

JDBC - podsumowanie

→ **sterownik** - zgodny z bazą danych, z którą chcemy się połączyć, np.:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.11</version>
</dependency>
```

→ **logger** - przyda się do logowania komunikatów:

```
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>1.2.3</version>
</dependency>
```

→ **konfiguracja bazy danych** - zewnętrzny plik z parametrami

→ **DataSource** - dostarczony w bibliotece sterownika, np.: `MysqlDataSource`

→ **fabryka** - dla obiektów `Connection`, wczytywanie konfiguracji z pliku

→ **DTO** - klasa przechowująca dane z tabelki

→ **DAO** - klasy odpowiedzialne za dostęp do bazy danych

Zadanie nr 4:

1. Stwórz nowy moduł Maven o nazwie: **zoo-keeper**, następnie stwórz nową lokalną bazę danych o nazwie: **zoo_keeper**
2. Przygotuj kod w module **zoo-keeper** do pracy z bazą danych zgodnie z rozpiską podaną na początku strony (zależności, plik z parametrami, fabryka połączeń)
3. W bazie danych stwórz (dowolną metodą) tabelkę **animals_types** z kolumnami: *id*, *name*. Stwórz klasę **AnimalType**, która będzie zawierała wszystkie pola jakie ma tabelka. Stwórz klasę **AnimalsTypesDao** i dodaj do metody, które pozwolą:
 - a. Dodać nowy rodzaj - argument: **AnimalType**
 - b. Pobrać z bazy wszystkie rodzaje zwierząt - lista obiektów **AnimalType**
 - c. Pobrać z bazy jeden rodzaj o podanym id - zwraca obiekt **AnimalType**
 - d. Usunąć rodzaj z bazy danych - argument: **AnimalType**
 - e. Zmienić nazwę rodzaju o podanym id - argument: **AnimalType**
4. W bazie danych stwórz (dowolną metodą) tabelkę **animals** z kolumnami: *id*, *name*, *age*, *type_id* (FK połączony z tabelką **animals_types**). Stwórz klasę **Animal**, która będzie zawierała wszystkie pola jakie ma tabelka czyli: *id*, *name*, *age*, *type* (pole o typie **AnimalType**). Stwórz klasę **AnimalsDao** i dodaj do niej metody, które pozwolą:
 - a. Dodać nowe zwierzę - argument: **Animal**
 - b. Pobrać z bazy wszystkie zwierzęta - lista obiektów **Animal**
 - c. Usunąć zwierzę o podanym id
 - d. Zmienić dane zwierzęcia o podanym id - argument: **Animal**
5. Stwórz klasę **ZooKeeper** i w metodzie *main()* dodaj kilka typów zwierząt, następnie dodaj kilka zwierząt, następnie pobierz i wyświetl wszystkie zwierzęta, usuń pierwsze zwierzę z listy i zmień nazwę dla zwierzęcia ostatniego na liście

6. **(dla chętnych)** Zmień metodę w klasie ***AnimalsTypesDao***, która pobiera rodzaje zwierząt tak żeby zwracała również ilość zwierząt przy każdym rodzaju lub 0 jeżeli nie ma zwierzęcia dla danego typu. Stwórz w tym celu nową klasę: ***AnimalTypeStatistics***, która będzie zawierała dwa pola: **typ(*AnimalType*)** i **ilość(int)**
7. **(dla chętnych)** Zmień metodę w klasie ***AnimalsDao***, która pobiera zwierząt tak żeby możliwe było filtrowanie po konkretnym typie zwierzęcia (np. pobierz wszystkie 'ptaki')