

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни

«Архітектура і технології Web-служб»

для студентів спеціальності 123 «Комп'ютерна інженерія»

всіх форм навчання

Частина 3

WEB-СЛУЖБИ ДЛЯ РОБОТИ З СКБД. JDBC API.

2019

Методичні вказівки до лабораторних робіт з дисципліни «Архітектури і технології Web-служб» для студентів спеціальності 123 «Комп'ютерна інженерія» всіх форм навчання. Частина 3. Web-служби для роботи з СКБД. JDBC API / Укл.: С.Д. Точилін. – Запоріжжя: НУ «Запорізька політехніка», 2019. – 14 с.

Укладачі:

С.Д. Точилін, доцент, к.ф.-м.н.

Рецензент:

І.Я. Зеленьова, доцент, к.т.н.

Відповідальний за випуск:

С.Д. Точилін, доцент, к.ф.-м.н.

Затверджено
на засіданні кафедри
«Комп'ютерних систем та мереж»
Протокол № 1 від 21.08.2019

Рекомендовано до видання:
НМК факультету КНТ
Протокол № 1 від 23.08.2019

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ	4
1.1 JDBC API.....	4
1.2 SQL	5
1.3 Створення баз даних та таблиць із використанням SQL	6
2 РОЗПОДІЛЕНИЙ ДОДАТОК ДЛЯ РОБОТИ З СКБД	7
2.1 Створення Web-сервісу на основі Axis для роботи з MySQL	7
2.2 Клієнт Web-сервісу	9
2.3 Запуск додатка	10
3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ	11
4 КОНТРОЛЬНІ ПИТАННЯ	12
5 ЗМІСТ ПИСЬМОВОГО ЗВІТУ	12
6 РЕКОМЕНДОВАНА ЛІТЕРАТУРА	14

Лабораторна робота №3

«WEB-служба для роботи з СКБД MySQL. JDBC API»

Мета роботи: ознайомитися з специфікацією JDBC. Навчитися створювати Web-сервіси для роботи з даними СКБД.

1. ЗАГАЛЬНІ ВІДОМОСТІ

1.1 JDBC API

Для доступу програм мови Java до реляційних баз даних використовується програмний інтерфейс JDBC API. Офіційно аббревіатура JDBC не розшифровується, але неофіційно вважається, що вона означає Java Database Connectivity.

Програмний інтерфейс JDBC API складається із двох пакетів:

- 1) `java.sql` - стандартно включений в усі версії Java SE;
- 2) `javax.sql` - включений у версії Java SE починаючи з 1.4.0.

Пакет `javax.sql` додатково підтримує корпоративні властивості, наприклад сервіс імен та створення пула з'єднань.

Для роботи з базами даних рекомендується використовувати JDBC-драйвер, що поставляється для конкретної системи керування базами даних (СКБД), наприклад PostgreSQL, MySQL та т.і. У крайньому випадку якщо драйвер JDBC відсутній, можна скористатися драйвером мосту ODBC-JDBC, що входить у стандартний пакет поставки Java. Пакети JDBC API являють собою набір класів та інтерфейсів. Найважливіші з них наведені в табл. 1.

Таблиця 1.1. – Деякі інтерфейси JDBC API

Інтерфейс	Призначення
Connection	Зв'язок з базами даних за допомогою класа DriverManager
Statement PreparedStatement CallableStatement	Виконання SQL- або DDL-тверджень
ResultSet	Повертає після виконання SQL-запиту результуючий набір даних
DatabaseMetaData	Дає інформацію про СКБД та базу даних у цілому
DatabaseResultSetMetaData	Містить інформацію про поля в результуючому наборі даних

1.2 SQL

Мова структурованих запитів SQL (Structured Query Language) - стандартна мова інтерактивної взаємодії з реляційними базами даних.

SQL був розроблений Е.Ф. Коддом в 70-х роках у компанії IBM. SQL має три підмножини з різним типом виконуваних функцій у БД:

- а) мова визначення даних DDL (Data Definition Language);
- б) мова обробки даних DML (Data Manipulation Language);
- в) мова керування даними DCL (Data Control Language).

Мову визначення даних становлять команди для створення та керування структурою бази даних (БД), див. табл. 2.

Таблиця 1.2. Основні команди DDL

Команда	Призначення
create database	Створення БД
create table	Створення таблиці БД
drop <об'єкт>	Видалення зазначеного об'єкта

Мову обробки даних становлять команди, які використовуються для додавання, модифікації, видалення й вибірки даних у БД, див. табл. 3.

Таблиця 1.3. Основні команди DML

Команда	Призначення
select	Витягас дані із БД
insert	Вставляє дані
update	Обновляє дані
delete	Видаляє дані

Мову керування даними становлять команди, які використовуються для обмеження доступу користувача до БД, виконання аудита та керування транзакціями.

Таблиця 1.4. Основні команди DCL

Команда	Призначення
grant	Надання користувачеві певних прав
commit	Фіксація транзакції
rollback	Відкіт транзакції

1.3 Створення баз даних та таблиць із використанням SQL

Алгоритм створення нової бази даних і її таблиць із використанням SQL розглянемо на прикладі створення БД storage (склад). Ця БД буде використовуватися для складського обліку товарів. Вона містить одну таблицю goods (товари), зі стовпцями - id (номер певного товару в БД), name (найменування товару), price (ціна товару), amount (кількість одиниць товару).

Базу даних та таблицю будемо створювати в середовищі СКБД MySQL, з використанням утиліти MySQL Command Line Client.

Основні етапи створення БД та таблиць:

1. Створимо спочатку саму БД за допомогою команди create database:

```
mysql > create database storage;
```

2. Для створення таблиці goods спочатку введемо:

```
mysql > use storage;
```

Потім за допомогою команди create table створимо таблицю:

```
mysql > create table goods (  
    id int,  
    name char(15),  
    price float(5,2),  
    amount int,  
    primary key(id));
```

3. Після цього в таблицю вставляємо дані:

```
mysql > insert into goods values(1,'Apple',5.55,100);  
mysql > insert into goods values(2,'Orange',7.77,50);  
mysql > insert into goods values(3,'Banana',4.44,75);
```

Дані таблиці можуть редагуватися за допомогою команд delete, update. Крім того, таблиця й сама база даних може бути цілком вилучені за допомогою команд:

drop table goods та drop database storage.

Деякі типи даних SQL СКБД MySQL наведені в таблиці 1.5.

Таблиця 1.5. Деякі типи даних SQL СКБД MySQL

Тип	Діапазон	Примітка
int[(M)]	$-2 \cdot 10^{31} \dots 2 \cdot 10^{31} - 1$	Звичайні цілі числа
float[(M,D)]	$\pm 1.175494351 \text{E} - 38 \dots \pm 3 \dots 402823466 \text{E} + 38$	Речовинні числа
double[(M,D)]	(2.225073858507201 E-308... (1.7976931348623157E+308	Речовинні числа
char(M)	від 1 до 255 символів	Рядки фіксованої довжини

М - загальне число символів. D - подання дробової частини. [] - необов'язковість використання елементів при оголошенні.

2 РОЗПОДІЛЕНИЙ ДОДАТОК ДЛЯ РОБОТИ З СКБД

2.1 Створення Web-сервісу на основі Axis для роботи з MySQL

Створимо Web-сервіс PrintDate, який надає одну Web-послугу getDate(), що повертає клієнтові всю інформацію, яка втримується в таблиці goods бази даних storage СКБД MySQL.

Клас PrintDate, що містить метод getDate(), наведений нижче.

Лістинг 1. Web-Служба PrintDate

```
import java.io.*;
import java.sql.*;

public class PrintDate
{
    public String [] getDate()
    {
        try {
            DriverManager.registerDriver(new com.mysql.jdbc.Driver());
```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/storage?user=root&password=root");
        Statement stm = con.createStatement();
        ResultSet rs = stm.executeQuery("select count(*) from goods");
        rs.next();
        int size=4*rs.getInt(1);
        String data [] = new String[size];
        rs = stm.executeQuery("select * from goods");
        int i=0;
        while (rs.next())
        {
            data[i]=rs.getString(1);
            data[i+1]=rs.getString(2);
            data[i+2]=rs.getString(3);
            data[i+3]=rs.getString(4);
            i=i+4;
        }
        rs.close();
        stm.close();
        con.close();
        return data;
    }
    catch(Exception e) {String dt [] = new String[1]; dt [0]="Error";
return dt;}
}
}

```

Назвемо файл із змістом лістингу 1 PrintDate.jws та покладемо його в каталог axis сервера додатків (C:\axis\webapps\axis). Після цього Web-сервіс готов до використання. Не потрібно компілювати клас PrintDate, Axis зробить це сам при першому запиті до Web-служби.

Пояснення коду лістингу 1. Об'єкт *con* типу *Connection* установлює з'єднання з базою даних з використанням класу *DriverManager*. Під час цього з'єднання виконуються всі операції із БД, які записуються за допомогою команд SQL.

При цьому, синтаксис з'єднання до певної БД СКБД MySQL має вигляд:

```
Connection con = DriverManager.getConnection ("jdbc:mysql:  
//localhost/Ім'я_БД?user=Ім'я_користувача&password=Пароль");
```

Для виконання команд SQL варто використовувати об'єкт *Statement*. Одержати цей об'єкт треба за допомогою методу *createStatement()*, об'єкта *Connection*. Інтерфейс *Statement* містить два корисних методи: *executeQuery()* і *executeUpdate()*.

Перший з них використовується для виконання SQL-запитів за допомогою команди вибірки - *select*, другий – команд модифікації даних: *insert*, *delete*, *update*.

Метод *executeQuery()* повертає об'єкт *rs*, що має тип *ResultSet*. *rs* являє собою покажчик на рядок таблиці, що надається методом *executeQuery()* як результат.

Спочатку даний покажчик установлений на одну позицію перед першим рядком у таблиці результатів. Для переміщення покажчика на одну позицію вперед використовують метод *next()* об'єкта *rs*.

Метод *next()* повертає значення *false* у тому випадку коли в таблиці результатів вибірки не залишилося рядків.

Отже, щоб переглянути всі рядки таблиці результатів вибірки можна користуватися циклом *while* с умовою *rs.next()*. Цей цикл завершується тоді коли метод *rs.next()* пройде всі рядки таблиці.

Об'єкт *rs* типу *ResultSet* також містить методи для отримання даних різного формату – *getInt()*, *getFloat()*, *getString()* та т.і.

SQL-твердження *select count(*) from goods* повертає число рядків у таблиці *goods*.

SQL-твердження *select * from goods* витягає всі дані з таблиці *goods*.

2.2 Клієнт Web-сервісу

Далі наведений код клієнта Web-сервісу *PrintDate*

```
import org.apache.axis.client.*;  
import java.net.*;  
  
public class PrintDateClient  
{  
    public static void main(String [] args)
```

```

{
    try{
        Service s = new Service();
        Call call = (Call)s.createCall();
        call.setTargetEndpointAddress(new
URL("http://localhost:8080/PrintDate.jws"));
        call.setOperationName("getDate");
        String [] response =(String [])call.invoke(new Object[]{});
        int d=response.length;
        for(int i=0;i<d;i=i+4)
        {
            System.out.println(response[i]+" "+response[i+1]+" "+response[i+2]+
                                " "+response[i+3]);
        }
    }
    catch(Exception e){ System.out.println(e);}
} }

```

У коді використані інтерфейси Service та Call пакета JAX-RPC. Вони реалізовані класами, що входять в Axis.

Основну роль в одержанні Web-послуги грає об'єкт типу Call. Він містить адресу Web-сервісу та назву Web-послуги *getDate*. У нього заносяться аргументи Web-послуги у вигляді масиву об'єктів.

У нашому прикладі масив аргументів не містить елементів. Нарешті, об'єкт call за допомогою методу *invoke* повертає результат виконання Web-послуги. У нашому випадку це вибірка всіх даних з таблиці *goods*.

2.3 Запуск додатка

У папці з Web-сервісом *PrintDate.jws* запусить *SimpleAxisServer*, виконавши команду:

```
> java org.apache.axis.transport.http.SimpleAxisServer
```

В окремій консолі запусить клієнта Web-сервіса, виконавши команду:

```
> java PrintDateClient
```

У результаті на консоль буде виведено:

1 Apple 5.55 100

2 Orange 7.77 50

3 Banana 4.44 75

3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Завдання 1 – Виконайте додаток описаний в п. 2.

Завдання 2 – Виконайте додаток описаний в п. 2, з використанням утиліти **tcpmon**.

Завдання 3 – Створіть розподілений Java-додаток Web-сервіс якого при запиті клієнта робить вибірку даних таблиці з бази даних storage. Таблиця автоматизує складський облік певних товарів відповідно до Вашого варіанта й повинна містити не менш 5 стовпців та 5 рядків.

№ варіанта	Назва таблиці	Складський облік
1	processor	Процесорів
2	memory	Пам'яті
3	mother_board	Материнських плат
4	case	Корпусів
5	video	Відеокарт
6	audio	Аудіокарт
7	hdd	Жорстких дисків
8	flash_memory	Flash-пам'яті
9	mp3_player	Мр 3-плеєрів
10	dvd_player	Dvd-плеєрів
11	monitor	Моніторів
12	modem	Модемів
13	printer	Принтерів
14	kooler	Кулерів
15	GPS_navigator	GPS-навігаторів
20	commutator	Комутаторів
21	router	Маршрутизаторів
22	netcard	Мережевих карт
23	mouse	Маніпуляторів миша
24	keyboard	Клавіатур
25	scanner	Сканерів

4 КОНТРОЛЬНІ ПИТАННЯ

1. Укажіть основні інтерфейси та методи JDBC API. Яке їх призначення?
2. Укажіть основні команди мови визначення даних DDL. Яке їх призначення?
3. Укажіть основні команди мови обробки даних DML. Яке їх призначення?
4. Укажіть основні команди мови керування даними DCL. Яке їх призначення?
5. Які типи даних SQL використовує СКБД MySQL?
6. Які функції виконує метод *executeQuery()* інтерфейсу *Statement*?
7. Які функції виконує метод *executeUpdate()* інтерфейсу *Statement*?
8. Які функції виконує метод *getFloat()* інтерфейсу *ResultSet*?
9. Які функції виконує метод *getString()* інтерфейсу *ResultSet*?
10. Які функції виконує метод *getInt()* інтерфейсу *ResultSet*?
11. Наведіть SQL-твердження яке повертає число рядків у таблиці *goods*.
12. Наведіть SQL-твердження яке витягає всі дані з таблиці *goods*.
13. Які важливі методи містить інтерфейс *ResultSet*?
14. Яку функцію виконує метод *createStatement()* інтерфейсу *Connection*?

5 ЗМІСТ ПИСЬМОВОГО ЗВІТУ

Письмовий звіт повинен містити.

1. Тему та мету роботи.
2. Тексти файлів програми-клієнта та Web-сервіса, які були створені при рішенні завдання
3. Крім того, приведіть та пояснить XML-файли, якими обмінюються клієнт та сервер при виконанні завдання 3.
4. Відповіді на контрольні питання.
5. Висновки.

Письмовий звіт повинен бути оформлений на комп'ютері, мати титульну сторінку із зазначенням назви дисципліни, теми

лабораторної роботи, автора звіту та викладача, дати складання письмового звіту. Далі на кожній сторінці звіту в колонтитулах має міститись прізвище, ім'я та номер групи студента (верхній колонтитул), тема лабораторної роботи та номер сторінки (нижній колонтитул). В іншому випадку звіт може бути оформлений в рукописному варіанті.

6 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Хабибулин И.Ш. Создание распределенных приложений на Java 2. – СПб.: БХВ-Петербург, 2002. – 704 с.
2. Хабибулин И.Ш. Самоучитель Java. – СПб.: БХВ-Петербург, 2001. – 464 с.
3. Хабибулин И.Ш. Создание Web-служб средствами Java. – СПб.: БХВ-Петербург, 2003. – 400 с.
4. Машнин Т.С. Web-сервисы Java. – СПб.: БХВ-Петербург, 2012. – 560 с.
5. Hewitt E. Java SOA Cookbook. – Sebastopol, CA: O'Reilly Media, 2009. – 742 p.
6. Bean J. SOA and Web Services Interface Design: Principles, Techniques, and Standards. – Burlington, MA, USA: Elsevier Inc., 2010. – 360 p.
7. DuBois Paul. MySQL Cookbook, Third Edition. – Sebastopol: O'Reilly Media, 2014. – 866 p.