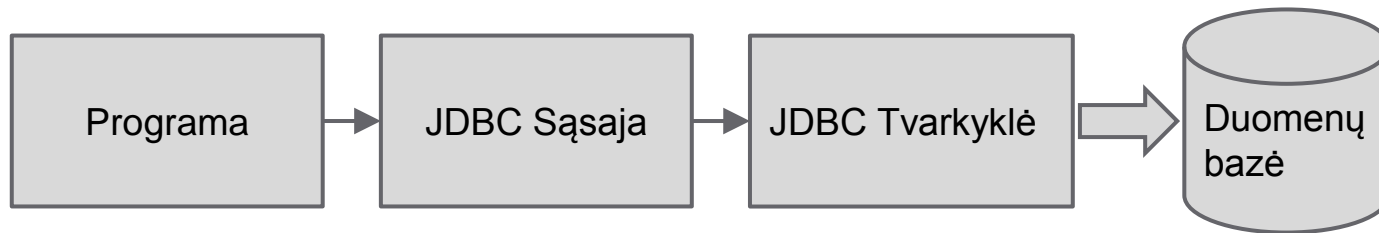


JDBC

Parengė Giedrius Graževičius

JDBC

- Standartinė sąsaja tarp Java programų ir reliacinių duomenų bazių



Pagrindiniai JDBC interfeisai

- Connection
- Statement, PreparedStatement
- ResultSet

Connection

- Reprezentuoja aktyvią sąsają su duomenų baze

Connection pavyzdys

```
Connection connection =  
    DriverManager.getConnection(  
        "jdbc:h2:file:/tmp/imones.db", // JDBC adresas  
        "sa", // Vartotojo vardas  
        ""); // Slaptažodis
```

Statement

- Reprezentuoja vieną duomenų bazės užklausą
 - Tas pats objektas gali būti panaudotas kelioms užklausoms, tačiau kiekvienu momentu aktyvi yra tik viena

Statement pavyzdys

```
Statement statement = connection.createStatement();  
statement.execute("delete from lentele");
```

PreparedStatement

- Naudojamas daugiakartiniam vienos užklauso vykdymui
- Gali saugiai priimti parametrus

PreparedStatement pavyzdys

```
PreparedStatement statement =  
    connection.prepareStatement(  
        "delete from lentele where laukas = ?");  
  
statement.setString(1, "Reikšmė");  
  
statement.execute();  
  
statement.close();
```

PreparedStatement parametrai

```
PreparedStatement statement =  
    connection.prepareStatement(  
        "delete from lentele where laukas = ? and kitas = ?");  
statement.setInt(1, 100);  
statement.setDate(2, new Date());
```

Parametrai numeruojami nuo 1!

```
PreparedStatement statement =  
    connection.prepareStatement(  
        "select * from darbuotojas where vardas=?");  
  
statement.setString(1, "Kestas");  
  
ResultSet rs = statement.executeQuery();
```

ResultSet

- Reprezentuoja aktyvų rezultatą
- Leidžia peržiūrėti rezultatus po eilutę
 - Veikia tarsi kolekcijos iteratorius iš duomenų bazės grąžinamiems rezultatams

ResultSet pavyzdys

```
PreparedStatement statement =  
    connection.prepareStatement(  
        "select * from darbuotojas where vardas=?");  
statement.setString(1, "Kestas");  
ResultSet rs = statement.executeQuery();
```

```
while (rs.next()) {  
    System.out.println(rs.getString("pavarde"));  
}
```

Atspausdins visų
Kęstų pavardes.

ResultSet duomenų prieiga

```
while (rs.next()) {  
    System.out.println(rs.getString("pavarde"));  
    System.out.println(rs.getDate("darbaNuo"));  
}
```

```
rs.getTipas("stulpelio pavadinimas");  
    arba  
rs.getTipas(stulpelio numeris); // Nepatartina naudoti, kuomet  
    stulpeliai nevardinami, o naudojama * (visų stulpelių išrinkimas)
```

Atnaujinimas per ResultSet

```
Statement statement =  
    connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,  
        ResultSet.CONCUR_UPDATABLE); // CONCUR_UPDATABLE  
ResultSet rs = statement.executeQuery("select * from darbuotojas");  
  
while (rs.next()) {  
    System.out.println(rs.getString("pavarde"));  
    rs.updateString("pavarde", rs.getString("pavarde").toUpperCase());  
    rs.updateRow();  
}
```

Užduotys

1. Išvesti visų darbuotojų vardus ir pavardes
2. Išvesti visus projektus ir juose dirbančius žmones:
= Projekto pavadinimas =
 1. Jonas Jonaitis
 2. Petras Petraitis
- 3.* Parašyti programą, leidžiančią įvesti naujus darbuotojus
- 4.* Papildyti #3 galimybe priskirti darbuotoją projektui