JAVA JDBC

Thực hành 1

Nội dung chính

- Ôn lại ngôn ngữ java
- Ôn lại truy xuất CSDL với JDBC
- Thời lượng: 2 tiết

- Viết được lớp truy cập DB
- Chạy được query truy vấn
- Xuất được dữ liệu ra console

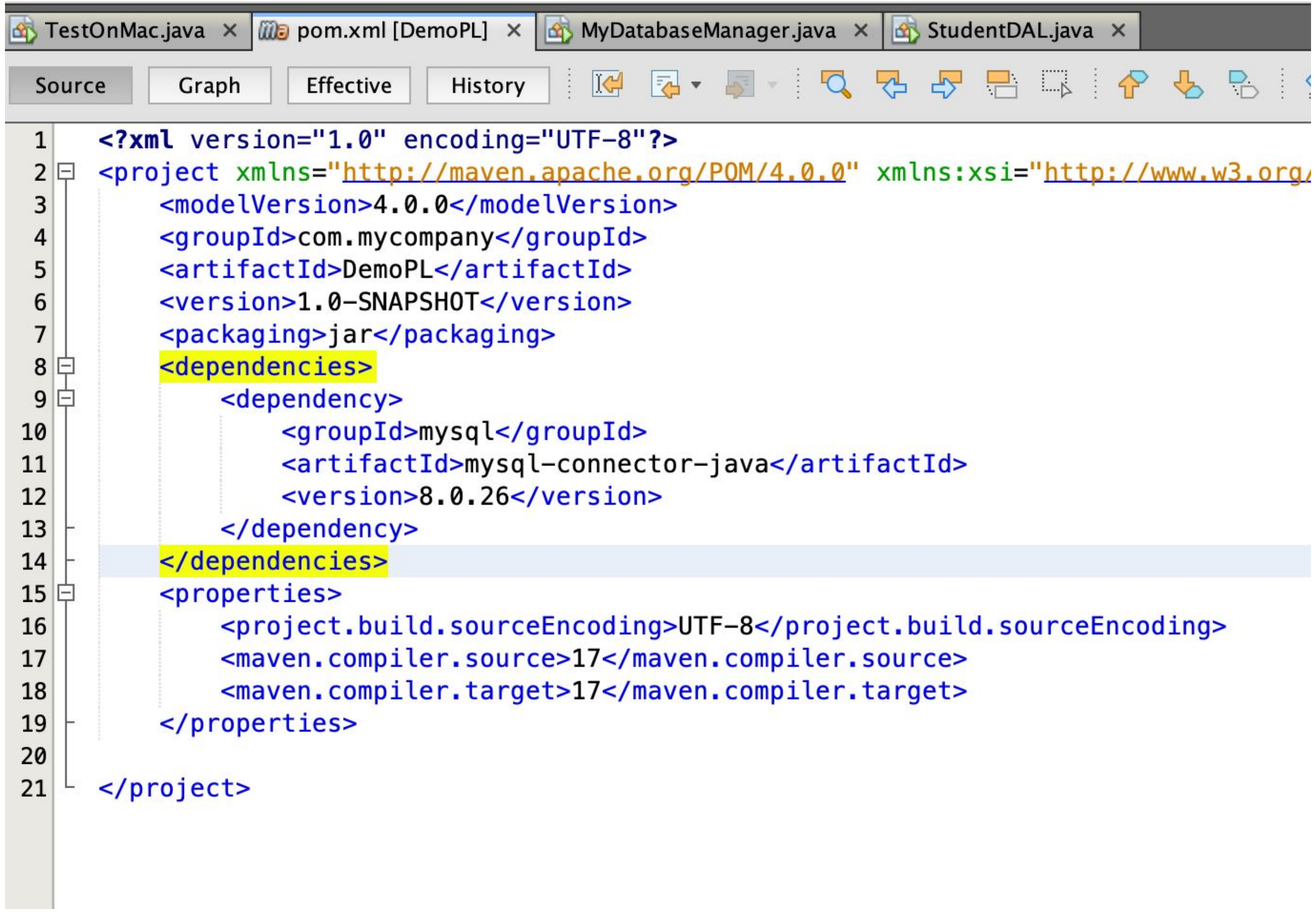
1. Chuẩn bị csdl

Tải CSDL School tại trang web của giảng viên

Import CSDL, sử dụng phpMyAdmin

2. Tạo và cấu hình project

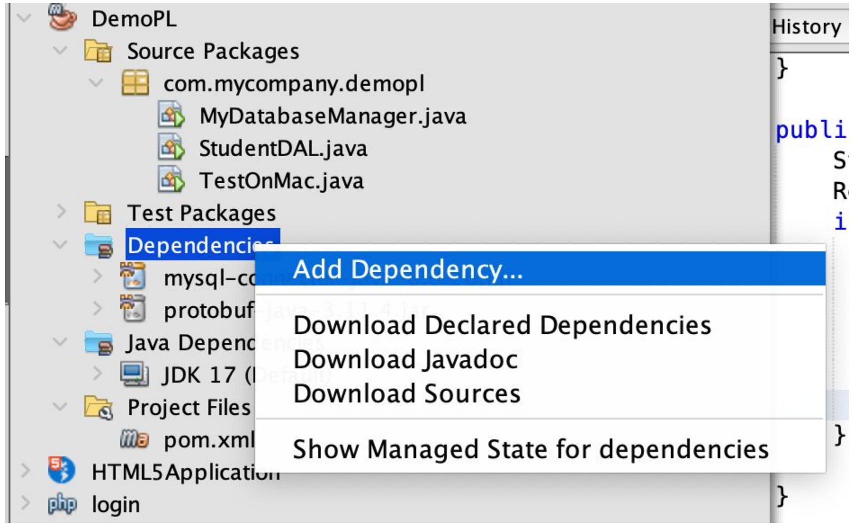
- Project mới (Maven Project)
- Cấu hình thêm các class thư viện (Add Dependency)

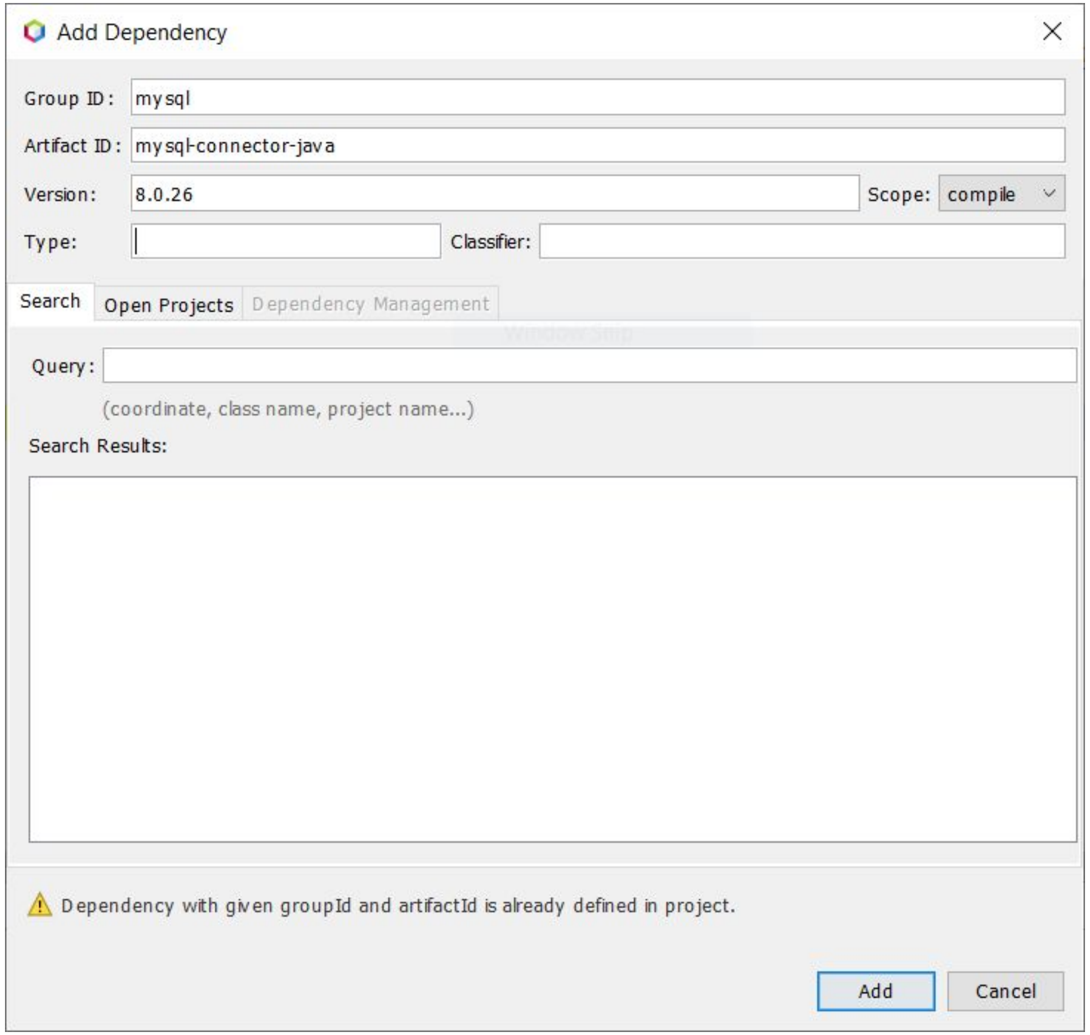


```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>DemoPL</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.26</version>
        </dependency>
    </dependencies>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
    </properties>

</project>
```

2. Tạo và cấu hình project

- Project mới (Maven Project)
- Cấu hình thêm các class thư viện (Add Dependency)





Group ID: mysql

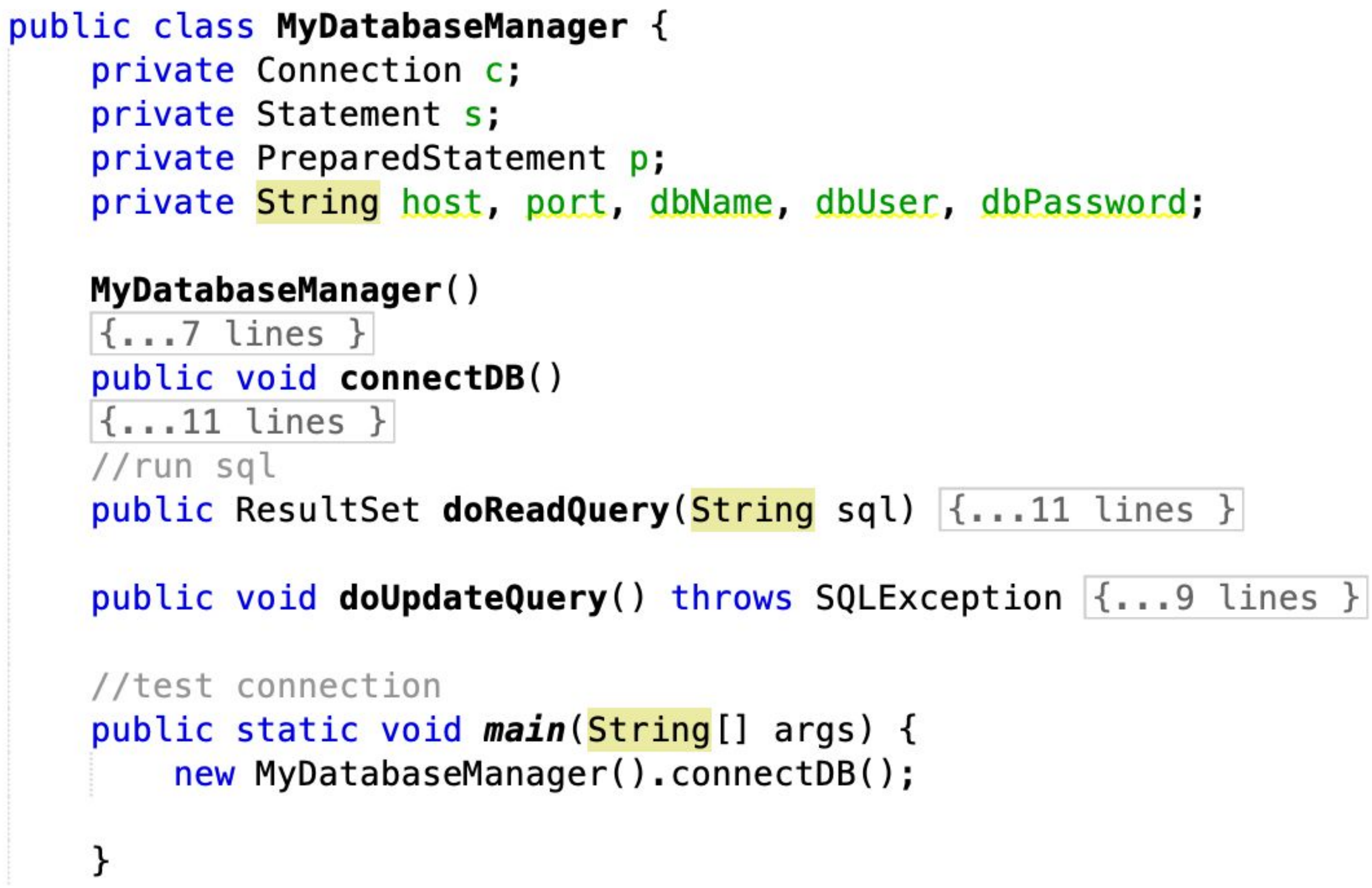
Artifact ID: mysql-connector-java

Version: 8.0.26 Scope: compile

Type: Classifier:

⚠ Dependency with given groupId and artifactId is already defined in project.

3. Code class đầu tiên



```
public class MyDatabaseManager {
    private Connection c;
    private Statement s;
    private PreparedStatement p;
    private String host, port, dbName, dbUser, dbPassword;

    MyDatabaseManager()
    {...7 lines }
    public void connectDB()
    {...11 lines }
    //run sql
    public ResultSet doReadQuery(String sql) {...11 lines }

    public void doUpdateQuery() throws SQLException {...9 lines }

    //test connection
    public static void main(String[] args) {
        new MyDatabaseManager().connectDB();

    }
```

3. Code class đầu tiên

Hàm khởi tạo

- Gán các biến thông tin của CSDL

```
MyDatabaseManager()
    {
        host = "localhost";
        port = "3306";
        dbUser = "root";
        dbName = ”school";
        dbPassword = "";
    }
```

3. Code class đầu tiên

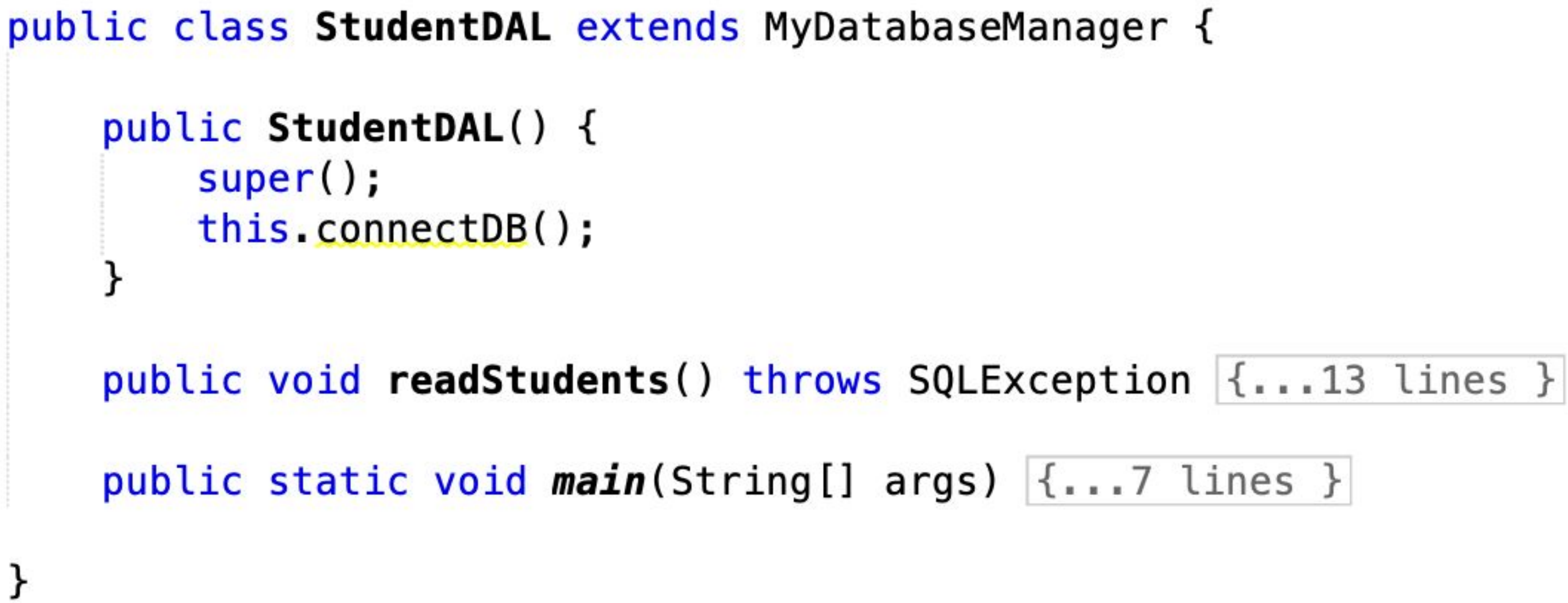
- Khởi tạo kết nối
- Khởi tạo statement

```
public void connectDB()
{
String dbPath = "jdbc:mysql://" + host + ":" + port + "/"
+ dbName + "?useUnicode=yes&characterEncoding=UTF-8";
try {
c = (Connection) DriverManager.getConnection(dbPath, dbUser, dbPassword);
s = c.createStatement();
System.out.print("Connected");
} catch (SQLException ex) {
System.out.print(ex.getMessage());
}
}
```

3. Code class đầu tiên

```
public ResultSet doReadQuery(String sql) {
ResultSet rs = null;
try {
rs = s.executeQuery(sql);
} catch (SQLException ex) {
Logger.getLogger(MyDatabaseManager.class.getName())
.log(Level.SEVERE, null, ex);
}
return rs;
}
```

4. Viết class thứ 2



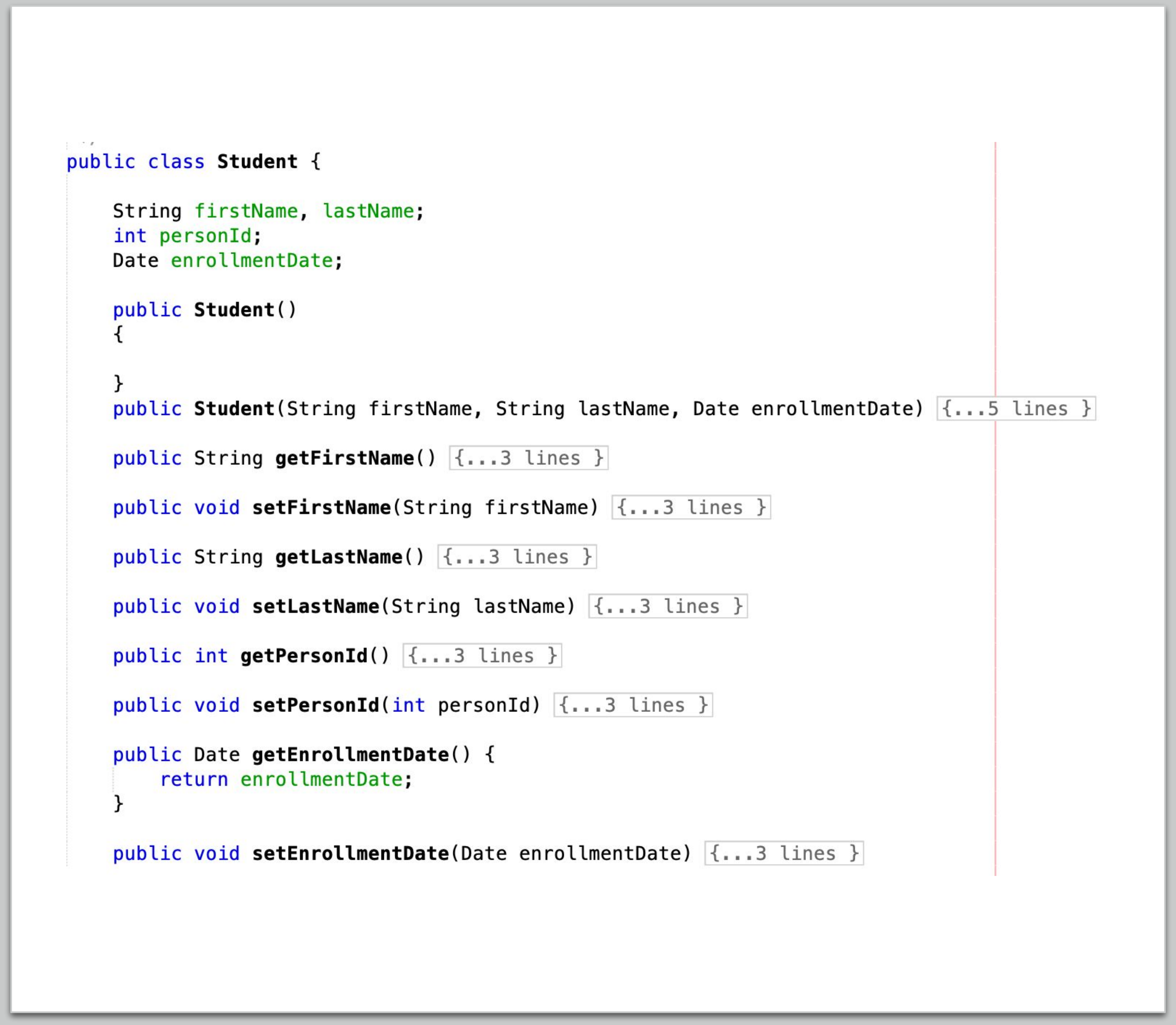
```
public class StudentDAL extends MyDatabaseManager {

    public StudentDAL() {
        super();
        this.connectDB();
    }

    public void readStudents() throws SQLException {...13 lines }

    public static void main(String[] args) {...7 lines }

}
```

4. Viết class thứ 2

```
public void readStudents() throws SQLException {
String query = "SELECT * FROM Person WHERE EnrollmentDate >0";
ResultSet rs = this.doReadQuery(query);
if (rs != null) {
int i = 1;
while (rs.next()) {
System.out.print(i + " - ");
System.out.println(rs.getString("Lastname") + " " + rs.getString("Firstname"));
i++;
}
}
}
```

5. Viết class thứ 3

- Lớp Student (lớp thực thể) sử dụng khi chèn thêm, cập nhật thông tin của 01 sinh viên



```
public class Student {

    String firstName, lastName;
    int personId;
    Date enrollmentDate;

    public Student()
    {

    }
    public Student(String firstName, String lastName, Date enrollmentDate) {...5 lines }

    public String getFirstName() {...3 lines }

    public void setFirstName(String firstName) {...3 lines }

    public String getLastName() {...3 lines }

    public void setLastName(String lastName) {...3 lines }

    public int getPersonId() {...3 lines }

    public void setPersonId(int personId) {...3 lines }

    public Date getEnrollmentDate() {
        return enrollmentDate;
    }

    public void setEnrollmentDate(Date enrollmentDate) {...3 lines }
```

6. Bổ sung các hàm CRUD

- Trên lớp StudentDAL bổ sung các hàm CRUD
  - Chèn thêm một sinh viên
  - Cập nhật thông tin sinh viên
  - Xoá sinh viên
  - Tìm kiếm

Xem danh sách

```
public void readStudents() throws SQLException {
String query = "SELECT * FROM Person WHERE EnrollmentDate >0";
ResultSet rs = this.doReadQuery(query);
if (rs != null) {
int i = 1;
System.out.println("TT \t PersonID \t FirstName \t\t LastName");
while (rs.next()) {
System.out.print(i + "\t" + rs.getInt("PersonID"));
System.out.println("\t\t" + rs.getString("FirstName") + "\t\t\t" + rs.getString("LastName"));
i++;
}
}
}
```

Update

```
public int updateStudent(Student s) throws SQLException {
String query = "Update Person SET FirstName = ? , LastName = ? "
+ " WHERE PersonID = ?";
PreparedStatement p = this.getConnection().prepareStatement(query);
p.setString(1, s.getFirstName());
p.setString(2, s.getLastName());
p.setInt(3, s.getPersonId());
int result = p.executeUpdate();
return result;
}
```

Insert

```
public int insertStudent(Student s) throws SQLException {
String query = "Insert Person (FirstName, LastName, EnrollmentDate) VALUES (?, ?, ?)";
PreparedStatement p = this.getConnection().prepareStatement(query);
p.setString(1, s.getFirstName());
p.setString(2, s.getLastName());
p.setString(3, s.getEnrollmentDate().toString());
int result = p.executeUpdate();
return result;
}
```

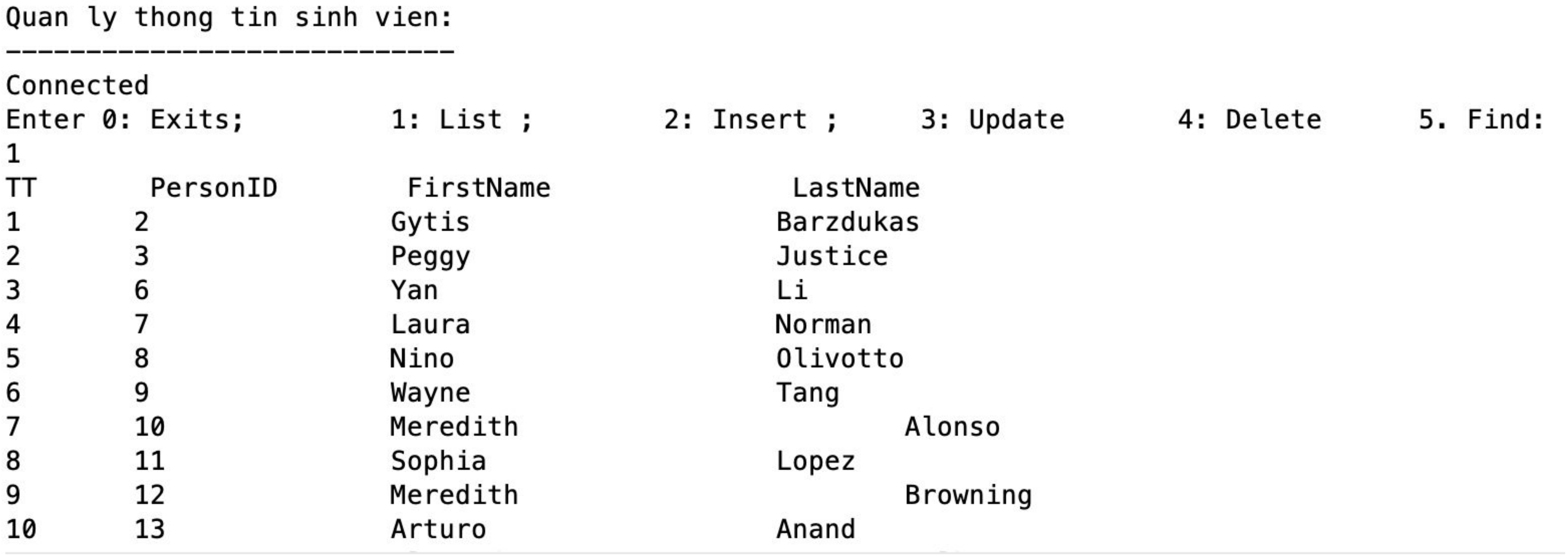
Tìm kiếm

```
public void findStudent(String fullName) throws SQLException {
String query = "SELECT * FROM Person WHERE concat(FirstName, ' ', LastName)  LIKE ?";
PreparedStatement p = this.getConnection().prepareStatement(query);
p.setString(1, "%" + fullName + "%");
ResultSet rs = p.executeQuery();
if (rs != null) {
int i = 1;
while (rs.next()) {
System.out.print(i + " - ");
System.out.println(rs.getString("Lastname") + " " + rs.getString("Firstname"));
i++;
}
} else {
System.out.println("Not found");
}
}
```

Xoá

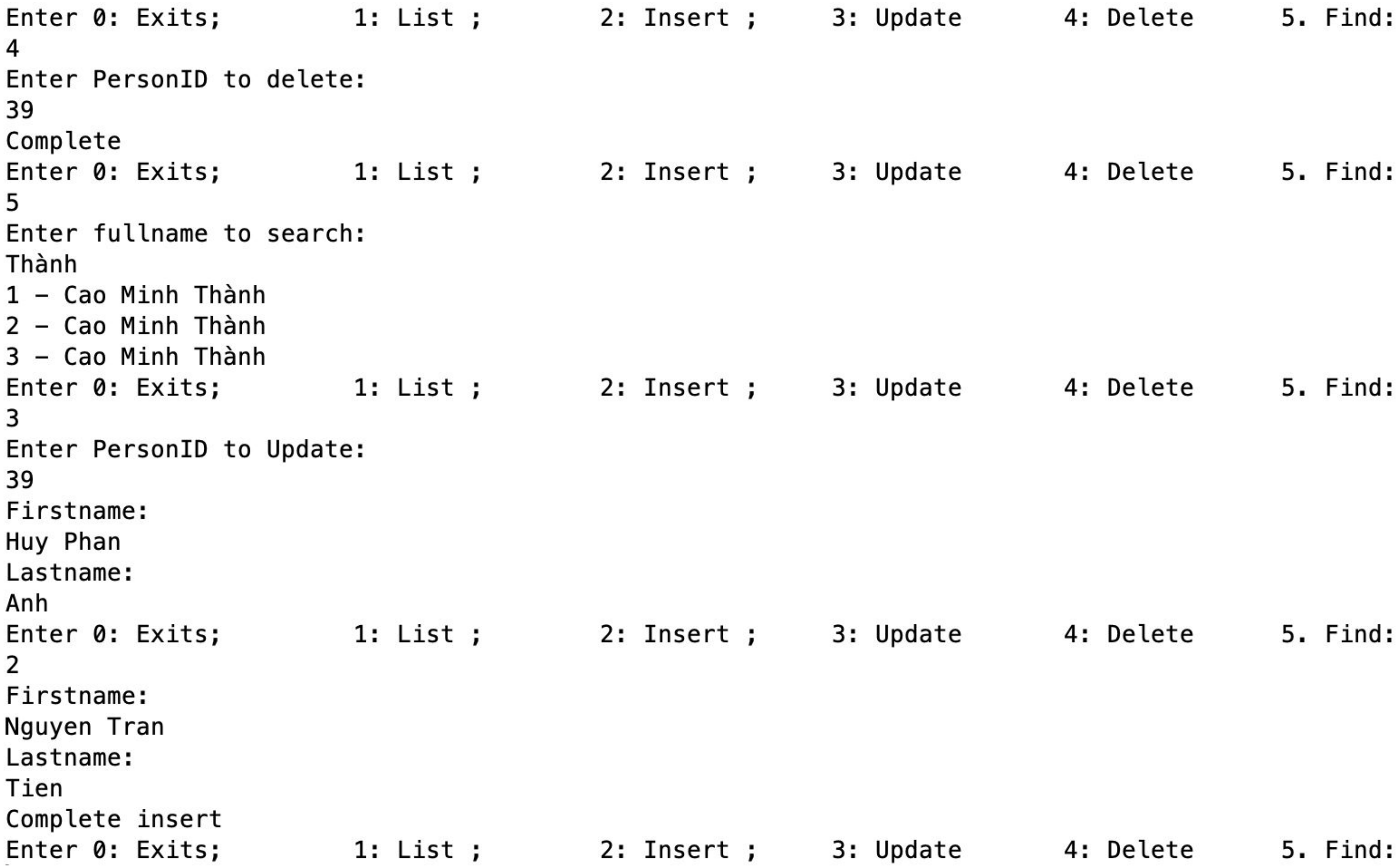
```
public int deleteStudent(int personID) throws SQLException {
String query = "DELETE FROM Person WHERE PersonID = ?";
PreparedStatement p = this.getConnection().prepareStatement(query);
p.setInt(1, personID);
int result = p.executeUpdate();
//this.readStudents();
return result;
}
```

Demo



```
Quan ly thong tin sinh vien:
----------------------------
Connected
Enter 0: Exits;         1: List ;        2: Insert ;     3: Update       4: Delete      5. Find:
1
TT       PersonID        FirstName               LastName
1       2               Gytis                   Barzdukas
2       3               Peggy                   Justice
3       6               Yan                     Li
4       7               Laura                   Norman
5       8               Nino                    Olivotto
6       9               Wayne                   Tang
7       10              Meredith                        Alonso
8       11              Sophia                  Lopez
9       12              Meredith                        Browning
10      13              Arturo                  Anand
```

Demo



```
Enter 0: Exits;         1: List ;        2: Insert ;     3: Update       4: Delete      5. Find:
4
Enter PersonID to delete:
39
Complete
Enter 0: Exits;         1: List ;        2: Insert ;     3: Update       4: Delete      5. Find:
5
Enter fullname to search:
Thành
1 - Cao Minh Thành
2 - Cao Minh Thành
3 - Cao Minh Thành
Enter 0: Exits;         1: List ;        2: Insert ;     3: Update       4: Delete      5. Find:
3
Enter PersonID to Update:
39
Firstname:
Huy Phan
Lastname:
Anh
Enter 0: Exits;         1: List ;        2: Insert ;     3: Update       4: Delete      5. Find:
2
Firstname:
Nguyen Tran
Lastname:
Tien
Complete insert
Enter 0: Exits;         1: List ;        2: Insert ;     3: Update       4: Delete      5. Find:
```