

## JAVA TECHNOLÓGIÁK 2009.

### 3. GYAKORLAT

---

#### Cél:

- adatbázist használó webalkalmazás készítése
- megjelenítési komponens megvalósítása szervlettel
- a `web.xml` telepítésleíró

Készítsen *webalkalmazást*, amely oktatási szolgáltatást nyújtó cég munkáját segíti. Az alkalmazást kétféle felhasználó fogja kezelni: *adminisztrátor* és olyan személyek, akik érdeklődnek a tanfolyamok iránt. Ezeket a továbbiakban *hallgatóknak* fogjuk nevezni.

Az adminisztrátor karbantartja a tanfolyamokat, tehát a következő műveleteket kell végeznie: új tanfolyam felvitele, törlés.

A hallgatók két műveletet végezhetnek: tanfolyamok listájának megtekintése, illetve regisztráció egy adott tanfolyamra.

#### 1. Az `index.jsp` módosítása

Cserélje le a lap törzsét a következőre:

```
<body>
  <h2>Hallgato</h2>
  <ul>
    <li><a href="list_courses.view">Tanfolyamok listazasa</a></li>
    <li><a href="TBD">Feliratkozas tanfolyamra</a></li>
  </ul>
  <h2>Adminisztrator</h2>
  <ul>
    <li><a href="TBD">Uj Tanfolyam</a></li>
    <li><a href="TBD">Tanfolyam torlese</a></li>
  </ul>
</body>
```

#### 2. Adatréteg

Hozzon létre egy **DISTEDUDB** nevű adatbázist és ebben egy **COURSE** nevű táblát:

```
CREATE TABLE Course (
  id          INT NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
  name        VARCHAR(100) NOT NULL,
  description VARCHAR(200) NOT NULL,
  price       DOUBLE NOT NULL
);
```

Töltse fel a táblát:

```
INSERT INTO Course (name, description, price) VALUES('C++', 'Generic
Programming', 500);
INSERT INTO Course (name, description, price) VALUES('Java SE', 'Object
Oriented Programming', 500);
INSERT INTO Course (name, description, price) VALUES('Java EE', 'Web
Programming', 1000)
```

# JAVA TECHNOLÓGIÁK 2009.

## 3. GYAKORLAT

---

### 3. Modell komponensek

Ezeket helyezze a `distedu.model` csomagba!

Készítsen egy `Course` osztályt, amely egy tanfolyam adatainak tárolására alkalmas:

- az attribútumok megegyeznek a táblában használt mezőnevekkel
- `get` és `set` metódus minden attribútumhoz
- paraméter nélküli konstruktor
- négyparaméteres konstruktor
- `toString` metódus

Készítsen egy `CourseDAO` osztályt. Ez az osztály felelős a `Course` táblával végezhető műveletekért. Elkészítési javaslat: ld. 2. gyakorlat `StudentDAO` osztály.

A `CourseDAO` osztály tartalmazzon egy `getCourses` metódust a következő szignatúrával:

```
public List<Course> getCourses();
```

### 4. Megjelenítési komponensek

Ezeket helyezze a `distedu.view` csomagba!

- Adjon hozzá a webalkalmazáshoz egy szervletet, amelynek neve `ViewCoursesServlet` és helyezze a `distedu.view` csomagba. Az új szervlet készítésének utolsó információs ablakában (Configure Servlet Deployment) változtassa meg az alapértelmezett URL `pattern` mezőt. Legyen az új érték: `/list_courses.view`
- Ellenőrizze a `web.xml` telepítésleíró tartalmát a szervlet hozzáadása után.
- Módosítsa a szervlet `doGet` metódusát úgy, hogy a szervlet által előállított válasz tartalmazza az összes tanfolyam adatait:
  - Példányosítsa a `CourseDAO` osztályt
  - Hívja meg a `getCourses` metódust, majd az így kapott listát dolgozza fel iterátor segítségével

Mivel adatbázist használunk a `derbyclient.jar` driver-t hozzá kell adni a projekthez. (Ezt megteheti a projekt tulajdonságaiban `Libraries -Add Jar-Folder`)