

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**ЛАБОРАТОРНА РОБОТА № 11**

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

«Маніпулювання даними в проектах з JPA»

Виконав студент групи КП-61с

Литвин Вадим Олександрович

Київ 2017

**Мета роботи**: Виконання додавання, видалення і оновлення сутностей в JPA-проекті.

Тема «ІС хлібозаводу».

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

**Хід роботи**

Для виконання поточної лабораторної роботи використовуємо базу даних з попередньої, десятої, лабораторної роботи. В структуру проекту було додано незначні зміни. В пакет ua.bakery.logic було додано новий клас Lab11.class в якому відбувалася реалізація поставленого в поточній лабораторній роботі завдання. В деяких класах було перевизначено метод toString().

|  |
| --- |
|  |

Рис. 1 Структура jdbc та JPA проекту «ІС хлібозаводу»

Використовувана СУБД в даній роботі – MySQL.

На рис. 1 відображено структуру проекту даної лабораторної роботи.

Демонстрація роботи програми (Додавання нової заявки, її редагування та видалення):

[EL Info]: 2017-04-01 18:51:39.157--ServerSession(1014508942)--EclipseLink, version: Eclipse Persistence Services - 2.5.0.v20130507-3faac2b

[EL Info]: connection: 2017-04-01 18:51:39.874--ServerSession(1014508942)--file:/G:/projecs/CIS-Bakery/bin/\_CISBakeryJPA login successful

Заявка: Заявка №50: Wed Aug 24 18:14:06 EEST 2016: виконано

----class ua.bakery.db.jpa.RequestProduction----

Назва | К-сть

Хліб Пшеничний (ФОРМОВИЙ) 0,5 : 6

Хліб арнаут КиІвський 0,6 : 6

Хліб УкраІнський (формовий) 1,00 : 4

ХЛІБ Колосок 0,8 : 5

Батон Дорожній в/г 0,45 : 6

Батон Дорожній в/г 0,45 різ. : 10

Пампушка 0,55 : 3

Плетінка здобна 0,45 : 3

Калач в/г 0,4 : 3

Ріжок з повидлом в/г 0,4 : 3

Булка Сімейка в/г 0,4 : 5

Завиток згущонка в/г 0,4 : 4

Пироги з маком в/г 0,5 : 6

Пиріг творог-абрикос в/г 0,4 : 6

Булочка "Бородянська" 0,1 : 5

Пплюшка Московська 0,1 : 5

Булочка Літо 0,1 : 5

--------------------------------------

----class ua.bakery.db.jpa.User----

vadya

Уманець Богдан Борисович

Чередніченко Галина

Гірчева Людмила

Рахманова Ольга

TestJPA TestJPA

--------------------------------------

----СТВОРЕННЯ ЗАМОВЛЕННЯ----

Користувач: Уманець Богдан Борисович

Клієнт : Шевченко

Дата форм.: Sat Apr 01 18:51:40 EEST 2017

Замовлення:

Хліб Пшеничний (ФОРМОВИЙ) 0,5 : 85

Хліб Пшеничний (ФОРМОВИЙ) 0,5 різ. : 11

Хліб Пшеничний (ФОРМОВИЙ) 0,9 : 52

Хліб Пшеничний (ФОРМОВИЙ) 0,8 : 26

Хліб арнаут КиІвський 0,6 : 2

Хліб арнаут 0,6 : 12

Хліб арнаут 0,6 різаний : 41

Хліб 0,6 НОВИЙ : 12

Хліб Кукурудзяний : 19

Хліб Рівенський (круглий) 0,65 : 85

Хліб Рівенський (з тміном) 0,65 : 10

----СТВОРЕНО----

----РЕДАГУВАННЯ----

Заявка №530: Sat Apr 01 18:51:40 EEST 2017: не виконано

Замовлення:

Хліб Пшеничний (ФОРМОВИЙ) 0,5 : 170

Хліб Пшеничний (ФОРМОВИЙ) 0,5 різ. : 22

Хліб Пшеничний (ФОРМОВИЙ) 0,9 : 104

Хліб Пшеничний (ФОРМОВИЙ) 0,8 : 52

Хліб арнаут КиІвський 0,6 : 4

Хліб арнаут 0,6 : 24

Хліб арнаут 0,6 різаний : 82

Хліб 0,6 НОВИЙ : 24

Хліб Кукурудзяний : 38

Хліб Рівенський (круглий) 0,65 : 170

Хліб Рівенський (з тміном) 0,65 : 20

----ВІДРЕДАГОВАНО----

----ВИДАЛЕННЯ----

Заявка №530: Sat Apr 01 18:51:40 EEST 2017: не виконано

----ВИДАЛЕНО----

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

|  |
| --- |
| Лістинг 1. Клас Lab11.java |
| **package** ua.bakery.logic;  **import** java.util.ArrayList;  **import** java.util.Date;  **import** java.util.List;  **import** java.util.Random;  **import** javax.persistence.EntityManager;  **import** javax.persistence.EntityManagerFactory;  **import** javax.persistence.Persistence;  **import** javax.persistence.Query;  **import** ua.bakery.db.jpa.Client;  **import** ua.bakery.db.jpa.Production;  **import** ua.bakery.db.jpa.Request;  **import** ua.bakery.db.jpa.RequestProduction;  **import** ua.bakery.db.jpa.User;  /\*\*  \* Клас для реалізації завдання з лабораторної роботи №11  \* **@author** Vadym  \*  \*/  **public** **class** Lab11 {  **private** **static** **final** String ***PERSISTENCE\_UNIT\_NAME*** = "CISBakeryJPA";  **private** **static** EntityManagerFactory *factory*;    **public** **static** **void** main(String[] args) {  // Фабричний метод. Отримуємо об'єкт EntityManagerFactory  *factory* = Persistence.*createEntityManagerFactory*(***PERSISTENCE\_UNIT\_NAME***);  *readData*();  **int** idRequest = *writeData*();  *editData*(idRequest);  *removeData*(idRequest);  }    /\*\*  \* Вибірка даних  \*/  **public** **static** **void** readData(){  EntityManager em = *factory*.createEntityManager();    // Читання замовлення і продукції в замовленні  Request request = (Request) em.find(Request.**class**, 50);  System.***out***.println("Заявка: " + request.toString());  *showList*(request.getRequestProductionCount());    // Читання усіх користувачів  Query q = em.createQuery("select user from User user");  @SuppressWarnings("unchecked")  List<User> users = q.getResultList();  *showList*(users);  }    /\*\*  \* Запис даних до БД  \*/  **public** **static** **int** writeData(){  System.***out***.println("----СТВОРЕННЯ ЗАМОВЛЕННЯ----");    Random random = **new** Random(); // Для генераціїї випадкових кількостей продукції  EntityManager em = *factory*.createEntityManager();    // Отримаємо існуючого користувача і клієнта  User user = (User) em.find(User.**class**, 2);  Client client = (Client) em.find(Client.**class**, 4);  System.***out***.println("Користувач: " + user.toString());  System.***out***.println("Клієнт : " + client.toString());  System.***out***.println("Дата форм.: " + **new** Date());    // Отримаємо всі продукти  Query q = em.createQuery("select product from Production product");  @SuppressWarnings("unchecked")  List<Production> productions = q.getResultList();  // Створимо замовлення  Request request = **new** Request();  request.setDate\_Request(**new** Date()); // Встановити дату виконання  request.setDate\_Oformleniya(**new** Date()); // Встановити дату формування  request.setClient(client);  request.setUser(user);  request.setOk(**false**);    // Сформуємо список необхідних продуктів    // Штучно обмежимо максимальну кількість продукції в замовленні - ЛИШЕ ДЛЯ ЗРУЧНОЇ ДЕМОНСТРАЦІЇ  **int** i = 0, MAX = 10;    System.***out***.println("Замовлення: ");  List<RequestProduction> requestProductions = **new** ArrayList<>();  **for**(Production production : productions){    **int** count = random.nextInt(100);    **if**(count != 0){  RequestProduction rp = **new** RequestProduction();  rp.setProduction(production);  rp.setRequest(request);  rp.setCount(count);  requestProductions.add(rp);  System.***out***.format("%-35s: %d%n", production.getName(), count);    **if**(++i>MAX) **break**;  }  }    request.setRequestProductionCount(requestProductions);    // Збережемо до БД  em.getTransaction().begin();  em.persist(request);  em.getTransaction().commit();  em.close();  System.***out***.println("----СТВОРЕНО----");    **return** request.getId();  }    /\*\*  \* Редагування даних  \*/  **public** **static** **void** editData(**int** requestId){  System.***out***.println("----РЕДАГУВАННЯ----");  EntityManager em = *factory*.createEntityManager();    Request request = (Request) em.find(Request.**class**, requestId); // Отримаємо замовлення  System.***out***.println(request);    // Змінимо замовлення  System.***out***.println("Замовлення: ");  em.getTransaction().begin();  **for**(RequestProduction requestProduction : request.getRequestProductionCount()){  requestProduction.setCount(requestProduction.getCount() \* 2);  System.***out***.format("%-35s: %d%n", requestProduction.getProduction().getName(), requestProduction.getCount());  }  em.getTransaction().commit();  em.close();  System.***out***.println("----ВІДРЕДАГОВАНО----");  }    /\*\*  \* Видалення даних  \* **@param** requestId  \*/  **public** **static** **void** removeData(**int** requestId){  System.***out***.println("----ВИДАЛЕННЯ----");  EntityManager em = *factory*.createEntityManager();    Request request = (Request) em.find(Request.**class**, requestId); // Отримаємо замовлення  System.***out***.println(request);    // Видалемо замовлення  em.getTransaction().begin();  em.remove(request);  em.getTransaction().commit();  em.close();  System.***out***.println("----ВИДАЛЕНО----");  }    /\*\*  \* Вивід списку  \* **@param** list  \*/  **public** **static** **void** showList(List<?> list){    **if**(list.size() == 0){  System.***out***.println("Список порожній!");  **return**;  }    System.***out***.println("----" + list.get(0).getClass() + "----");  **if**(list.get(0) **instanceof** RequestProduction){  System.***out***.format("%-35s| %s%n", "Назва ", "К-сть");  **for**(Object ent: list){  System.***out***.format("%-35s: %d%n", ((RequestProduction)ent).getProduction(), ((RequestProduction)ent).getCount());  }  }  **else**{  **for**(Object ent: list){  System.***out***.println(ent.toString());  }  }    System.***out***.println("--------------------------------------");  }  } |