

JDBC - tworzenie połączenia - sposób nr 2

java.sql.Driver (sterownik do łączenia aplikacji z bazą danych)
javax.sql.DataSource (fabryka do tworzenia połączeń z bazą danych, preferowany sposób tworzenia połączeń, "pod spodem" wykorzystuje sterownik do stworzenia połączenia)
java.sql.Connection (połączenie z bazą danych, informacje o bazie danych, wywoływanie zapytań do bazy danych)

- DataSource umożliwia nam oddzielenie konfiguracji połączenia do bazy danych od pracy na tym połączeniu
- Dobrą praktyką jest przechowywanie parametrów dostępu do bazy danych w osobnym pliku (np.: database.properties) - tak żeby można było zmieniać konfigurację podczas wdrożeń na różne środowiska bez potrzeby zmiany kodu.

Zadanie nr 2:

1. Otwórz klasę ConnectionViaDataSource i uzupełnij konfigurację bazy danych tak żeby połączyć się ze swoją lokalną bazą MySQL
2. Uruchom metodę main i sprawdź czy połączenie do bazy danych się powiodło
3. Przeanalizuj kod - czy widzisz różnicę między użyciem DataSource a DriverManager?
4. Stwórz klasę ConnectionFactory, która będzie posiadać jedną metodę getConnection(), wykorzystaj ją do pobrania DataSource w klasie ConnectionViaDataSource
5. Przenieś parametry bazy danych do pliku database.properties (w katalogu resources):
 - a. Utwórz plik database.properties i zapisz w nim pary klucz-wartość dla każdego parametru:

```
pl.sda.jdbc.db.server={SERVER}
pl.sda.jdbc.db.name={DB_NAME}
pl.sda.jdbc.db.user={USER}
pl.sda.jdbc.db.password={PASSWORD}
```
 - b. Wczytaj parametry z pliku database.properties za pomocą klasy java.util.Properties i metody: Properties.load(InputStream inStream)
 - c. inStream można pobrać wykorzystując właściwości classloadera:

```
ConnectionFactory.class.getResourceAsStream(name);
```
 - d. Po wczytaniu pliku wystarczy pobrać parametry za pomocą metody Properties.getProperty(String key)
 - e. Uruchom i sprawdź czy program działa