CallableStatement
4. SQL eredmények feldolgozása
5. Kapcsolat lezárása

1. Adatbázis-meghajtó betöltése

Driverek

- Derby:
  ```
  Class.forName("org.apache.derby.jdbc.ClientDriver");
  ```
- MySql: `Class.forName("com.mysql.jdbc.Driver");`

2. Adatbázis-kapcsolat kiépítése

Szintaxis

`jdbc:<alprotokoll>:<adatbázis hivatkozás>`

- Derby adatbázis URL:
  `jdbc:derby://localhost:1527//distedu`
- MySql adatbázis URL:
  `jdbc:mysql://localhost:3306/distedu`

```
String url = "jdbc:derby://localhost:1527//distedu";
Connection con = DriverManager.getConnection(
                    url,"username","password");
```

3. Statement példány létrehozása

```
Statement stat  = con.createStatement();
```

4. SQL eredmények feldolgozása

Szintaxis:

```
ResultSet executeQuery( String sql )
int executeUpdate( String sql )
```

- `executeQuery`: SELECT
- `executeUpdate`: UPDATE, DELETE, CREATE TABLE, DROP TABLE

A `ResultSet` objektum

Kurzor mozgató műveletek

- `next()`: a következő rekordra lép
- `previous()`: az előző rekordra lép
- `last()`: az utolsó rekordra lép
- `first()`: az első rekordra lép

Adattábla

```
CREATE TABLE course (
    id int NOT NULL GENERATED ALWAYS AS IDENTITY,
    name varchar(100) NOT NULL,
    description varchar(200) NOT NULL,
    price double NOT NULL
)
```

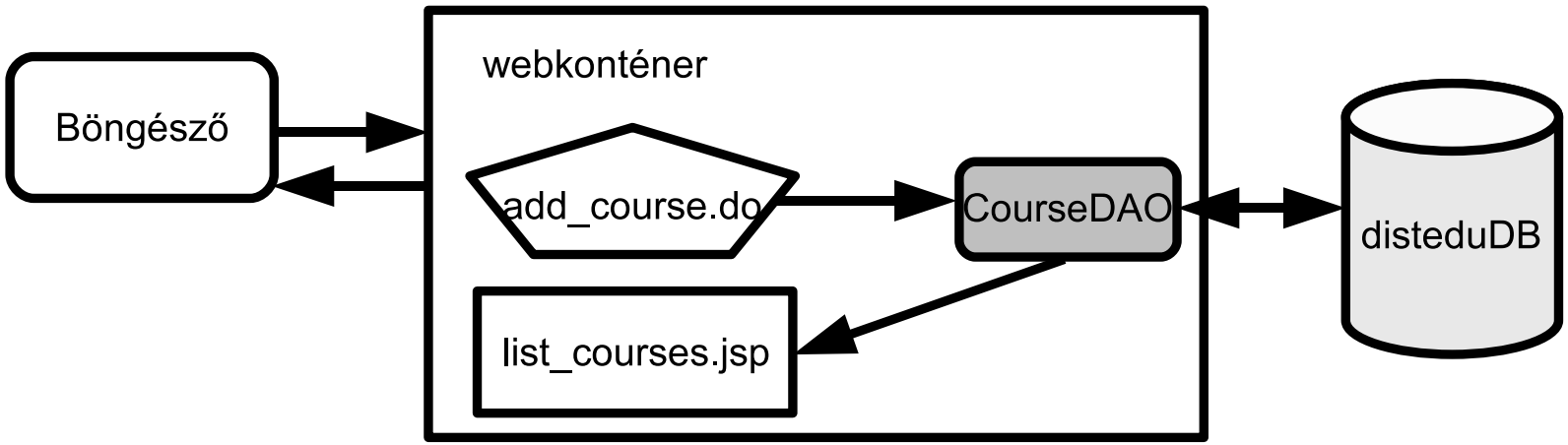
Példa lekérdezésre

```
try{
    Statement stmt = con.createStatement();
    ResultSet rs =
      stmt.executeQuery("select * from course");
    while( rs.next() ){
     System.out.println(
     "ID: "+rs.getInt(1)+
     "NAME: "+rs.getString(2)+
     "DESCRIPTION: "+rs.getString(3)+
     "PRICE: "+rs.getDouble(4)
    );
  }
}
catch (Exception e) {
  e.printStackTrace();
}
```

Példa adatmanipulációra

```
try{
    Statement stmt = con.createStatement();
    int n = stmt.executeUpdate(
    "insert into course (name, description, price)
     values ('Java SE','Java Standard Edition', 1000)");
    //Ha n erteke 1, akkor sikeres volt a beszuras
}
catch (Exception e) {
  e.printStackTrace();
}
```

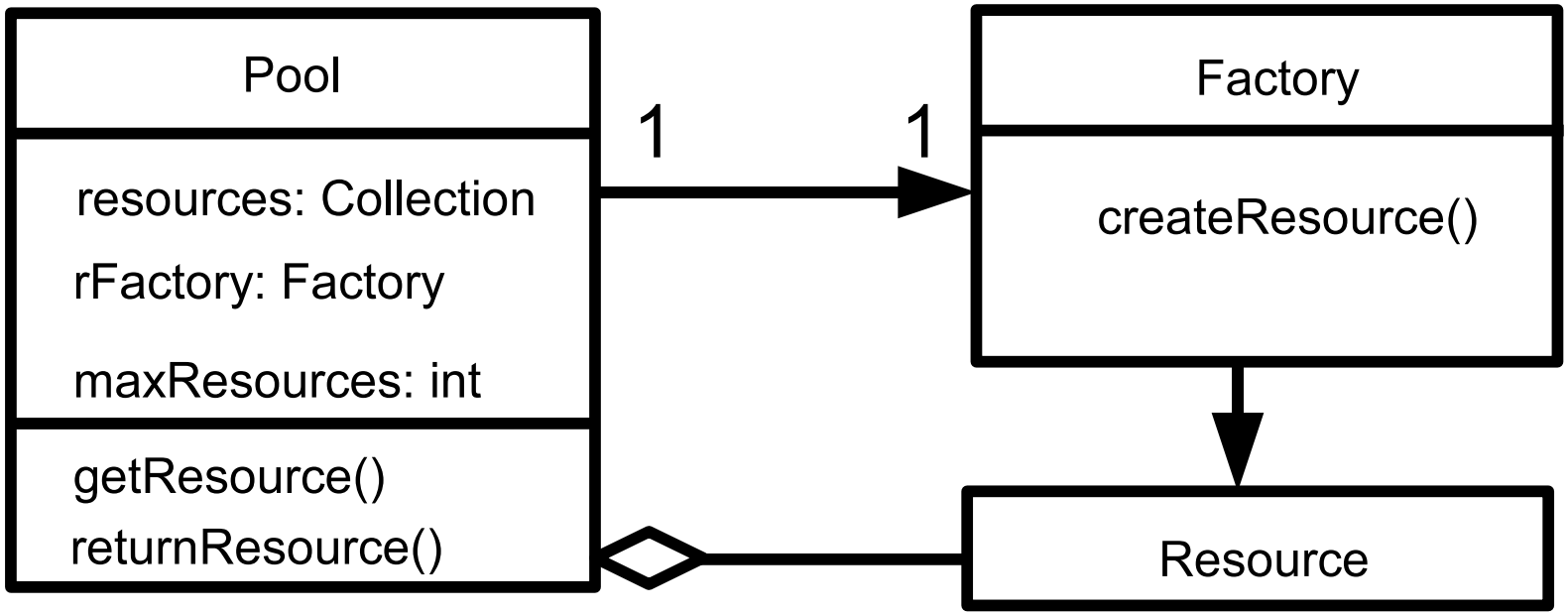
A Data Access Object tervezési minta



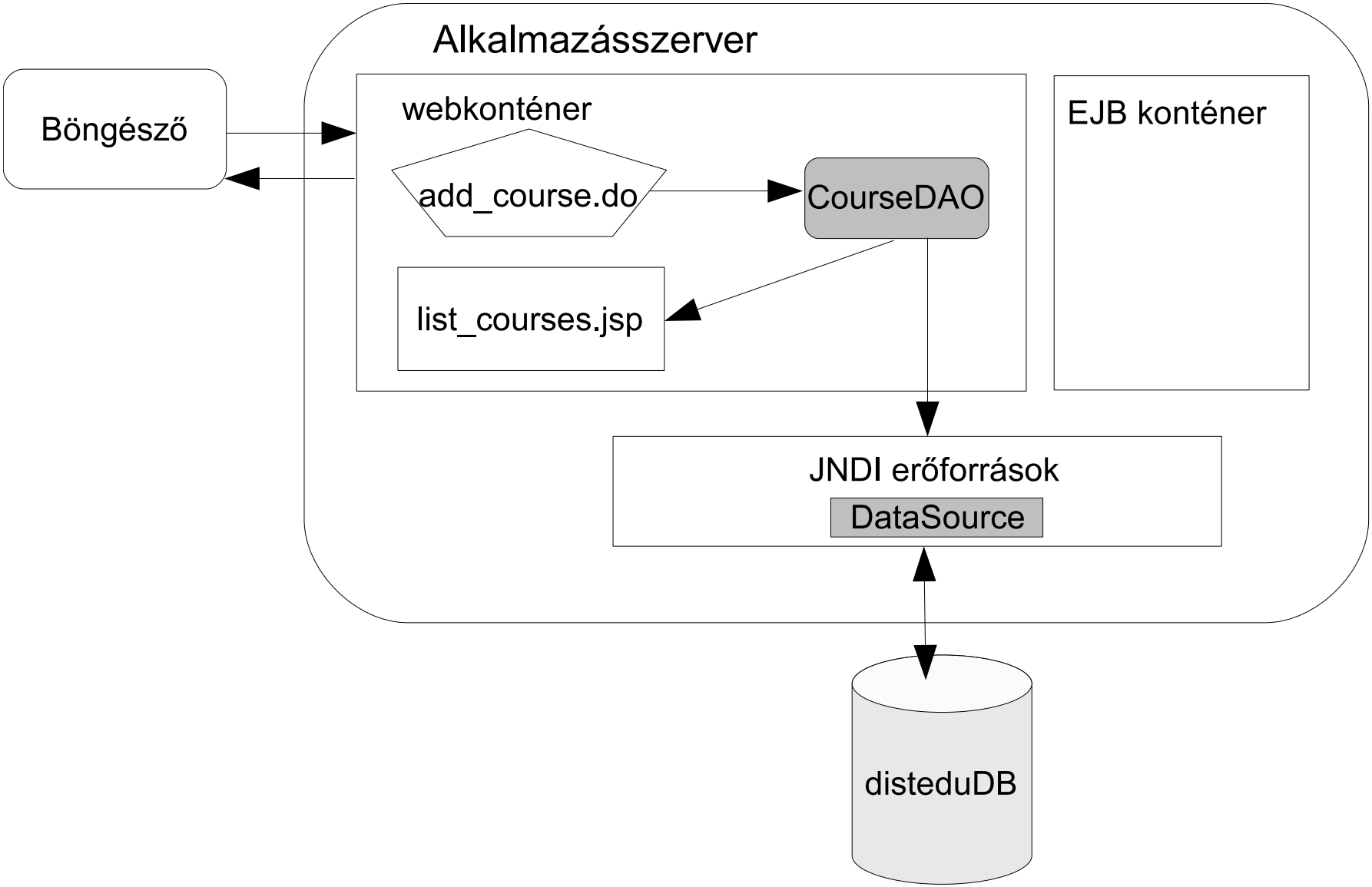
A CourseDAO osztály

```
public class CourseDAO {
    private String driver;
    private String url;
    private String userName;
    private String password;
    private Connection con;

    public CourseDAO(){ ... }

    public void connect(){...}

    public void disConnect(){...}

    public List<Course> getAllCourses(){...}

    public boolean insertCourse(Course c) {
}
```

Connection Pool



Adatforrás - DataSource



Adatbázis-kapcsolat DataSource komponenssel

Glassfish

```
DataSource datasource = null;

Context ctx = new InitialContext();
if( ctx == null )
        throw new RuntimeException
          ("JNDI Context coluld not be found");
datasource = DataSource) ctx.lookup("jdbc/disteduDS");

Connection con = null;
con = datasource.getConnection();
```

Adatbázis-kapcsolat DataSource komponenssel

Tomcat 6.0

```
DataSource datasource = null;

Context initCtx = new InitialContext();
Context envCtx =
  (Context) initCtx.lookup("java:comp/env");
datasource =
  (DataSource) envCtx.lookup("jdbc/disteduDS");

Connection con = null;
con = datasource.getConnection();
```

Távoktatás alkalmazás

Feladatok: 177. oldal

1. Az ApplicationListener osztály törlése
2. Adatbázis létrehozása
3. A Course adattábla feltöltése
4. A model.Course osztály módosítása
5. A CourseDAO modell osztály létrehozása
6. A list_courses.jsp lap módosítása
7. Az AddCourse szervlet módosítása
8. Tanfolyamok törlése: delete_course.jsp (űrlap) és a DeleteCourse szervlet (URL: delete_course.do)