

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**ЛАБОРАТОРНА РОБОТА № 15**

**з кредитного модуля «Проектування корпоративних інформаційних систем» на тему:**

«Створення проектів з відображенням ієрархії класів в таблиці БД»

Виконав студент групи КП-61с

Литвин Вадим Олександрович

Київ 2017

**Мета роботи**: Створити і налаштувати JPA-проект з ієрархією класів і відобразити їх у нову БД.

Тема «ІС хлібозаводу».

Вихідний код проекту: <https://github.com/VadyaVL/CIS-Bakery>

**Хід роботи**

Для виконання поточної лабораторної роботи використовуємо ієрархію класів з попередньої, тринадцятої, лабораторної роботи. В структуру проекту було додано незначні зміни. В пакет ua.bakery.logic було додано новий клас Lab15.class в якому відбувалася реалізація поставленого в поточній лабораторній роботі завдання. Було змінено persistence.xml.

Використовувана СУБД в даній роботі – MySQL.

На рис. 1 відображено діаграму класів пакету ua.bakery.db.jpa.

Демонстрація роботи програми (вивід M-to-M прайс-листа):

Місто: Рокитне, центр

Батон 7,98 UAH

Чорний хліб 7,34 UAH

Пампушки 2,44 UAH

Місто: Біла Церква

Батон 12,41 UAH

Чорний хліб 6,88 UAH

Пампушки 7,68 UAH

Місто: Київ

Батон 3,27 UAH

Чорний хліб 12,17 UAH

Пампушки 13,68 UAH

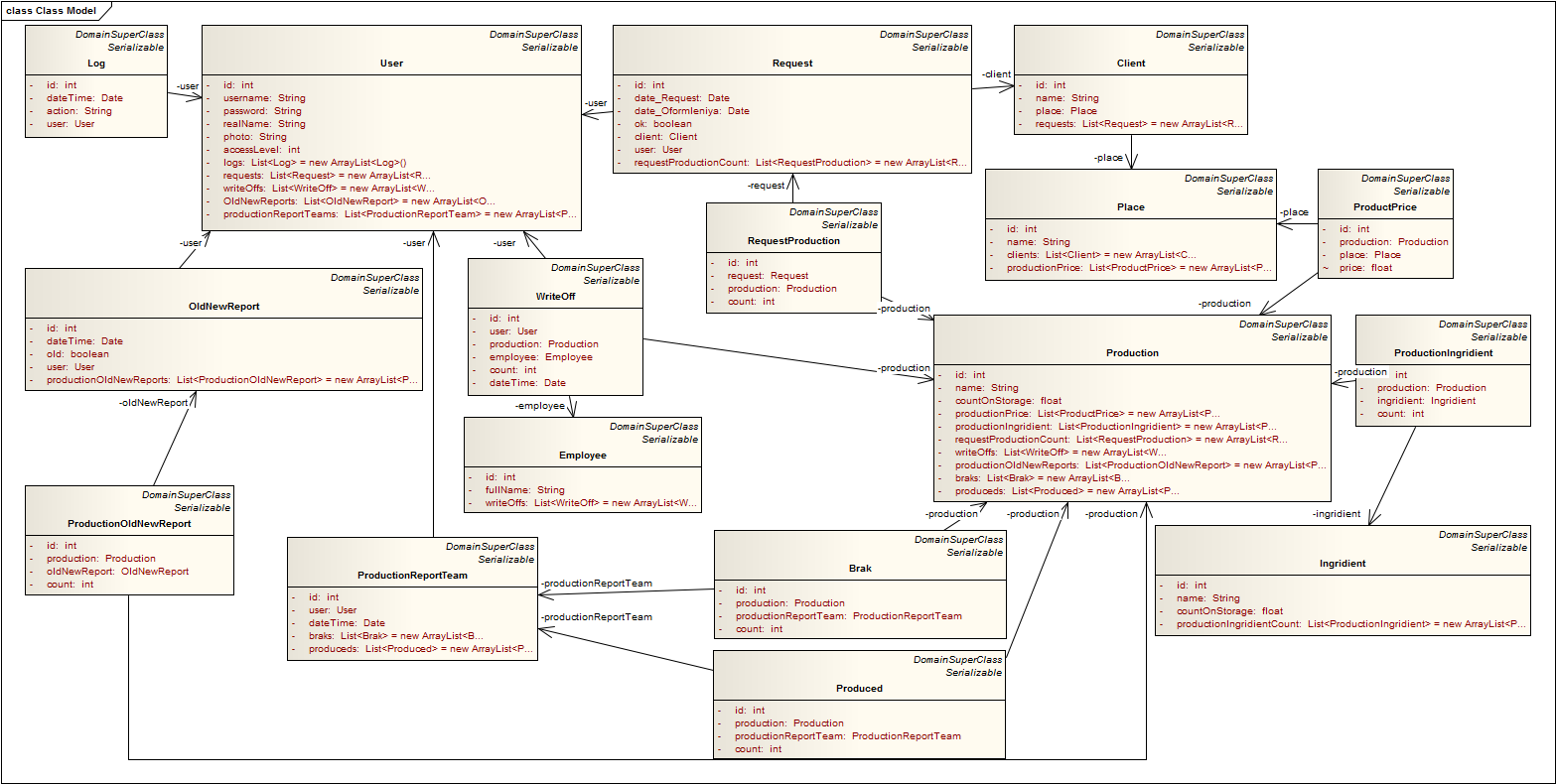


Рис. 1 Діаграма класів пакету ua.bakery.db.jpa

Для реалізації поставленого завдання створимо нову пусту БД MySQL (див. рис. 2). Як видно з малюнку вона не містить в собі жодної таблиці.

|  |
| --- |
| C:\Users\Vadym\Desktop\pic2.PNG |

Рис. 2 Нова порожня БД bakery\_from\_classes

Далій додамо певні зміни до файлу persistence.xml. Лістинг даного файлу можна побачити в лістингу 1.

|  |
| --- |
| Лістинг 1. Файл persistence.xml |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <persistence version=*"2.1"* xmlns=*"http://xmlns.jcp.org/xml/ns/persistence"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence\_2\_1.xsd"*>  <persistence-unit name=*"CISBakeryJPA"* transaction-type=*"RESOURCE\_LOCAL"*>  <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>  <!--  <class>ua.bakery.db.jpa.DomainSuperClass</class>  -->  <class>ua.bakery.db.jpa.User</class>  <class>ua.bakery.db.jpa.Log</class>  <class>ua.bakery.db.jpa.Client</class>  <class>ua.bakery.db.jpa.Place</class>  <class>ua.bakery.db.jpa.Employee</class>  <class>ua.bakery.db.jpa.Production</class>  <class>ua.bakery.db.jpa.Request</class>  <class>ua.bakery.db.jpa.Ingridient</class>  <class>ua.bakery.db.jpa.OldNewReport</class>  <class>ua.bakery.db.jpa.ProductionReportTeam</class>      <class>ua.bakery.db.jpa.ProductPrice</class>  <class>ua.bakery.db.jpa.ProductionIngridient</class>  <class>ua.bakery.db.jpa.RequestProduction</class>  <class>ua.bakery.db.jpa.WriteOff</class>  <class>ua.bakery.db.jpa.ProductionOldNewReport</class>  <class>ua.bakery.db.jpa.Brak</class>  <class>ua.bakery.db.jpa.Produced</class>    <properties>    <property name=*"javax.persistence.schema-generation.database.action"* value=*"drop-and-create"*/>  <property name=*"javax.persistence.schema-generation.scripts.action"* value=*"drop-and-create"*/>  <property name=*"javax.persistence.schema-generation.scripts.create-target"* value=*"sampleCreate.ddl"*/>  <property name=*"javax.persistence.schema-generation.scripts.drop-target"* value=*"sampleDrop.ddl"*/>    <property name=*"javax.persistence.sql-load-script-source"* value=*"insert.sql"*/>    <property name=*"javax.persistence.jdbc.driver"* value=*"com.mysql.jdbc.Driver"*/>  <property name=*"javax.persistence.jdbc.url"* value=*"jdbc:mysql://localhost:3306/bakery\_from\_classes"*/>  <property name=*"javax.persistence.jdbc.user"* value=*"root"*/>  <property name=*"javax.persistence.jdbc.password"* value=*"1111"*/>  </properties>  </persistence-unit>  </persistence> |

До файлу було додано нові властивості (properties), які дозволять згенерувати нову БД. Дізнатися про кожну нову властивість можна за посиланням: [Generating Database Schemas with JPA 2.1](https://antoniogoncalves.org/2014/12/11/generating-database-schemas-with-jpa-2-1/).

Нижче наведемо лістинг деяких інших файлів з проекту. Повний код лабораторної роботи можна побачити за посилання вказаним на початку даного документу.

|  |
| --- |
| Лістинг 2. Клас Lab15.java |
| **package** ua.bakery.logic;  **import** java.util.ArrayList;  **import** java.util.Date;  **import** java.util.List;  **import** java.util.Random;  **import** javax.persistence.Persistence;  **import** ua.bakery.db.DAO.IGenericDAO;  **import** ua.bakery.db.DAO.impl.GenericDAOImpl;  **import** ua.bakery.db.jpa.Place;  **import** ua.bakery.db.jpa.ProductPrice;  **public** **class** Lab15 {  **private** **static** **final** String ***PERSISTENCE\_UNIT\_NAME*** = "CISBakeryJPA";  **public** **static** **void** main(String[] args) {  // Згенерувати БД з існуючих класів JPA  /\*  try{  System.out.println((new Date()) + ": початок генерації БД!");  Persistence.generateSchema(PERSISTENCE\_UNIT\_NAME, null);  System.out.println((new Date()) + ": БД згенеровано успышно!");  }  catch(Exception ex){  System.err.println((new Date()) + ": Помилка: " + ex.getMessage());  }  \*/    /\*  List<String> places = new ArrayList<>();  places.add("Рокитне, центр");  places.add("Біла Церква");  places.add("Київ");    IGenericDAO<Place> genericDAO = new GenericDAOImpl<Place>(Place.class,  Persistence.createEntityManagerFactory("CISBakeryJPA"));    for(String strPlace : places) {  Place currentPlace = new Place();  currentPlace.setName(strPlace);  genericDAO.save(currentPlace);  }\*/  /\*  List<String> products = new ArrayList<>();  products.add("Батон");  products.add("Чорний хліб");  products.add("Пампушки");    IGenericDAO<Production> genericDAO = new GenericDAOImpl<Production>(Production.class,  Persistence.createEntityManagerFactory("CISBakeryJPA"));    for(String strProds : products) {  Production currentProduction = new Production();  currentProduction.setName(strProds);  genericDAO.save(currentProduction);  }  \*/  /\*  Random rand = new Random();  IGenericDAO<Place> placeDAO = new GenericDAOImpl<Place>(Place.class,  Persistence.createEntityManagerFactory("CISBakeryJPA"));  IGenericDAO<Production> productionDAO = new GenericDAOImpl<Production>(Production.class,  Persistence.createEntityManagerFactory("CISBakeryJPA"));    IGenericDAO<ProductPrice> productPriceDAO = new GenericDAOImpl<ProductPrice>(ProductPrice.class,  Persistence.createEntityManagerFactory("CISBakeryJPA"));  List<Place> places = (List<Place>) placeDAO.getAll();  List<Production> produs = (List<Production>) productionDAO.getAll();  System.out.println(places.size());  System.out.println(produs.size());    for(Place pl : places){  for(Production pr : produs){  ProductPrice pp = new ProductPrice();  pp.setPlace(pl);  pp.setProduction(pr);  pp.setPrice(rand.nextFloat()\*15.0f);  productPriceDAO.save(pp);  }  }  \*/  IGenericDAO<Place> placeDAO = **new** GenericDAOImpl<Place>(Place.**class**,  Persistence.*createEntityManagerFactory*("CISBakeryJPA"));  System.***out***.println("Прайс-лист по містам:");    **for**(Place place : placeDAO.getAll()){  System.***out***.println("\nМісто: " + place.getName());    **if**(place.getProductionPrice().size() == 0 )  System.***out***.println("Дані відсутні");    **for**(ProductPrice pp : place.getProductionPrice()){  System.***out***.format("%-25s %25.2f UAH\n",  pp.getProduction().getName(),  pp.getPrice());  }  }  }  } |

В лістингу присутні чотири блоки багаторядкових коментарів:

1. Створення нової БД одним рядком: Persistence.generateSchema(PERSISTENCE\_UNIT\_NAME, null);
2. Додавання міст/адрес;
3. Додавання продукції;
4. Додавання цін;
5. Та не за коментований блок з виводом прайс-листа (див. вивід програми);

Після виконання даного коду ми отримуємо нову БД (див. рис. 3), аналогічну тим, що були в попередніх лабораторних роботах та наявні в ній записи (див. рис. 3).

|  |
| --- |
| C:\Users\Vadym\Desktop\pic3.PNG |

Рис. 3 Нова БД bakery\_from\_classes

|  |
| --- |
|  |

Рис. 4 Записи місця/адреси та продукція в БД bakery\_from\_classes