

Lab 4: LƯU TRỮ

I. MỤC TIÊU:

- Cung cấp các giải pháp lưu trữ trong chương trình java:
 - File: txt, properties
 - JDBC: truy xuất database sử dụng JDBC API

II. TÓM TẮT LÝ THUYẾT:

A. Text File

1. Đọc file:

```
try (BufferedReader br = new BufferedReader(new FileReader(FILENAME))) {
    String line;
    while ((line = br.readLine()) != null) {
        //xử lý line
    }
} catch (IOException e) {
    e.printStackTrace();
}
```

2. Ghi file:

```
try (BufferedReader br = new BufferedReader(new FileReader(FILENAME))) {
    String content = "This is the content to write into file\n";
    bw.write(content);
} catch (IOException e) {
    e.printStackTrace();
}
```

3. Ghi nối file:

```
BufferedWriter bw = null;
FileWriter fw = null;
String data = " This is new content";
File file = new File(FILENAME);
// if file doesnt exists, then create it
if (!file.exists()) {
    file.createNewFile();
}
fw = new FileWriter(file.getAbsolutePath(), true);
BufferedWriter bw = new BufferedWriter(fw);
bw.write(data);

//Close
bw.close();
fw.close();
```

B. Properties File:**1. Đọc file:**

```
Properties prop = new Properties();
InputStream input = new FileInputStream("config.properties");
// load a properties file
prop.load(input);
// get the property value and print it out
prop.getProperty("database");
prop.getProperty("user");
prop.getProperty("pwd");
input.close();
```

2. Ghi file:

```
Properties prop = new Properties();
OutputStream output = null;
output = new FileOutputStream("config.properties");
// set the properties value
prop.setProperty("database", "localhost");
prop.setProperty("user", "kha.hodinh");
prop.setProperty("pwd", "123456");
// save properties to project root folder
prop.store(output, null);
// close file
output.close();
```

3. Đọc properties file từ classpath:

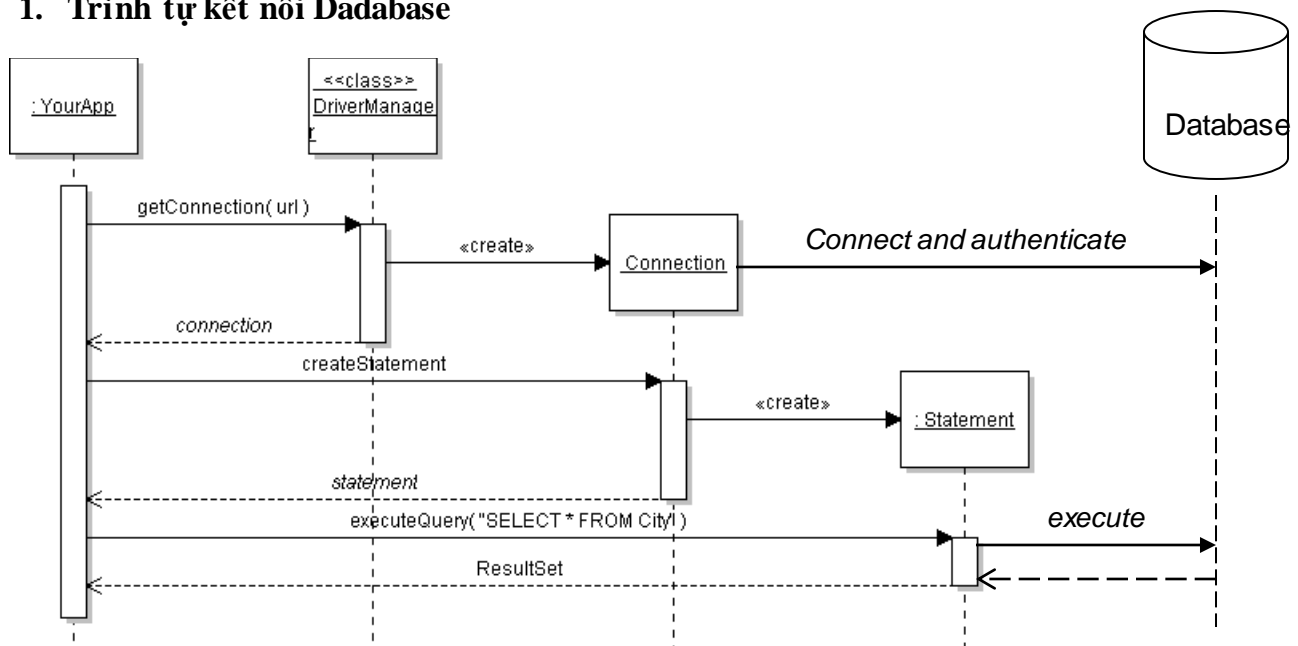
```
Properties prop = new Properties();
String filename = "config.properties";
InputStream input = null;
input = App3.class.getClassLoader().getResourceAsStream(filename);
//load a properties file from class path, inside static method
prop.load(input);
input.close();
```

4. Đọc properties file bằng ResourceBundle:

```
ResourceBundle bundle = ResourceBundle.getBundle("config"); //không có .properties
bundle.getString("database"); // localhost
bundle.getString("user"); // kha.hodinh
bundle.getString("pwd"); // 123456
```

C. JDBC:

1. Trình tự kết nối Database



2. Cấu hình DriverManager

URL	Database
<code>jdbc:mysql://se.cpe.ku.ac.th/world</code>	MySQL database trên server
<code>jdbc:mysql://localhost:3306/world</code>	MySQL database trên local
<code>jdbc:derby:/database/world</code>	Derby database
<code>jdbc:hsqldb:file:/database/world</code>	HSQldb database

- Cấu trúc URL tổng quát :

protocol:subprotocol://host:port/database

3. Tạo kết nối Database

- Có hai cách tạo một đối tượng kết nối Database:

(1) Sử dụng DriverManager - Ứng dụng stand-alone

(2) Sử dụng DataSource - Ứng dụng Web hoặc Phân tán.

- DriverManager có 2 phương thức khởi tạo kết nối:

(1) `connection = DriverManager.getConnection(url, username, password);`


(2) `connection = DriverManager.getConnection(url, properties);`

JDBC Drivers và Class Name:

Database	Driver JAR file	Driver Class Name
MySQL	<code>mysql-connector-java-5.X.Y.jar</code>	<code>com.mysql.jdbc.Driver</code>
Derby embedded	<code>derby.jar</code>	<code>org.apache.derby.jdbc.EmbeddedDriver</code>
Derby client-server	<code>derbyclient.jar</code>	<code>org.apache.derby.jdbc.ClientDriver</code>
HSQLDB	<code>hsqldb.jar</code>	<code>org.hsqldb.jdbcDriver</code>
SQLServer	?	?
Oracle	?	?
Access	?	?

Sinh viên điền thêm các thông số còn trống.


Lưu ý: Khi class Driver được nạp, nó tạo ra một thể hiện và đăng kí với DriverManager. Một ngoại lệ “ClassNotFoundException” gây ra khi driver không được nạp.

 **Đối tượng Statement:** Được tạo sử dụng phương thức `createStatement` của class Connection

`Statement stm=connection.createStatement();`

4. Thực hiện truy vấn và hiển thị kết quả.

Sử dụng phương thức `Statement.executeQuery(string)` thực hiện truy vấn Database

 Nếu lệnh là một truy vấn (Select), sử dụng phương thức `executeQuery()` của đối tượng statement:

ResultSet rs=smt.executeQuery(S1)

Phương thức `executeQuery` trả về đối tượng **rs** dạng **ResultSet** (dạng bảng)

Phương thức **rs.next()** di chuyển tới hàng kế của **ResultSet** hay NULL nếu hết bảng.

Tham khảo các thuộc tính trong mỗi hàng sử dụng phương thức **getXXX**(getString, getInteger, getDouble, ...) với tham số là tên thuộc tính hoặc thứ tự thuộc tính

🚦 Nếu câu lệnh SQL có dạng Action (insert ,update ,delete) sử dụng phương thức **executeUpdate** :

```
int rowcount= smt.executeUpdate(S2)
```

Phương thức **executeUpdate** trả về một giá trị nguyên là số hàng bị tác động

Một số phương thức **Statement** ..

Command	Statement Method	Ví dụ:
SELECT	executeQuery("select");	ResultSet rs = stmt.executeQuery("SELECT * FROM city);
INSERT, UPDATE, DELETE	executeUpdate("sql command")	int count = stmt.executeUpdate("DELETE FROM city WHERE name='Bangsaen'")
INSERT	executeUpdate("sql command", Statement.RETURN_GENERATED_KEYS)	Thực hiện INSERT và tạo “auto-generated key”. Thường dùng cho DAO.
Khác	execute("sql command")	boolean results = execute("CREATE TABLE countryflag ...")

5. Sử dụng đối tượng PreparedStatement

Ví dụ: Tìm tất cả EMPLOYEE ứng với FNAME:

```
PreparedStatement pstmt = connection.prepareStatement(
    "SELECT * FROM EMPLOYEE WHERE FNAME=?");

// Chèn giá trị vào PreparedStatement
pstmt.setString(1,"kathy"); // Thay thế kí tự ? bằng chuỗi "kathy"
ResultSet rs = pstmt.executeQuery();
```

Lưu ý : vị trí kí tự ? trong câu lệnh sẽ có thứ tự tương ứng 1,2,......

Ví dụ: Để thêm mới Employee chúng ta cần 3 tham số (SSN, FNAME, LNAME):

```
PreparedStatement pstmt = connection.prepareStatement(  
    "INSERT INTO EMPLOYEE (SSN, FNAME, LNAME)  VALUES (?,?,?) " );  
  
// Chèn dữ liệu  
  
pstmt.setString( 1, "02341" );  
pstmt.setString( 2, " Pole" );  
pstmt.setString( 3, "Smith" );  
  
int count = pstmt.executeUpdate( );
```

Thuận lợi khi sử dụng PreparedStatement so với Statement ?

III. NỘI DUNG THỰC HÀNH :

1. Mở Lab4_1.java :

```
1  import java.io.*;  
2  import java.util.*;  
3  import java.util.logging.Level;  
4  import java.util.logging.Logger;  
5  public class Lab4_1 {  
6      public static void main(String[] args) {  
7          // TODO code application logic here  
8          Scanner input = new Scanner(System.in);  
9  
10         try {BufferedWriter br = new BufferedWriter(new FileWriter("data1.txt"))} {  
11             System.out.println("Nhap noi dung file: ");  
12             String content = input.nextLine();  
13             br.write(content);  
14             System.out.println("Done!");  
15         } catch (IOException ex) {  
16             Logger.getLogger(Lab4_1.class.getName()).log(Level.SEVERE, null, ex);  
17         }  
18  
19         //Doc noi dung file vua tao  
20  
21     }  
22 }
```

Yêu cầu :

- Chạy chương trình, nhập nội dung file.
- Kiểm tra nội dung file **data1.txt** vừa tạo trong thư mục Lab4
- Viết code đọc nội dung file **data1.txt** vừa tạo, in kết quả ra màn hình.

Bài tập 1:

- Viết chương trình nối nội dung : « Họ tên » « Lớp » vào file **data1.txt**.

2. Mở Lab4_2.java :

```
1  import java.util.*;
2
3  public class Lab4_2 {
4      public static void main(String[] args) {
5          // TODO code application logic here
6          ResourceBundle bundle = ResourceBundle.getBundle("prop1");
7          bundle.keySet();
8          for(String key:bundle.keySet())
9              System.out.println("key = " + key + ", value = " + bundle.getString(key));
10     }
11 }
```

Yêu cầu :

- Chạy chương trình, quan sát kết quả
- Thêm 2 thuộc tính vào file prop1.properties như sau :

khoa = CNTT

chucvu = Giảng viên

- Chạy lại chương trình, quan sát kết quả

Bài tập 2:

- Tạo file **prop2.properties** có nội dung:

hoten = « Họ tên »

mssv = « MSSV »

lop = « Lớp »

- Viết chương trình đọc nội dung file vừa tạo, in kết quả ra màn hình.

3. Lab4_3 :

- Start WAMP server , tạo database **labjdbc** ;
- Import file **labjdbc.sql** vào database vừa tạo ;
- Mở file **Lab4_3.java** :

```
1 import java.sql.*;
2 public class Lab4_3 {
3     public static void main(String[] args) {
4         // TODO code application logic here
5         Connection con = null;
6         try {
7             Class.forName("com.mysql.jdbc.Driver");
8             con = DriverManager.getConnection("jdbc:mysql://localhost/labjdbc", "root", "");
9             Statement stmt = con.createStatement();
10            String sql = "select code, name, population from country limit 0,10";
11            ResultSet rs = stmt.executeQuery(sql);
12            while (rs.next()) {
13                System.out.println(rs.getString("code") + " - "
14                    + rs.getString("name") + " - " + rs.getDouble("population"));
15            }
16            con.close();
17        } catch (Exception e) {
18            e.printStackTrace();
19        }
20    }
21 }
```

Yêu cầu:

- Chạy chương trình, quan sát kết quả.
- Viết code truy vấn những nước có tên bắt đầu bằng chữ A.

Bài tập 3:

- Viết chương trình truy vấn những thành phố gồm các field: name, district, population có dân số nhỏ hơn 100000 (lấy 20 kết quả đầu)

4. Lab4_4 :

- Cấu trúc dự án:

domain: chứa các đối tượng thông tin

dao: chứa các đối tượng truy xuất CSDL

- Mở Lab4_4.java:

```
7 public class Lab4_4 {
8     public static void main(String[] args) {
9         Scanner input = new Scanner(System.in);
10        System.out.println("Nhap countrycode: ");
11        String code = input.nextLine();
12        try {
13            String sql = "select * from city where countrycode = ?";
14            PreparedStatement pstmt = DBHelper.getConnection().prepareStatement(sql);
15            pstmt.setString(1, code);
16            ResultSet rs = pstmt.executeQuery();
17            if (rs.isBeforeFirst()) {
18                while (rs.next()) {
19                    System.out.printf("Id=%d; Name=%s; CountryCode=%s; "
20                                     + "District=%s; Population=%d;\n",
21                                     rs.getInt(1), rs.getString(2), rs.getString(3),
22                                     rs.getString(4), rs.getInt(5));
23                }
24            } else {
25                System.out.println("Khong tim thay!");
26            }
27        } catch (SQLException ex) {
28            Logger.getLogger(Lab4_4.class.getName()).log(Level.SEVERE, null, ex);
29        } finally {
30            DBHelper.close();
31        }
32    }
33 }
```

- Mở file **dao/DbHelper.java**:

```
10 public class DbHelper {
11     private static Connection con = null;
12     public static Connection getConnection() {
13         if (con == null) {
14             //Doc file cau hinh properties
15             ResourceBundle bundle = ResourceBundle.getBundle("dao.db");
16             String url = bundle.getString("url");
17             String user = bundle.getString("user");
18             String pwd = bundle.getString("pwd");
19             try {
20                 Class.forName("com.mysql.jdbc.Driver");
21                 con = DriverManager.getConnection(url, user, pwd);
22             } catch (ClassNotFoundException | SQLException ex) {
23                 Logger.getLogger(DbHelper.class.getName()).log(Level.SEVERE, null, ex);
24             }
25         }
26         return con;
27     }
28     public static void close() {
29         if (con != null) {
30             try {
31                 con.close();
32             } catch (SQLException ex) {
33                 Logger.getLogger(DbHelper.class.getName()).log(Level.SEVERE, null, ex);
34             }
35         }
36     }
37 }
```

Yêu cầu:

- Chạy chương trình, nhập vào các giá trị ‘NLD’, ‘ABC’ quan sát kết quả.
- Viết thêm code trong Lab4_4.java, thêm một thành phố mới :

Name=LT CSDL; CountryCode=VNM; District=VN; Population=1000;

Bài tập 4 :

- Viết chương trình hiển thị những country có population < 50000 và localname được nhập từ bàn phím, lấy 10 kết quả đầu.

5. Lab4_5:

- Mở Lab4_5.java :

```
6 public class Lab4_5 {
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9         System.out.println("Nhap countrycode: ");
10        String code[] = input.nextLine();
11        try {
12            String sql = "select * from city where countrycode = ?";
13            PreparedStatement pstmt = DbHelper.getConnection().prepareStatement(sql);
14            pstmt.setString(1, code);
15            ResultSet rs = pstmt.executeQuery();
16            if (rs.isBeforeFirst()) {
17                while (rs.next()) {
18                    City city = City.fromResult(rs);
19                    System.out.println(city);
20                }
21            } else {
22                System.out.println("Khong tim thay!");
23            }
24        } catch (SQLException ex) {
25            Logger.getLogger(Lab4_4.class.getName()).log(Level.SEVERE, null, ex);
26        } finally {
27            DbHelper.close();
28        }
29    }
30 }
```

- Mở file domain/City.java:

```
31 class City {
32     private int id;
33     private String name;
34     private String countryCode;
35     private String district;
36     private int population;
37
38     public static City fromResult(ResultSet rs) throws SQLException {
39         City city = new City();
40         city.setId(rs.getInt(1));
41         city.setName(rs.getString(2));
42         city.setCountryCode(rs.getString(3));
43         city.setDistrict(rs.getString(4));
44         city.setPopulation(rs.getInt(5));
45         return city;
46     }
47 }
```

Yêu cầu:

- Chạy chương trình, nhập vào giá trị NLD, ABC, quan sát kết quả.
- Viết thêm code, thêm một đối tượng City mới :
City ('Tay Ninh', 'VNM', 'VN', 7000);

Bài tập 5 :

- Viết chương trình hiển thị những city có district nhập từ bàn phím.
- Chạy chương trình nhập vào VN.

6. Mở Lab4_6.java:

```
7 public class Lab4_6 {
8     public static void main(String[] args) {
9         // TODO code application logic here
10        CityDAO cityDao = new CityDAO();
11        List<City> cities = cityDao.findAll(10); // Lay 10 ket qua
12        for (City city : cities)
13            System.out.println(city);
14        DBHelper.close();
15    }
16 }
17 class CityDAO {
18     private Connection con;
19     public CityDAO() {
20         con = DBHelper.getConnection();
21     }
22     public List<City> findAll(int rows) {
23         List<City> result = null;
24         try {
25             PreparedStatement pstmt = con.prepareStatement("select * from city limit 0, " + rows);
26             ResultSet rs = pstmt.executeQuery();
27             if (rs.isBeforeFirst()) {
28                 result = new ArrayList<>();
29                 while (rs.next()) {
30                     City city = City.fromResult(rs);
31                     result.add(city);
32                 }
33             }
34         } catch (SQLException ex) {
35             Logger.getLogger(CityDAO.class.getName()).log(Level.SEVERE, null, ex);
36         }
37         return result;
38     }
39 }
```

Yêu cầu:

- Chạy chương trình, quan sát kết quả.
- Trong lớp CityDAO, viết thêm phương thức **findByCountryCode()** tìm city theo countrycode.
- Trong phương thức main, viết code hiển thị city theo countrycode.

Bài tập 6 :

- Trong lớp CityDAO, viết thêm phương thức **save()** lưu đối tượng city, **delete()** xóa đối tượng city theo id.
- Viết chương trình chính thêm 3 đối city mới, hiển thị 3 đối tượng vừa tạo, sau đó xóa các đối tượng đó.

IV. BÀI TẬP LÀM THÊM:

Thêm một bảng Authentication vào Database, Thiết kế một ứng dụng GUI kiểm tra tài khoản đăng nhập. Nếu thành công hiển thị GUI các chức năng của ứng dụng.

Bảng Authentication:

FName	LName	Password
snow	white	123456

GUI đề nghị:

