

Hibernate, JPA – laboratorium

Instrukcja zawiera dwie części

Część I – mocno „przewodnikowa” – to jest to co powinno udać się zrealizować w ramach zajęć.

Część II – (od punktu II włącznie) to co stanowi przedmiot zadania domowego.

Na koniec zajęć proszę o umieszczenie w moodlu screenshot’a pokazujący to co udało się zrealizować

I. Basics

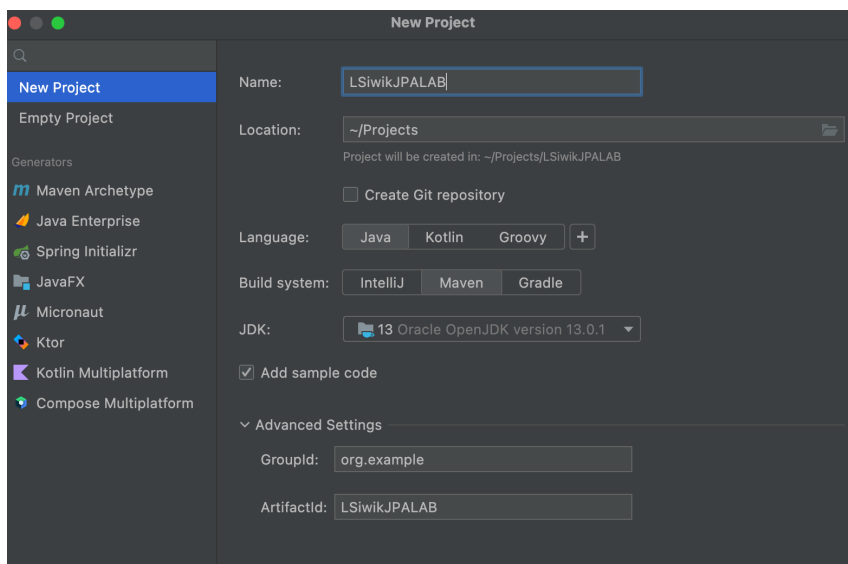
- a. Ściągnij i rozpakuj serwer bazodanowy Apache Derby (pobierz wersję stosownie do wersji Javy którą masz na swoim komputerze
https://db.apache.org/derby/derby_downloads.html

- b. Uruchom serwer Derby (skrypt startNetworkServer z podkatalogu bin ściągniętej paczki). Powinieneś uzyskać efekt podobny do poniższego:

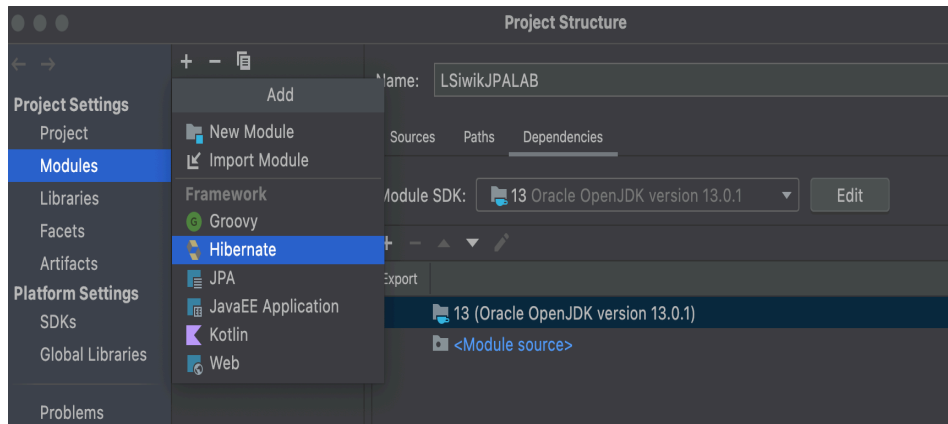
```
-rw-r--r-- 1 macbookpro staff 1389 6 sty 2019 sysinfo.bat
(base) MacBook-Pro-macbook:bin macbookpro$ ./startNetworkServer
Wed Apr 28 21:17:11 CEST 2021 : Security manager installed using the Basic server security policy.
Wed Apr 28 21:17:22 CEST 2021 : Serwer sieciowy Apache Derby - 10.15.2.0 - (1873585) uruchomiony i gotowy do zaakceptowania połączeń na porcie 1527 w {3}
```

- c. Wędrujemy do IntelliJ’a. Jeśli masz wersję Ultimate, możesz skorzystać z wbudowanych wizerdów i wtedy powinny zadziałać kroki i-xii poniżej

- i. Tworzymy nowy projekt „Jawowo – Mavenowy”,

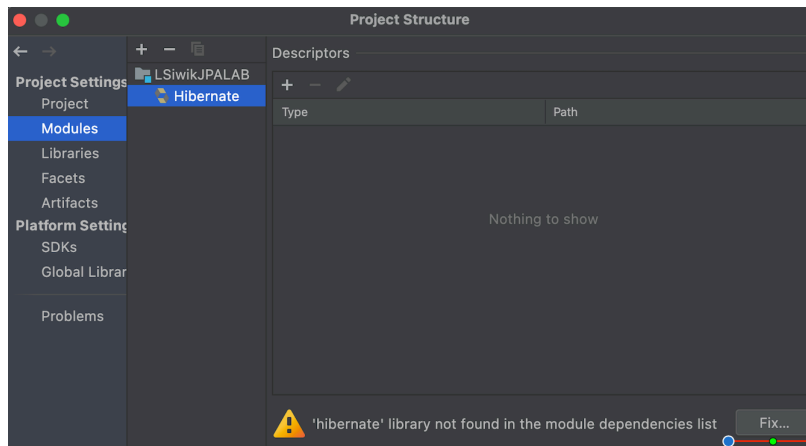


- ii. Po utworzeniu projektu wędrujemy do okna zarządzania zależnościami (File->Project Structure). Do modułów dodajemy Hibernate’a



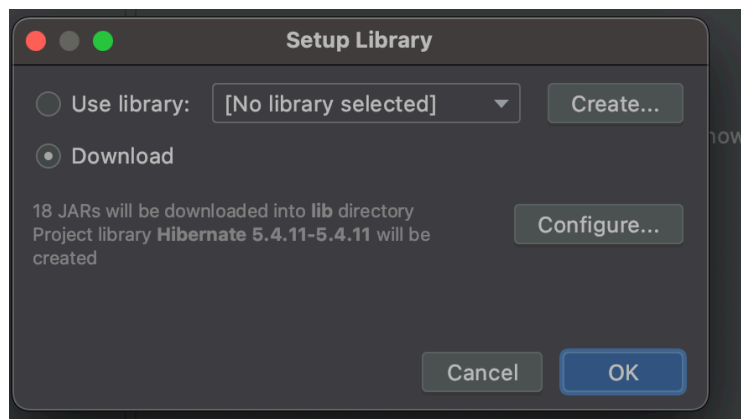
iii.

iv. Następnie fiksujemy problem braku hibernate na liście zależności

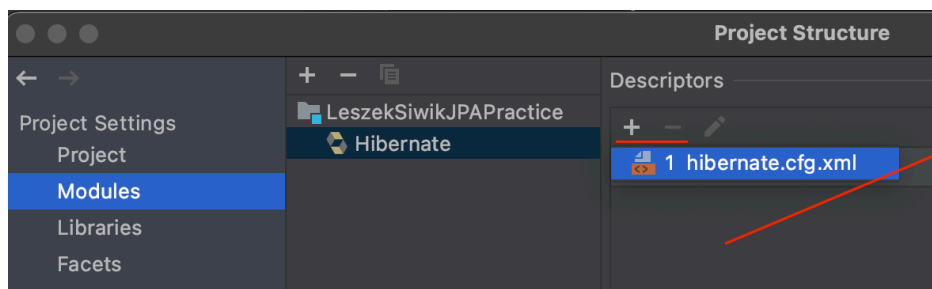


v.

vi. I dociągamy hibernate'a do projektu

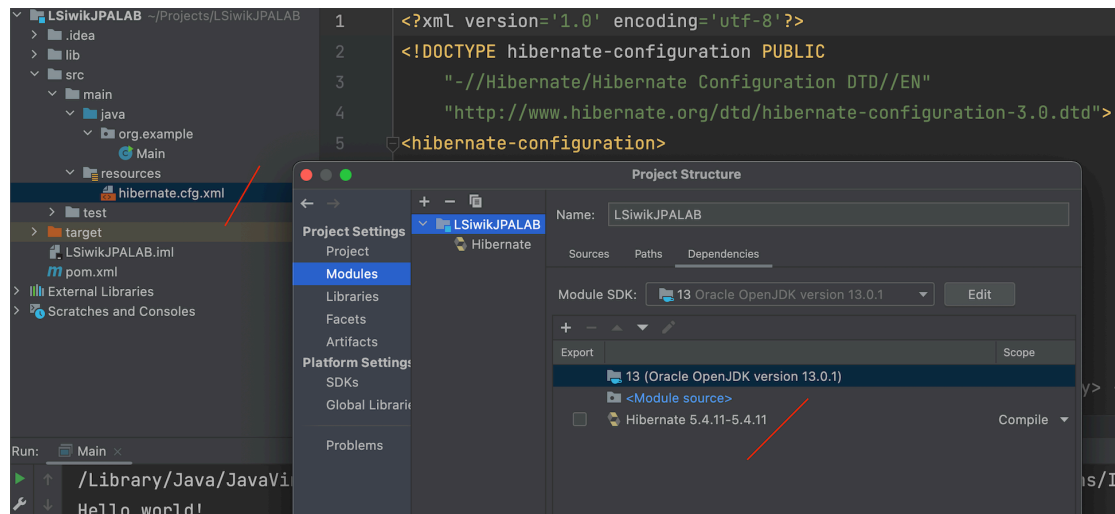


vii. Od razu możemy wygenerować sobie plik konfiguracyjny



viii.

- ix. Po dotychczasowych krokach sytuacja powinna być taka, że w zależnościach projektu mamy Hibernate'a, i dodatkowo w źródłach mamy wygenerowany plik konfiguracyjny

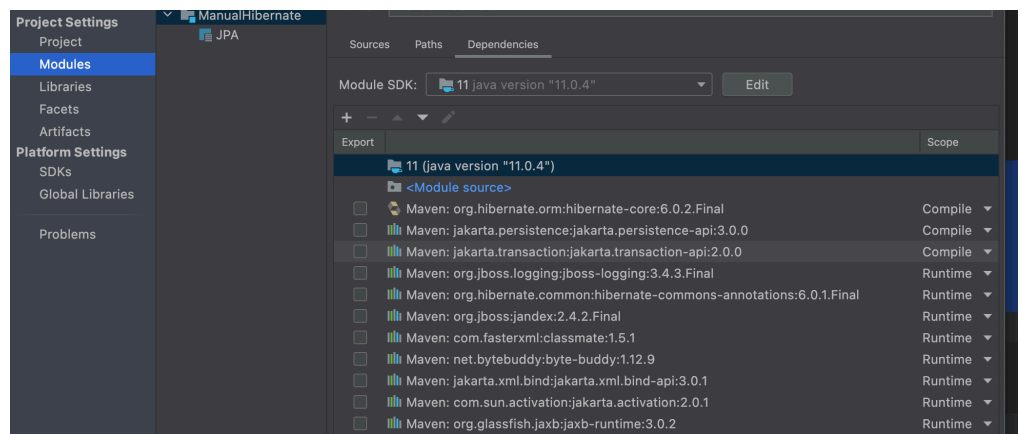


x.

- d. Jeśli masz wersję Community to dwie opcje: upgrade to wersji ultimate i wracamy do kroku poprzedniego, lub dociągamy/konfigurujemy projekt „ręcznie”. Możemy np. stworzyć projekt mavenowy, i żeby dociągnąć/dołączyć do projektu Hibernate w pliku konfiguracyjnym Mavena (pom.xml) dodajemy następującą zależność:

```
II. <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>6.0.2.Final</version>
</dependency>
```

- i. Po zsynchronizowaniu/zbudowaniu, powinniśmy w zależnościach projektu zobaczyć m.in. to na czym nam zależało czyli Hibernate'a



ii.

- b.
- c. Dołącz do projektu (File→Project Structure → Modules→ Dependencies) Jar-ki związane z obsługą/komunikacją z Derby (derby.jar, derbyclient.jar, derbynet.jar, derbytools.jar). Znajdziesz je w podkatalogu lib ściągniętego Apache Derby.

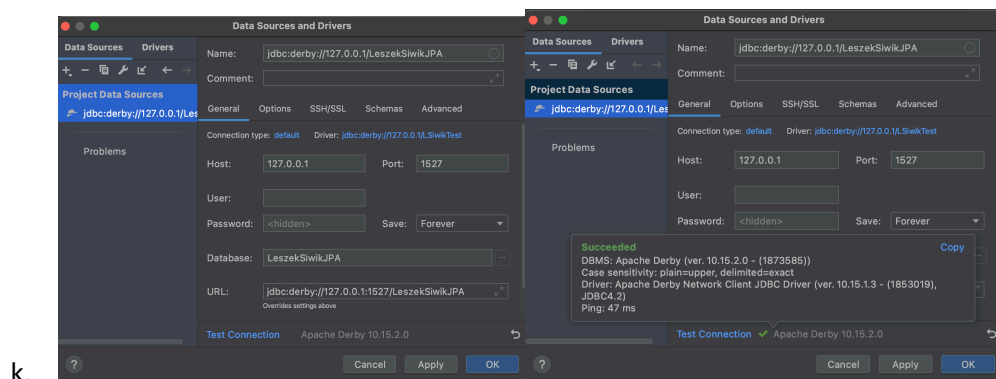
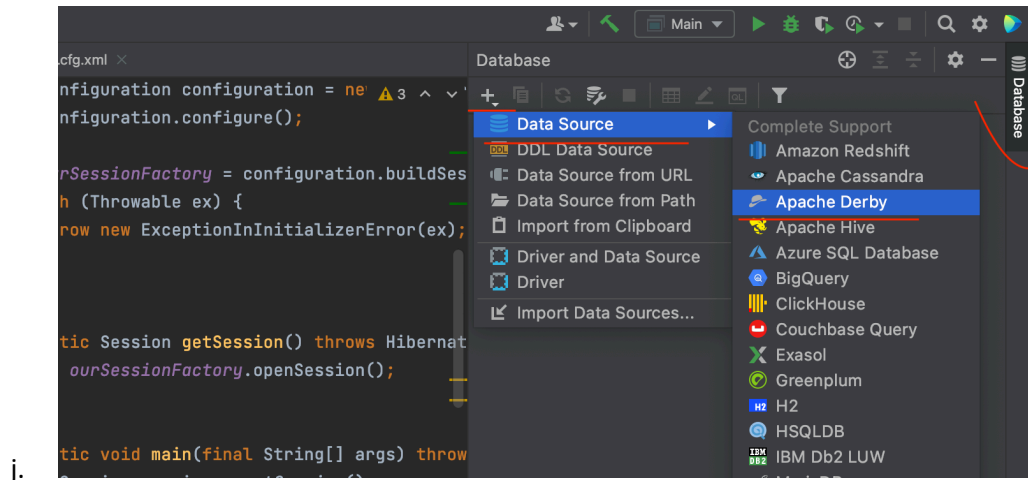
- d. Uzupełnij wpisy w hibernate.cfg.xml podając driver, connection_url, dodaj opcje show_sql oraz format_sql, a także hbm2ddl na update. Przed pierwszym uruchomieniem connection_url powinien wyglądać jak poniżej jdbc:derby://127.0.0.1/INazwiskoJPA;create=true; (z dokładnością do nazwy bazy danych). Po pierwszym uruchomieniu dopisek create=true może zostać usunięty. Reasumując config powinien wyglądać podobnie do poniższego:

e.

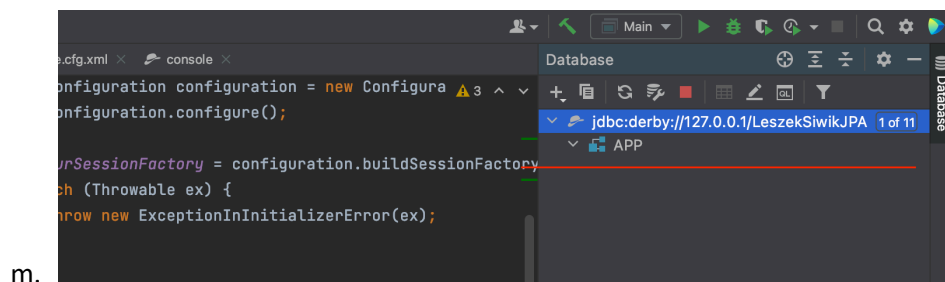
- f. Uzupełnij klasę main zgodnie ze wzorcem dostępnym na platformie UPEL
- g. Uruchom projekt, Na razie nie będzie się wiele działo ale na konsoli powinieneś zobaczyć wpisy hibernate, bez żadnych wyjątków, czyli mniej więcej stan jak poniżej:

h.

- i. a na serwerze powinna się założyć baza o zdefiniowanej w konfigu nazwie. Możesz to zweryfikować podpinając się np. z poziomu IntelliJ do uruchomionego serwera Derby do bazy o podanej nazwie i to połączenie powinno się powieść.



- l. Po podpięciu do bazy na serwerze powinien być widoczny schemat APP – na razie pusty bo nie dodawaliśmy tam żadnych tabel – czyli stan jak poniżej



l. Praca z modelem

- Stwórz klasę produktu z polami `ProductName`, `UnitsOnStock`
- Uzupełnij definicję klasy o elementy niezbędne do jej zmapowania do bazy danych przez Hibernate (adnotacja `@Entity`, nominowanie pola ID, pusty konstruktor)
- Rozszerzamy maina o stworzenie nowego produktu i zapisanie go w bazie danych z wykorzystaniem hibernate'a. Uruchamiamy i testujemy projekt. Efekty powinny być następujące:

d.

```

final Session session = getSession();
Product product = new Product("Krzyszto", unitsOnStock: 111);
try {
    Transaction tx = session.beginTransaction();
    session.save(product);
    tx.commit();
} catch (Exception e) {
    // ...
}

```

```

INFO: HHH18081581: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInt
Hibernate:
create table Product (
  ProductID integer not null,
  ProductName varchar(255),
  UnitsOnStock integer,
  primary key (ProductID)
)
Hibernate: create sequence hibernate_sequence start with 1 increment by 1
kwi 30, 2021 10:51:05 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH080490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate:
values
  next value for hibernate_sequence
Hibernate:
insert
into
  Product
(ProductName, UnitsOnStock, ProductID)
values
  (?, ?, ?)

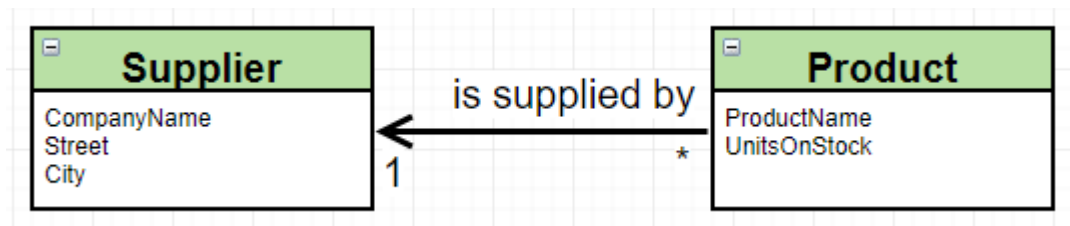
```

e.

PRODUCTID	PRODUCTNAME	UNITSONSTOCK
1	Krzyszto	111

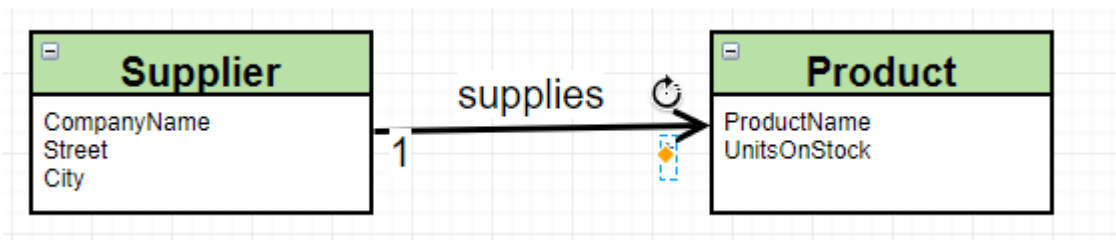
f. Udokumentuj wykonane kroki oraz uzyskany rezultat (logi wywołań słowowych, describe table/schemat bazy, select * from....)

II. Zmodyfikuj model wprowadzając pojęcie Dostawcy jak poniżej



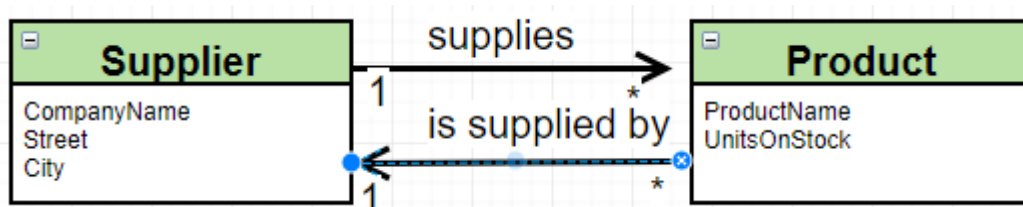
- Stworz nowego dostawce.
- Znajdz poprzednio wprowadzony produkt i ustaw jego dostawce na właśnie dodanego.
- Udokumentuj wykonane kroki oraz uzyskany rezultat (ogi wywołań słowowych, describe table/schemat bazy danych, select * from....)

III. Odwróć relacje zgodnie z poniższym schematem



- Zamodeluj powyższe w dwóch wariantach „z” i „bez” tabeli łącznikowej
- Stworz kilka produktów
- Dodaj je do produktów dostarczanych przez nowo stworzonego dostawcę
- Udokumentuj wykonane kroki oraz uzyskane rezultaty w obu wariantach (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**

IV. Zamodeluj relację dwustronną jak poniżej:

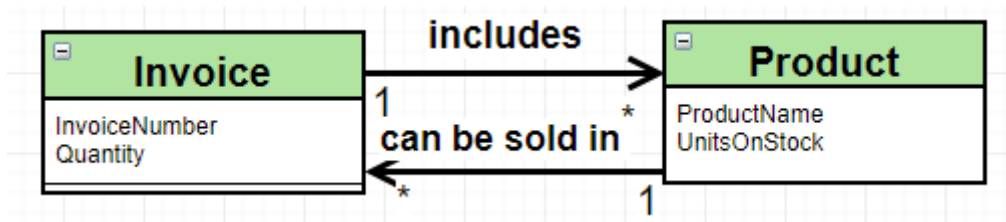


- Tradycyjnie: Stworz kilka produktów
- Dodaj je do produktów dostarczanych przez nowo stworzonego dostawcę (pamiętaj o poprawnej obsłudze dwustronności relacji)
- Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**

V. Dodaj klasę Category z property int CategoryID, String Name oraz listą produktów List<Product> Products

- Zmodyfikuj produkty dodając wskazanie na kategorii do której należy.
- Stworz kilka produktów i kilka kategorii
- Dodaj kilka produktów do wybranej kategorii
- Wydobądź produkty z wybranej kategorii oraz kategorię do której należy wybrany produkt
- Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**

VI. Zamodeluj relacje wiele-do-wielu, jak poniżej:



- a. Stórz kilka produktów i “sprzedaj” je na kilku transakcjach.
- b. Pokaż produkty sprzedane w ramach wybranej faktury/transakcji
- c. Pokaż faktury w ramach których był sprzedany wybrany produkt
- d. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**

VII. JPA

- a. Stwórz nowego maina w którym zrobisz to samo co w poprzednim ale z wykorzystaniem JPA
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**

VIII. Kaskady

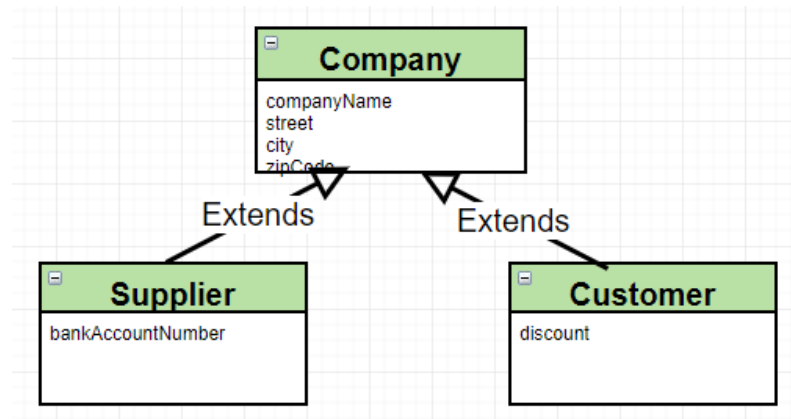
- a. Zmodyfikuj model w taki sposób aby było możliwe kaskadowe tworzenie faktur wraz z nowymi produktami, oraz produktów wraz z nową fakturą
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**

IX. Embedded class

- a. Dodaj do modelu klasę adres. „Wbuduj” ją do tabeli Dostawców.
- b. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat, select * from....)**
- c. Zmodyfikuj model w taki sposób, że dane adresowe znajdują się w klasie dostawców. Zmapuj to do dwóch osobnych tabel.
- d. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**

X. Dziedziczenie

- a. Wprowadź do modelu następującą hierarchię:



- b. Dodaj i pobierz z bazy kilka firm obu rodzajów stosując po kolei trzy różne strategie mapowania dziedziczenia.
- c. **Udokumentuj wykonane kroki oraz uzyskane rezultaty (logi wywołań sqlowych, describe table/schemat bazy danych, select * from....)**