

CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM (504073)

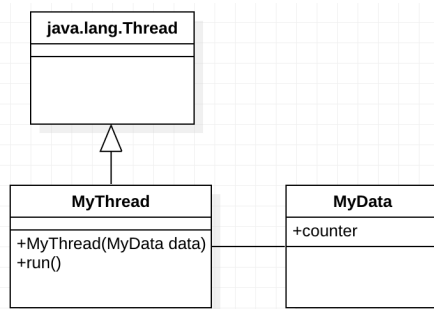
(TUẦN 06)

BIÊN SOẠN: MAI VĂN MẠNH

Java

Bài 1. Bài tập minh họa vấn đề xảy ra khi 2 thread cùng ghi vào một nguồn dữ liệu.

- Tạo lớp **MyData** và lớp **MyThread** như mô tả sau:



- Phương thức **run()** của **MyThread** chạy một vòng lặp 1 triệu lần, mỗi lần lặp tăng giá trị **counter** của đối tượng **MyData** lên một đơn vị.
- Trong hàm main, tạo một đối tượng **data** thuộc kiểu **MyData**. Tạo tiếp hai đối tượng **th1**, **th2** thuộc kiểu **MyThread**, hai đối tượng này cùng nhận **data** làm tham số cho phương thức khởi tạo.
- Chạy hai thread **th1**, **th2** đồng thời buộc main thread tạm ngưng thực thi cho đến khi chúng chạy xong, sau đó in ra màn hình giá trị của biến đếm **counter**. Kiểm tra xem giá trị của biến đếm có bằng **2,000,000** không (đáp án: không).
- Điều chỉnh lại **MyThread**, sử dụng cơ chế **synchronization** để khắc phục tình trạng hai thread ghi dữ liệu cùng lúc, đảm bảo kết quả cho ra biến đếm **2,000,000**.

Bài 2. Sử dụng JDBC để thao tác dữ liệu trong SQL Server. Tạo lớp **POCO** đại diện cho đối tượng sinh viên, lớp **StudentDAO** chứa các phương thức thao tác dữ liệu sinh viên như hình bên dưới:

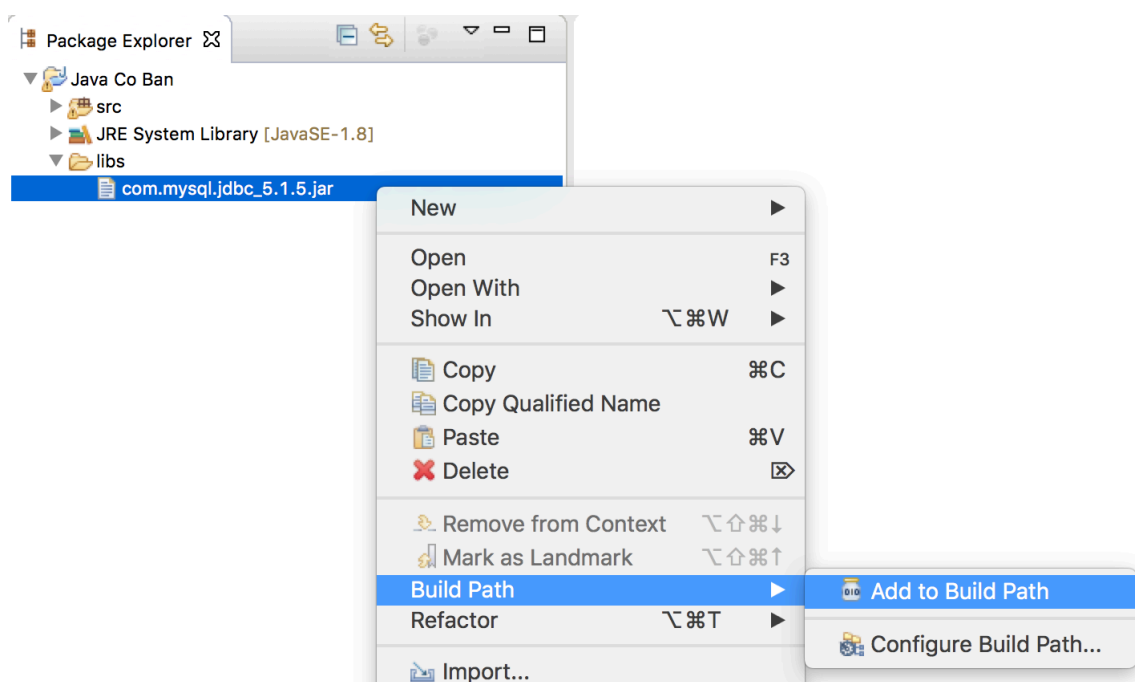
StudentDAO
+add(Student s) +delete(String id) +update(Student s) +getAll(): List<Student> +getById(String id): Student +getByAge(int age): List<Student> +getByCountry(String country): List<Student> +getStudentSize() +getOldestStudent(): Student

Student
+Id: Int +Name: String +Birthday: Date +Gender: Boolean +Email: String

- **StudentDAO** đọc/ghi dữ liệu từ SQL Server sử dụng JDBC.
- Viết chương trình chứa hàm main để kiểm tra lại các chức năng vừa thực hiện.

HƯỚNG DẪN THIẾT LẬP JDBC

Để sử dụng JDBC trước hết cần tải về Driver tương ứng với hệ quản trị cơ sở dữ liệu mà chúng ta muốn sử dụng. Các Driver này thường là tập tin *.jar và có thể được tải tại trang web của các nhà cung cấp (Microsoft, Oracle, Mysql). Sau khi có được thư viện, chúng ta chép vào project và thêm chúng vào classpath của project. Nếu làm việc với Eclipse IDE, chúng ta có thể tạo một thư mục với tên libs, sau đó chép các thư viện *.jar vào thư mục này và thêm chúng vào classpath bằng cách chọn thư viện, nhấn phải chuột và chọn Build Path > Add to Build Path.



Các bước thực hiện để thao tác với CSDL

1. Đăng ký driver
2. Tạo và thực hiện kết nối tới CSDL
3. Thực hiện truy vấn

4. Đóng kết nối

1. Đăng ký Driver: Ví dụ này thực hiện với cơ sở dữ liệu Oracle

Cách 1 (thường được sử dụng):

```
try {  
    Class.forName("oracle.jdbc.driver.OracleDriver");  
}  
catch(ClassNotFoundException ex) {  
    System.out.println("Error: unable to load driver class!");  
    System.exit(1);  
}
```

Cách 2:

```
try {  
    Driver myDriver = new oracle.jdbc.driver.OracleDriver();  
    DriverManager.registerDriver( myDriver );  
}  
catch(ClassNotFoundException ex) {  
    System.out.println("Error: unable to load driver class!");  
    System.exit(1);  
}
```

Trong đó "oracle.jdbc.driver.OracleDriver" là tên class drive của Oracle.

- MySql: "com.mysql.jdbc.Driver"
- SQL Server: "com.microsoft.sqlserver.jdbc.SQLServerDriver"

2. Tạo và kết nối tới CSDL

Trước khi bắt đầu kết nối, cần phải xác định được Database URL. Một số ví dụ về URL:

RDBMS	JDBC driver name	URL format
MySQL	com.mysql.jdbc.Driver	jdbc:mysql:// hostname/ databaseName
ORACLE	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@ hostname:port Number:databaseName
DB2	COM.ibm.db2.jdbc.net.DB2Driver	jdbc:db2: hostname:port Number/databaseName
Sybase	com.sybase.jdbc.SybDriver	jdbc:sybase:Tds: hostname: port Number/databaseName

+ SQL Server:

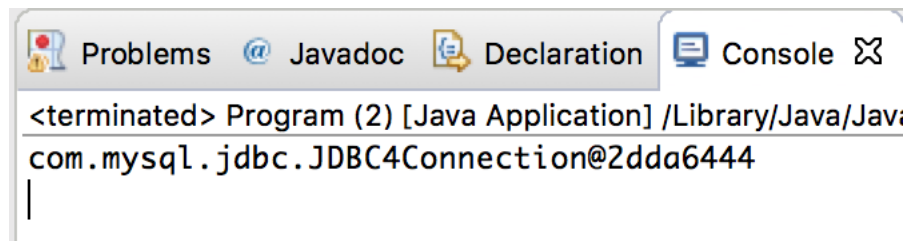
jdbc:sqlserver://localhost;user=mvmanh;password=111111;databaseName=CdJava

+ My SQL: **jdbc:mysql://localhost/CdJava**

Thực hiện kết nối tới CSDL

```
String url = "jdbc:mysql://localhost/CdJava";  
String username = "root";  
String password = "";  
  
try {  
  
    //đăng ký driver  
    Class.forName("com.mysql.jdbc.Driver");  
  
    // tạo và thực hiện kết nối  
    Connection conn = (Connection) DriverManager.getConnection(url, username, password);  
  
    // kiểm tra kết nối  
    System.out.println(conn);  
  
    // đóng kết nối  
    conn.close();  
} catch (ClassNotFoundException e) {  
    e.printStackTrace();  
} catch (SQLException e) {  
    e.printStackTrace();  
}  
}
```

Sau khi kết nối thành công sẽ có kết quả tương tự như sau



3. Thực hiện truy vấn

3.1 Truy vấn không trả về kết quả

```
// tạo và thực hiện kết nối
Connection conn = (Connection) DriverManager.getConnection(url, username, password);

String sql = "Create table student(name nvarchar(30) primary key, age int)";
Statement stm = (Statement) conn.createStatement(); // tạo statement
stm.executeUpdate(sql); // thực thi câu lệnh

stm.close();
conn.close();
```

3.2 Kiểm tra kết quả của truy vấn

```
String sql = "insert into student values(?,?)";
PreparedStatement ptm = (PreparedStatement) conn.prepareStatement(sql);

ptm.setString(1, "Le Minh");
ptm.setInt(2, 23);

int rows = ptm.executeUpdate();

if (rows == 1) {
    System.out.println("Them thanh cong");
}
```

3.3 Truy vấn trả về dữ liệu

```
String sql = "select * from student";
Statement stm = (Statement) conn.createStatement();

// truy vấn
ResultSet data = stm.executeQuery(sql);

List<Student> studentList = new ArrayList<Student>();

// trích xuất dữ liệu
while (data.next()) {
    String name = data.getString(1);
    int age = data.getInt(2);
    Student s = new Student(name, age);
    studentList.add(s);
}

// dữ liệu đã được lưu vào ArrayList
for (Student s: studentList) {
    System.out.println(s);
}

stm.close();
conn.close();
```