

# **BÀI TẬP LẬP TRÌNH JAVA CƠ BẢN**

**Hệ: Đại học**

## BÀI TẬP THỰC HÀNH TUẦN 1

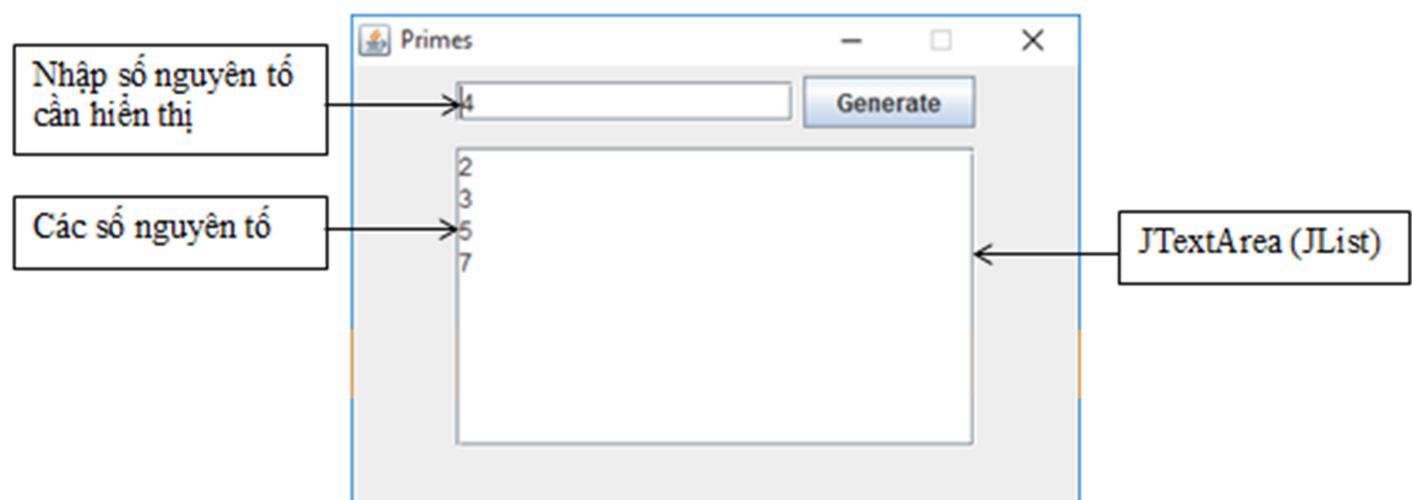
Nội dung thực hành:

- Swing container components
- Event Handling
- Swing Components
- Layout Manager

1. Thiết kế và hiện thực giao diện sau:



2. Viết một chương trình hiển thị N số nguyên tố, trong đó N là một giá trị nhập trong 1 JTextField. Khi người sử dụng click vào JButton “Generate”, N số nguyên tố đầu tiên được ghi vào một JTextArea (JList) như hình bên dưới. Một số nguyên tố là một số nguyên lớn hơn 1, chỉ chia hết cho chính nó và 1.



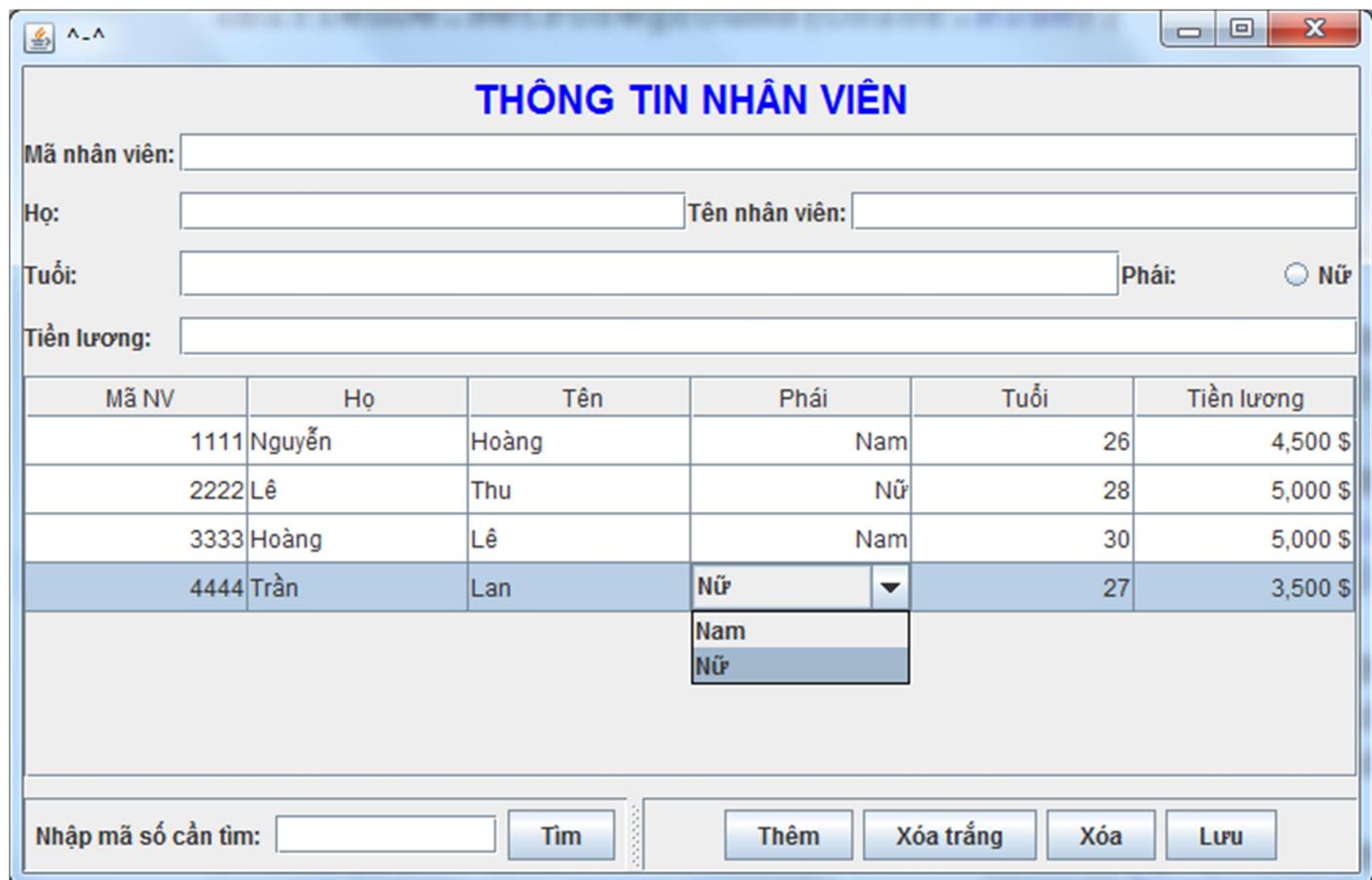
3. Thiết kế và hiện thực giao diện sau (*bài tập làm ở nhà*):

## BÀI TẬP THỰC HÀNH TUẦN 2

Nội dung thực hành: Swing Advanced Controls

- Thực hành đưa dữ liệu lên JTable (JList), xử lý dữ liệu.
- Thực hành TableColumn.
- Thực hành TableCellRenderer.
- Thực hành TableCellEditor

1. Thiết kế và hiện thực giao diện sau:

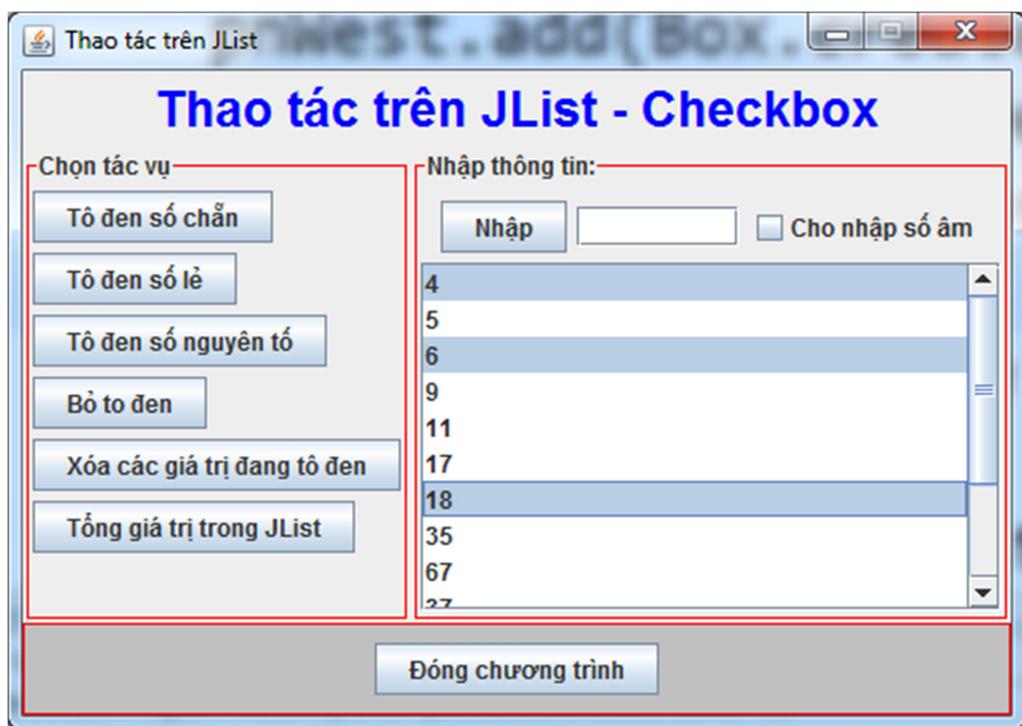


- Chương trình cho phép người sử dụng cập nhật dữ liệu trực tiếp trên bảng. Không cho phép cập nhật mã nhân viên.

Customize: Sao chép project sang 1 project mới. Hiển thị dữ liệu trong JList theo dạng cột.

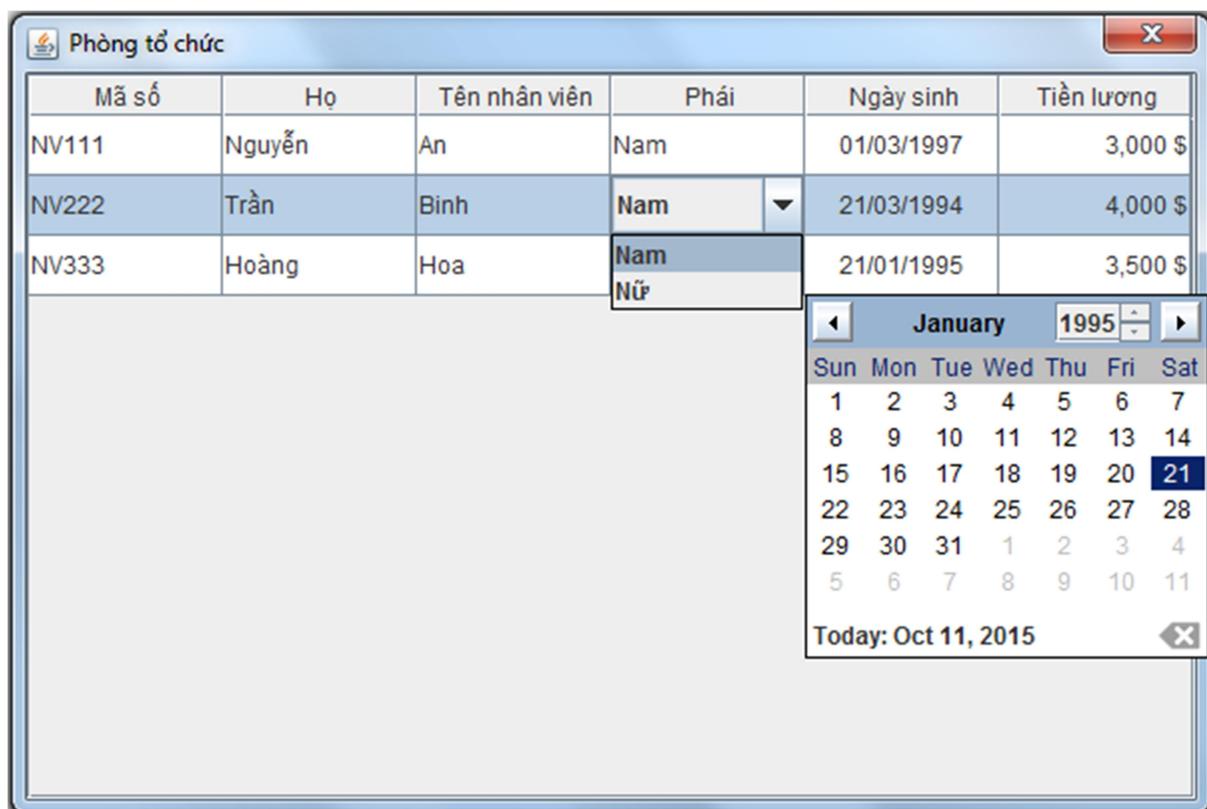
1111	Nguyễn	Hoàng	Nam	26 (Tuổi)	4,500.0 \$
2222	Lê	Thu	Nữ	28 (Tuổi)	5,000.0 \$
3333	Hoàng	Lê	Nam	30 (Tuổi)	5,000.0 \$
4444	Trần	Lan	Nữ	27 (Tuổi)	3,500.0 \$

2. Thiết kế và hiện thực giao diện sau (*bài tập làm ở nhà*):



- ### 3. Thiết kế và hiện thực giao diện sau (*bài tập làm ở nhà*):

- Download JDatePicker : <http://sourceforge.net/projects/jdatepicker/files/Releases/1.3.x/>



## BÀI TẬP THỰC HÀNH TUẦN 3

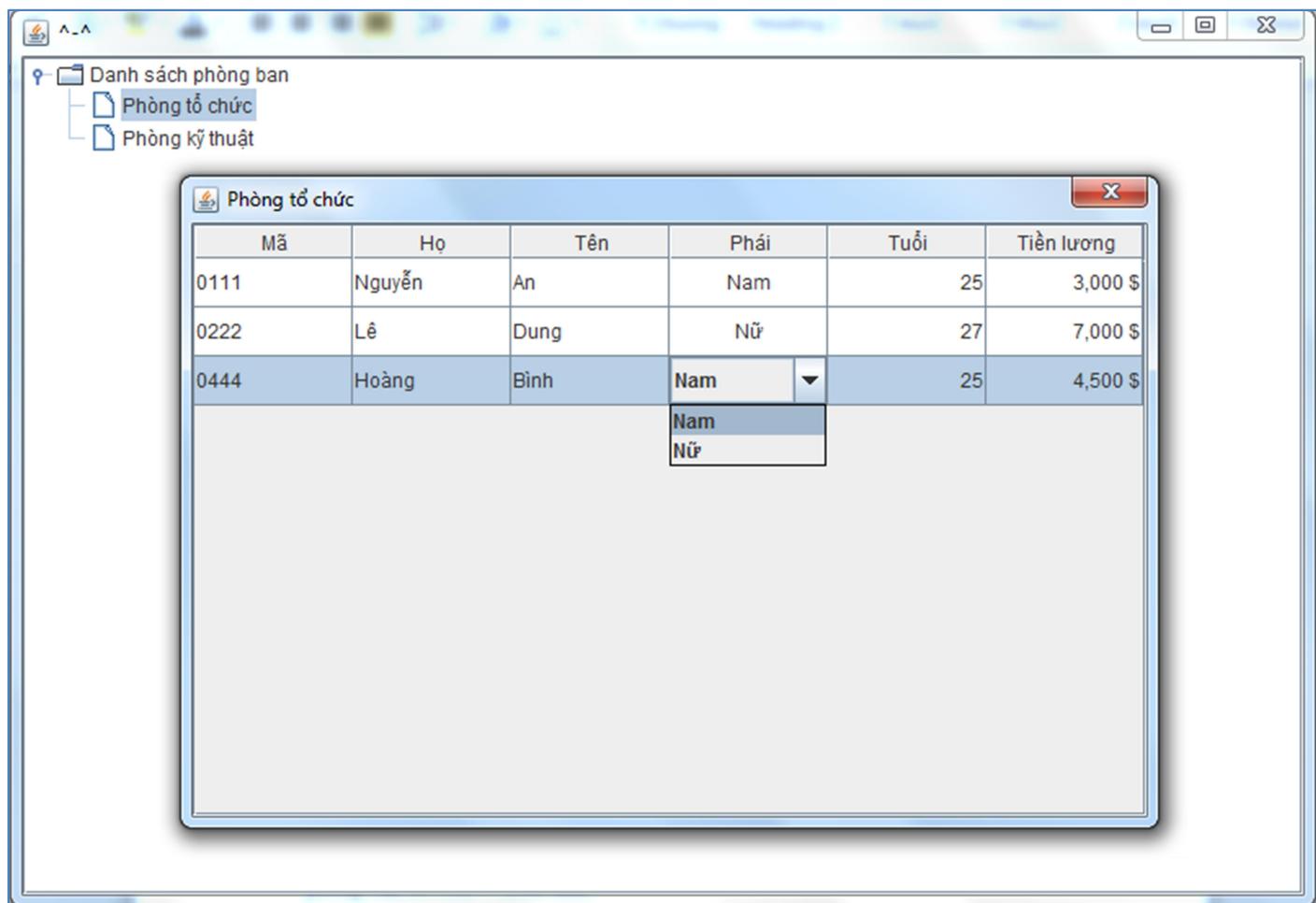
Nội dung thực hành:

- Thực hành JTree
- Thực hành JTabbedPane

Cho class diagram:



1. Thiết kế và hiện thực giao diện sau:



- Form chính liệt kê danh sách các phòng ban.
- Khi người dùng double click vào 1 phòng ban, sẽ liệt kê danh sách các nhân viên của phòng ban đó trên 1 form khác.

2. Thiết kế và hiện thực giao diện sau (*bài tập làm ở nhà*):

Employee!!!

Danh sách nhân viên      Thêm mới nhân viên

Phòng ban:		1. Phòng tổ chức	Phái	Tuổi	Tiền lương
Mã	H	0. Tất cả	Nam	25	3,000 \$
0111	Nguyễn	1. Phòng tổ chức	Nữ	27	7,000 \$
0222	Lê	2. Phòng kỹ thuật	Nam	25	4,500 \$
0444	Hoàng	3. Phòng nhân sự			
		4. Phòng tài vụ			
		Bình			

Employee!!!

Danh sách nhân viên      Thêm mới nhân viên

Mã nhân viên:

Họ:

Tên nhân viên:

Tuổi:

Phái:  Phòng ban:

1. Phòng tổ chức  
2. Phòng kỹ thuật  
3. Phòng nhân sự  
4. Phòng tài vụ

Tiền lương:

## **BÀI TẬP THỰC HÀNH TUẦN 4**

*Nội dung thực hành:*

- Thực hành IO
- Thực hành Regular Expression

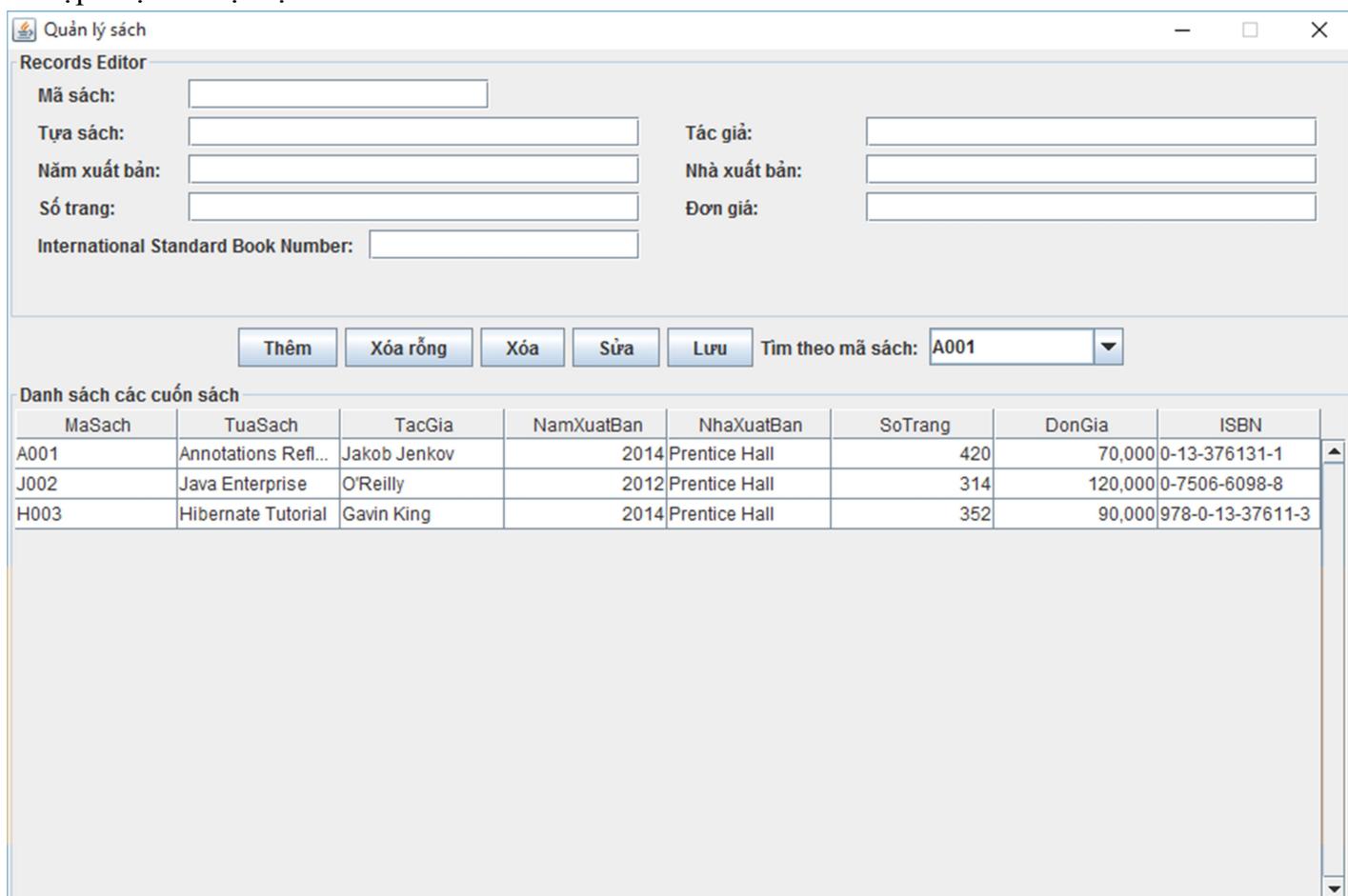
### **Bài 1: Bài toán:**

Cho tập tin *DuLieu.txt*, chứa dữ liệu về thông tin các *cuốn sách* dạng văn bản. Dòng đầu tiên là tiêu đề cột, các dòng kế tiếp là giá trị dữ liệu tương ứng cho từng cột. Mỗi giá trị trong dòng ngắn cách bằng dấu “;”.

MaSach;TuaSach;TacGia;NamXuatBan;NhaXuatBan;SoTrang;DonGia;ISBN  
A001;Annotations Reflection;Jakob Jenkov;2014;Prentice Hall;420;70000;0-13-376131-1  
J002;Java Enterprise;O'Reilly;2012;Prentice Hall;314;120000;0-7506-6098-8  
H003;Hibernate Tutorial;Gavin King;2014;Prentice Hall;352;90000;978-0-13-37611-3

### **Yêu cầu:**

Viết chương trình đọc dữ liệu từ tập tin, đưa lên trên JTable của giao diện, thao tác trên giao diện và cập nhật dữ liệu lại file.



The screenshot shows a Java Swing application window titled "Quản lý sách". The top half contains a "Records Editor" panel with fields for MaSach, TuaSach, TacGia, NamXuatBan, NhaXuatBan, SoTrang, DonGia, and ISBN. Below this is a search bar labeled "Tim theo mã sách: A001". The bottom half displays a "Danh sách các cuốn sách" table with three rows of book data:

MaSach	TuaSach	TacGia	NamXuatBan	NhaXuatBan	SoTrang	DonGia	ISBN
A001	Annotations Refl...	Jakob Jenkov	2014	Prentice Hall	420	70,000	0-13-376131-1
J002	Java Enterprise	O'Reilly	2012	Prentice Hall	314	120,000	0-7506-6098-8
H003	Hibernate Tutorial	Gavin King	2014	Prentice Hall	352	90,000	978-0-13-37611-3

1. Thiết kế giao diện trên.
2. Hiện thực giao diện với các yêu cầu sau:
  - Tạo file *data/DuLieu.txt* trong project.

- Khi chương trình thực thi, sẽ đọc nội dung file đã cho theo dòng rồi tách chuỗi để đưa vào một tập các đối tượng *Sách* (mỗi đối tượng *Sách* có thuộc tính như trên) và đưa vào 1 JTable như hình.
  - Chương trình không cho phép người sử dụng sửa dữ liệu trực tiếp trên JTable. Khi người sử dụng click chuột chọn 1 dòng trên JTable, dữ liệu sẽ hiện thị trên các component.
  - Nút *Sửa*: Chính sửa giá trị dữ liệu trên các component của cuốn sách được chọn (*không được phép cập nhật mã sách*).
  - Nút *Thêm*: Thêm 1 cuốn sách mới.
    - Mã sách không được rỗng, không được trùng.
    - Mã sách phải theo qui ước: **Có ký tự đầu là ký tự chữ HOA, sau đó luôn có 3 ký số.**
    - Tựa sách không được để trống, có thể gồm nhiều từ ngăn cách bởi khoảng trắng. Có thể có các ký số, dấu -, ( và ). Không chứa các ký tự đặc biệt khác.

*Ví dụ: Core Java Volume I - Fundamentals (9th Edition)*

    - Tác giả không được để trống, có thể gồm nhiều từ ngăn cách bởi khoảng trắng. Không chứa ký số hoặc các ký tự đặc biệt khác, ngoại trừ ký tự ‘.
    - Số trang phải là số nguyên dương. Nếu người dùng để trống thì gán giá trị mặc định là 0.
    - Năm xuất bản phải có miền giá trị từ 1900 đến năm hiện tại.
    - Đơn giá phải là số dương. Nếu người dùng để trống thì gán giá trị mặc định là 0.
    - *ISBN các em tham khảo :* <https://vi.wikipedia.org/wiki/ISBN>

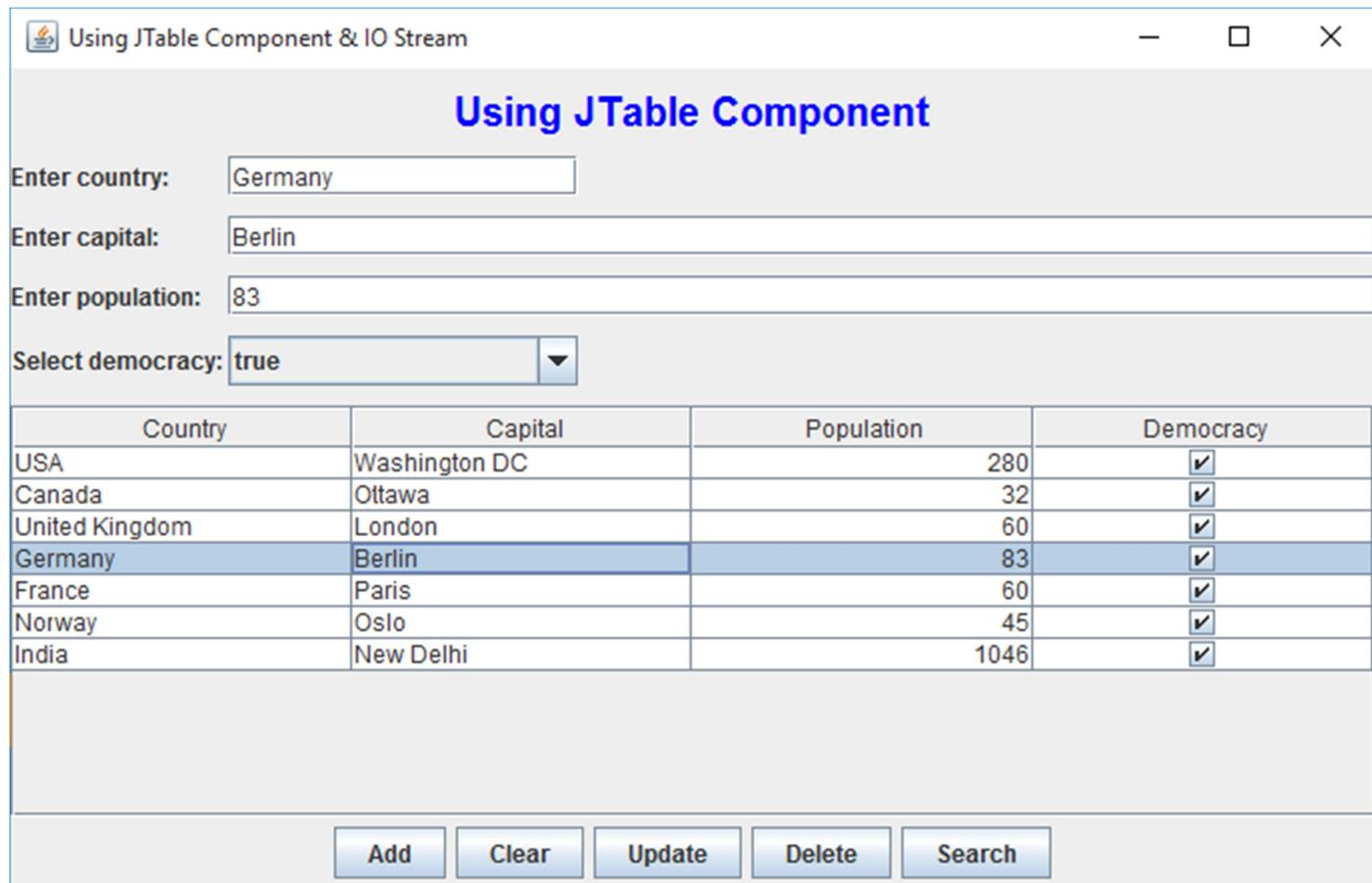
Để giảm độ phức tạp, yêu cầu ISBN có mẫu dạng X-X-X-X (hoặc X-X-X-X-X). Trong đó, X chỉ chứa các ký số, ít nhất là 1 ký số.
  - Nút *Xóa rỗng*: Xóa rỗng nội dung trong các component.
  - Nút *Xóa*: Xóa dòng dữ liệu được chọn. Hỏi nhắc người dùng trước khi xóa.
  - *Chức năng Tìm theo mã sách*: Tìm kiếm cuốn sách dựa theo mã sách. Khi người dùng chọn 1 cuốn sách trong JComboBox, hiển thị thông tin trên các component, tô đen dòng vừa tìm trên JTable và scroll tới nó.
- 

#### *Ghi chú:*

1. *Thao tác Add, Update, Delete phải cập nhật lại file, cập nhật dữ liệu trên các component, hiển thị lại dữ liệu trên JTable, JComboBox.*
2. *Chương trình phải thông báo các message cụ thể nếu người dùng gõ dữ liệu không phù hợp.*

## Bài 2: (Bài tập làm ở nhà)

Viết chương trình thao tác trên giao diện và cập nhật dữ liệu lại file. Sử dụng luồng đối tượng để đọc ghi dữ liệu trên file (*ObjectInputStream / ObjectOutputStream*).



- Khi người dùng click chuột chọn 1 dòng trong JTable, sẽ hiển thị thông tin của dòng được chọn trên các JTextFields.
- Add: Thêm 1 Country vào file
  - o Không được trùng tên nước.
  - o Tên nước, và thủ đô có thể gồm nhiều từ ngăn cách bằng khoảng trắng, không chứa ký số hoặc các ký tự đặc biệt khác.
  - o Dân số phải là 1 số nguyên dương.
- Clear: Xóa rỗng các JTextFields, JComboBox có giá trị mặc định là false.
- Update: Có thể sửa dữ liệu trực tiếp trên các ô của Jtable, hoặc có thể chỉnh sửa dữ liệu trên các component của dòng được chọn.
- Delete: Xóa dòng dữ liệu được chọn. Hỏi nhắc người dùng trước khi xóa.
- Save: Lưu dữ liệu xuống file (*bổ sung vào giao diện*).
- Search: Tìm theo tên country, nhập dữ liệu từ hộp thoại input dialog.

- Tìm thấy nạp vào các JTextField, chỉ hiển thị dòng tìm thấy trên JTable. Nếu người dùng click nút clear, sẽ hiển thị lại toàn bộ thông tin lại trên JTable.
  - Thông báo “Not found” nếu không tìm thấy.
- 

*Ghi chú:*

1. *Thao tác Add, Update, Delete phải cập nhật lại file, cập nhật dữ liệu trên các component, hiển thị lại dữ liệu trên JTable*
2. *Chương trình phải thông báo các message cụ thể nếu người dùng gõ dữ liệu không phù hợp.*

## BÀI TẬP THỰC HÀNH TUẦN 5

Nội dung thực hành

- Thực hành Java Reflection

**Bài 1:** Chuyển từ 1 class bất kỳ sang interface.

Ví dụ, ta có class **Test\_Mixed** như sau:

```
public class Test_Mixed{
    private static final int pi = 32;
    public static final byte b = 32;
    public static final short sh = 32;
    public static final int i = 32;
    public static final long l = 32;
    public final boolean bl = true;
    public static final char c = 'd';
    public final float f = 3.001f;
    public static final double bar = 300.14;
    public static final int[] a = {3, 42, 9};
    public static final double[] d = {3.34, 4.2, 9.7};
    public final float[] listFloat = {3.34f, 4.2f, 9.7f};
    public static final String[] list = {"An", "Lan", "Binh"};
    public static final String s = "testing \n testing \t testing \r
testing \f testing";
    public static final Double od = new Double(4.2);
    public static final Character oc = new Character('d');
    public int x;

    public Test_Mixed() {}
    public Test_Mixed(int x) {
        this.x = x;
    }
    private void privateMethod() {}
    public static void staticMethod() {}
    public void nullMethod() {}
    public int simpleMethod(double d) {
        return (int) d;
    }
    public int complexMethod(int i, int j, double d) {
        return 0;
    }
    public int complexMethod(char c) throws IllegalAccessException,
ArithmaticException {
        if(c == 'd')
            throw new IllegalAccessException("D-exception!");
        if(c == 'x')
            throw new ArithmaticException("A-exception");
        return 0;
    }
}
```

Chuyển sang interface như sau:

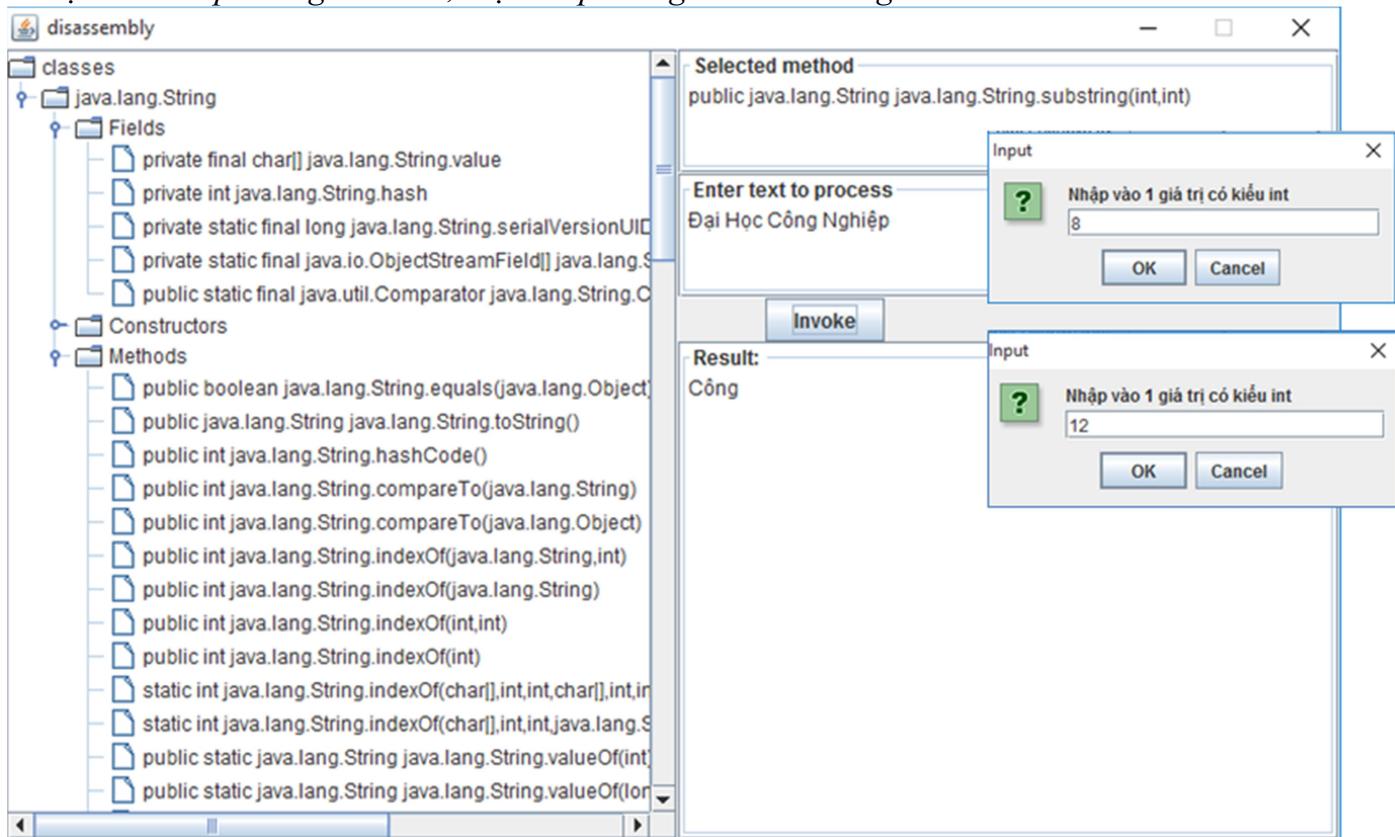
```
public interface ITest_Mixed{
    public static final byte b = 32;
    public static final short sh = 32;
    public static final int i = 32;
    public static final long l = 32;
    public final boolean bl = true;
    public static final char c = 'd';
    public final float f = 3.001f;
    public static final double bar = 300.14;
    public static final int[] a = {3, 42, 9};
    public static final double[] d = {3.34, 4.2, 9.7};
    public final float[] listFloat = {3.34f, 4.2f, 9.7f};
    public static final String[] list = {"An", "Lan", "Binh"};
    public static final String s = "testing \n testing \t testing \r
testing \f testing";
    public static final Double od = new Double(4.2);
    public static final Character oc = new Character('d');
    public int simpleMethod(double arg0);
    public int complexMethod(int arg0, int arg1, double arg2);
    public int complexMethod(char arg0) throws IllegalAccessException,
ArithmaticException;
    public void nullMethod();
}
```

## Bài 2:

Thực thi phương thức bất kỳ của một lớp.

1. Thiết kế giao diện:
  - Thêm bất kỳ class nào vào cây, với nút gốc là “classes”.
  - Mỗi class sẽ chứa 3 nút con gồm các fields, constructors và các methods của class.
  - Đọc cấu trúc của lớp, nạp tương ứng lên cây như trong hình.
2. Khi người dùng chọn bất kỳ 1 phương thức, click nút Invoke, sẽ tự động hiển thị lần lượt các JOptionPane tương ứng với số tham số của phương thức để người dùng nhập giá trị tương ứng với kiểu của từng tham số.  
Nhập đủ tham số, sẽ thực thi phương thức và hiển thị kết quả.

Ví dụ: Thêm lớp String vào tree, thực thi phương thức substring với 2 tham số kiểu int:



## Bài 3: (Bài tập làm ở nhà)

Thay vì dùng JOptionPane để nhập liệu cho các tham số. Ta dùng các JLabel và JTextFields thay thế. Khi người dùng chọn 1 phương thức, sẽ tự động hiển thị số component tương ứng với số tham số của phương thức được chọn.

Người dùng nhập giá trị tương ứng cho từng tham số. Click nút Invoke, thực thi phương thức và hiển thị kết quả.

## BÀI TẬP THỰC HÀNH TUẦN 6

Nội dung thực hành

- Thực hành Annotations

**Bài 1:** Object/Relational Mapping (*mapping java class to table*). Đọc class file, phát sinh tự động câu lệnh SQL để tạo các bảng tạo CSDL tương ứng.

**Primitive types:**

Mapping type	Java type	ANSI SQL Type
integer	int or java.lang.Integer	INTEGER
long	long or java.lang.Long	BIGINT
short	short or java.lang.Short	SMALLINT
float	float or java.lang.Float	FLOAT
double	double or java.lang.Double	DOUBLE PRECISION
big_decimal	java.math.BigDecimal	NUMERIC
character	java.lang.String	CHAR(1)
string	java.lang.String	VARCHAR
byte	byte or java.lang.Byte	TINYINT
boolean	boolean or java.lang.Boolean	BIT
yes/no	boolean or java.lang.Boolean	CHAR(1) ('Y' or 'N')
true/false	boolean or java.lang.Boolean	CHAR(1) ('T' or 'F')

**Date and time types:**

Mapping type	Java type	ANSI SQL Type
date	java.util.Date or java.sql.Date	DATE
time	java.util.Date or java.sql.Time	TIME
timestamp	java.util.Date or java.sql.Timestamp	TIMESTAMP
calendar	java.util.Calendar	TIMESTAMP
calendar_date	java.util.Calendar	DATE

Ví dụ:

Java class

```
public class Employee {
    private int id;
    private String firstName;
    private String lastName;
    private long salary;
    public Employee(int id, String firstName, String lastName, long salary) {
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.salary = salary;
    }
    public int getId() {
        return id;
    }
}
```

```

    }
    public String getFirstName() {
        return firstName;
    }
    public String getLastname() {
        return lastName;
    }
    public long getSalary() {
        return salary;
    }
    public void setSalary(long salary) {
        this.salary = salary;
    }
}

```

### Java class with annotations

```

@Entity
public class Employee {
    @Id
    private int id;
    @Column(nullable = false)
    private String firstName;
    private String lastName;
    private long salary;

    public Employee() {
        this(0, "", "", 01);
    }
    public Employee(int id, String firstName, String lastName, long salary) {
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.salary = salary;
    }
    public int getId() {
        return id;
    }
    public String getFirstName() {
        return firstName;
    }
    public String getLastname() {
        return lastName;
    }
    public long getSalary() {
        return salary;
    }
    public void setSalary(long salary) {
        this.salary = salary;
    }
}

```

RDBMS table:

```
Create table Employee (
    id INTEGER primary key,
    first_name NVARCHAR(40) not null,
    last_name NVARCHAR(40),
    salary BITINT
);
```

**Bài 2:** (*Bài tập làm ở nhà*)

Mở rộng bài 1 cho Association Mappings.

Mapping type
Many-to-One
One-to-One
One-to-Many
Many-to-Many

## BÀI TẬP THỰC HÀNH TUẦN 7

Nội dung thực hành: Thực hành XML

Parsing XML document:

- Dùng SAX Parser
- Dùng DOM Parser

**Bài 1:** Cho tài liệu XML có cấu trúc như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<productList>
    <product id="TV1025">
        <name>TV Led</name>
        <manufacture>Sony</manufacture>
        <description>TV led voi cong nghe</description>
        <supplier>
            <name>Thiên Hòa</name>
            <country>Việt Nam</country>
            <website>www.thienhoa.com</website>
        </supplier>
        <price>500.0</price>
    </product>
    <product id="Phone1002">
        <name>ĐT NOKIA lumia 630</name>
        <manufacture>nokia</manufacture>
        <description>Điện thoại cảm ứng</description>
        <supplier>
            <name>FBT</name>
            <country>Việt Nam</country>
            <website>www.fbt.com</website>
        </supplier>
        <price>800.0</price>
    </product>
    <product id="iPhone5S">
        <name>Điện thoại IPhne 5S</name>
        <manufacture>Apple</manufacture>
        <description>Điện thoại cảm ứng</description>
        <supplier>
            <name>Thế giới di động</name>
            <country>Việt Nam</country>
            <website>www.diDong.com</website>
        </supplier>
        <price>1200.0</price>
    </product>
</productList>
```

1. Tạo file DTD mô tả văn phạm cho tài liệu XML.
2. Tạo file XML Schema mô tả văn phạm cho tài liệu XML.
3. Đọc tập tin XML dùng SAX Parser.
4. Dùng DOM Parser:

- Đọc tài liệu XML, chuyển thành tập các đối tượng product. Mỗi đối tượng product có các thuộc tính như trong tài liệu XML.
- Hiển thị tập các đối tượng lên 1 JTable.
- Khi người dùng chọn một phần tử trong JTable sẽ hiển thị thông tin chi tiết lên các JTextField.
- Thêm 1 product.
- Cập nhật thuộc tính (đã có) của một product.
- Cập nhật giá trị (đã có) của một product.
- Xóa một product.
- Tìm kiếm thông tin của 1 product

*Thêm vào giao diện các nút Thêm, Lưu, Xóa, Sửa và Tìm kiếm với các chức năng tương ứng.*

## 5. Sinh viên thêm vào vài sản phẩm nữa để thử.

- Dùng StAX duyệt qua và in cấu trúc của file XML
- Dùng StAX tìm sản phẩm có giá >600\$.
- Dùng StAX thêm vào 1 sản phẩm mới

## 6. Hiển thị tài liệu XML trên web browser dạng:

Product ID	Product name	Manufacture	Description	Supplier name	Country	Website	Price
Phone1002	DT NOKIA lumia 630	nokia	Điện thoại cảm ứng	FBT	Việt Nam	www.fbt.com	800.0
TV1025	TV Led	Sony	TV led voi cong nghe	Thiên Hòa	Việt Nam	www.thienhoa.com	500.0
iPhone5S	Điện thoại IPnhe 5S	Apple	Điện thoại cảm ứng	Thế giới di động	Việt Nam	www.diDong.com	1200.0

## Bài 2: (Bài tập làm ở nhà)

- Tạo 1 tài liệu XML chứa tập các sản phẩm (*ít nhất phải có 1 sản phẩm*).
  - Mỗi phần tử (element) sản phẩm có các thuộc tính (attribute): Mã sản phẩm (*bắt buộc phải có*).
  - Mỗi phần tử sản phẩm còn có các phần tử con: tên sản phẩm, đơn vị tính, số lượng tồn và đơn giá (*không nhất thiết phải tuân theo thứ tự*).
- Tạo file DTD mô tả văn phạm cho tài liệu XML trên.
- Tạo file XML Schema để mô tả văn phạm cho tài liệu trên.
- Dùng DOM parser:
  - Xóa tất cả Empty text node có trong tài liệu XML.
  - Xóa trong tài liệu XML trên 1 sản phẩm bất kỳ khi biết mã sản phẩm.
  - Cập nhật đơn giá của 1 sản phẩm khi biết mã sản phẩm.
  - Thêm vào tài liệu XML trên 1 sản phẩm bất kỳ.
- Chỉnh sửa văn phạm trên, bổ sung thêm 1 thuộc tính là **trạng thái** của sản phẩm (*tùy chọn – không bắt buộc phải có*).
  - Dùng DOM parser cập nhật trạng thái sản phẩm khi biết mã sản phẩm.

(Kết quả của việc thêm xóa sửa phải được lưu vào file result.xml hoặc xuất ra màn hình để kiểm thử)

Cây hiển thị tài liệu XML – các em dùng phần mềm XMLNotepad để mở tài liệu XML đã tạo để quan sát cây.

Lưu ý: Đây là cây hiển thị thôi, còn cây thực được tạo ra trong bộ nhớ (DOM parser) còn thêm các node là các **empty text node**, điều này là do chúng ta muốn tinh chỉnh để thuận tiện trong việc tạo tài liệu XML vì vậy không thể bắt ép chúng ta phải viết trên 1 dòng duy nhất.

Chúng ta có thể viết 1 phương thức để loại bỏ tất cả các **empty text node** trước khi phân tích tài liệu XML bằng DOM Parser.

The screenshot shows the XMLNotepad interface with two tabs: "Tree View" and "XSL Output".

**Tree View:**

- xml
- #comment
- dssp
  - xmlns:xsi
  - xsi:noNamespaceSchemaLocation
  - #comment
  - sanpham
    - masp
    - dongia
      - #text
    - tensp
      - #text
    - donvitinh
      - #text
    - soluongton
      - #text
  - #comment
  - sanpham
    - masp
    - tensp
      - #text
    - donvitinh
      - #text
    - soluongton
      - #text
  - #comment
  - sanpham
    - masp
    - tensp
      - #text
    - donvitinh
      - #text
    - soluongton
      - #text

version="1.0" encoding="UTF-8"  
Danh sách sản phẩm  
<http://www.w3.org/2001/XMLSchema-instance>  
sanPhamXMSchema.xsd  
sản phẩm 1  
  
TV\_12  
35000  
Tủ vải 1m2  
Cái  
10  
sản phẩm 2  
  
TV\_16  
Tủ vải 1m6  
Cái  
5  
45000  
sản phẩm 3  
  
TN\_6  
Tủ nhựa 6 ngăn  
Cái  
15

### Bài 3: (Bài tập làm ở nhà)

Cho tài liệu XML sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<orderlist>
    <order id="YC106A" orderdate="2016-03-31">
        <lineitem id="111" name="Sữa chua dâu" price="4200" quantity="6" unit="hủ"></lineitem>
        <lineitem id="110" name="Sữa chua nguyên chất" price="4000" quantity="4" unit="hủ"></lineitem>
        <customer>
            <name>Hoàng Lan</name>
            <phone>0903555777</phone>
            <address>12 Lê Lợi Q.Gò Vấp</address>
        </customer>
    </order>

    <order id="YD107D" orderdate="2016-04-01">
        <lineitem id="210" name="Gạo tám thơm" price="19000" quantity="50" unit="kg"></lineitem>
        <lineitem id="310" name="Thịt bò thăn loại I" price="320000" quantity="2" unit="kg"></lineitem>
        <lineitem id="311" name="Gà ta còn sống" price="120000" quantity="5" unit="kg"></lineitem>
        <lineitem id="111" name="Sữa chua dâu" price="4200" quantity="12" unit="hủ"></lineitem>
        <lineitem id="410" name="Giò lụa" price="45000" quantity="1" unit="kg"></lineitem>
        <customer>
            <name>Hòa Hùng</name>
            <phone>083544754</phone>
            <address>12 Lê Lai Q.Gò Vấp</address>
        </customer>
    </order>
</orderlist>
```

1. Tạo file XML schema để mô tả văn phạm cho tài liệu XML theo mô tả sau:

- Danh sách hóa đơn (*orderlist*) có ít nhất 1 hóa đơn (*order*).
  - Hóa đơn gồm các thuộc tính: Mã hóa đơn (*orderid*), ngày lập hóa đơn (*orderdate*)
    - o 1 hóa đơn phải của một khách hàng (*customer*) duy nhất. Thông tin khách hàng gồm: Tên (*name*), điện thoại (*phone*) và địa chỉ (*address*).
    - o 1 hóa đơn có thể mua nhiều sản phẩm nhưng ít nhất phải có 1 sản phẩm.
    - o 1 sản phẩm khi lập hóa đơn cho khách hàng (*lineitem*) gồm các thông tin sau: Mã sản phẩm (*id*), tên sản phẩm (*name*), đơn giá (*price*), số lượng (*quantity*), đơn vị tính (*unit*).
    - o Mã hóa đơn luôn gồm 6 ký tự.
    - o Số lượng phải là số nguyên dương > 0.

2. Dùng SAX Parser để đọc tài liệu XML đã cho và in ra mẫu:

\*\*\*\*\*HÓA ĐƠN BÁN LẺ\*\*\*\*\*

Mã hóa đơn: YC106A

Ngày lập hóa đơn: 2016-03-31

1 111	Sữa chua dâu	4,200	*	6hủ	=	25,200
2 110	Sữa chua nguyên chất	4,000	*	4hủ	=	16,000

Tổng tiền thanh toán: 41,200

Tổng số lượng hàng: 10

Thông tin khách hàng:

Họ tên: Hoàng Lan

Điện thoại: 0903555777

Địa chỉ: 12 Lê Lợi Q.Gò vấp

\*\*\*\*\*HÓA ĐƠN BÁN LẺ\*\*\*\*\*

Mã hóa đơn: YD107D

Ngày lập hóa đơn: 2016-04-01

1 210	Gạo tám thơm	19,000	*	50kg	=	950,000
2 310	Thịt bò thăn loại I	320,000	*	2kg	=	640,000
3 311	Gà ta còn sống	120,000	*	5kg	=	600,000
4 111	Sữa chua dâu	4,200	*	12hủ	=	50,400
5 410	Giò lụa	45,000	*	1kg	=	45,000

Tổng tiền thanh toán: 2,285,400

Tổng số lượng hàng: 70

Thông tin khách hàng:

Họ tên: Hòa Hùng

Điện thoại: 083544754

Địa chỉ: 12 Lê Lai Q.Gò vấp

## BÀI TẬP THỰC HÀNH TUẦN 8

Nội dung thực hành:

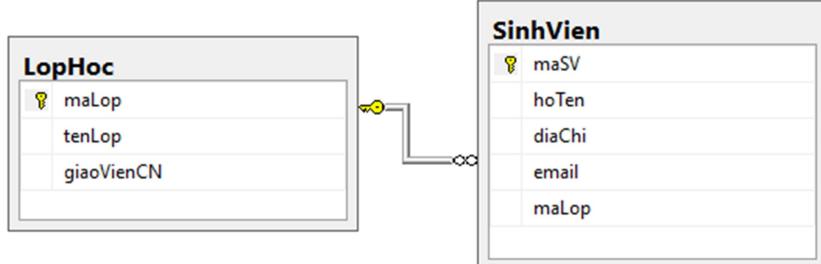
- Kết nối CSDL
- Các câu lệnh xử lý dữ liệu JDBC : CRUD
- JDBC ResultSet

### Bài 1:

Làm các demonraction về kết nối CSDL và thực thi các phương thức của đối tượng Statement, PreparedStatement: executeQuery, executeUpdate, executeBatch

### Bài 2:

Dùng hệ quản trị cơ sở dữ liệu SQL Server, tạo CSDL với các bảng sau:



Tạo 1 project bằng ngôn ngữ lập trình java thực hiện các công việc sau:

- Kết nối với CSDL và hiển thị thông tin lên JFrame như hình sau:

Mã số lớp	Tên lớp	Giáo viên chủ nhiệm
DHTH10A	Đại học tin học 10A	Cô Khánh
DHTH10B	Đại học tin học 10B	Thầy Phúc
DHTH10C	Đại học tin học 10C	Thầy Hải
DHTH10D	Đại học tin học 10D	Cô Minh

- Khi nhấn nút Thêm

Thông tin lớp học

## THÔNG TIN LỚP HỌC

Mã lớp:

Tên lớp:

Giáo viên chủ nhiệm:

← → 3/4 | Hủy Lưu Sửa Xóa

Danh sách lớp học		
Mã số lớp	Tên lớp	Giáo viên chủ nhiệm
DHTH10A	Đại học tin học 10A	Cô Khánh
DHTH10B	Đại học tin học 10B	Thầy Phúc
DHTH10C	Đại học tin học 10C	Thầy Hải
DHTH10D	Đại học tin học 10D	Cô Minh

Xem danh sách sinh viên của lớp hiện tại

- Nhấn Hủy hoặc Lưu, sau khi thực hiện xong tác vụ sẽ khôi phục giao diện trở lại như ban đầu. Tương tự cho nút Sửa.
- Khi nhấn nút “Xem danh sách sinh viên lớp hiện tại”, 1 cửa sổ như sau xuất hiện hiển thị các sinh viên thuộc lớp hiện tại đang được chọn.

Thông tin sinh viên

## THÔNG TIN SINH VIÊN

Mã số sinh viên: 4444

Họ tên: Lê Thanh Tú

Email: tu@gmail.com

Địa chỉ: q tân Bình

← → 1/3 | Thêm Lưu Sửa Xóa

Danh sách sinh viên của lớp : Đại học tin học 10C			
Mã số sinh viên	Họ và tên	Email liên hệ	Địa chỉ
4444	Lê Thanh Tú	tu@gmail.com	q tân Bình
5555	Trần Thị Thu Thủy	thuy@gmail.com	Q1
6666	Trần Thị Lan	lanthi@gmail.com	Q2

### Bài 3: Master – Details Form

Với cơ sở dữ liệu của bài 2, thiết kế và hiện thực form theo như hình sau:

The application window title is "Thong tin lop hoc". The main title is "QUẢN LÝ SINH VIÊN".

Left sidebar (Danh sách lớp học):

- Danh sách lớp học
  - Đại học tin học 10A
  - Đại học tin học 10B
  - Đại học tin học 10C** (highlighted)
  - Đại học tin học 10D

Main content area:

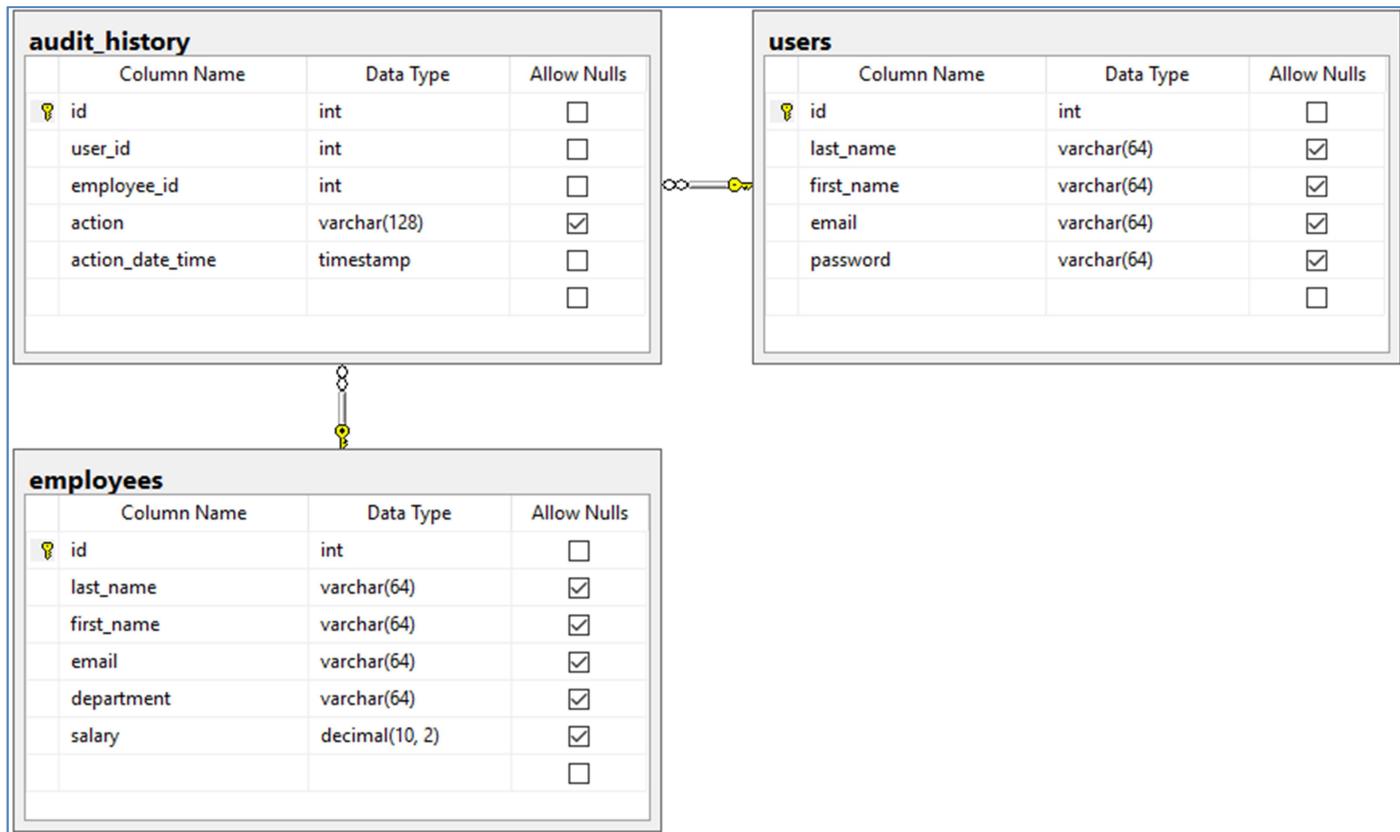
Mã số sinh viên: 4444  
Họ tên: Lê Thanh Tú  
Địa chỉ: q tân Bình  
Email: tu@gmail.com

Buttons: Back, Forward, Page 1/3, Add (Thêm), Save (Lưu), Edit (Sửa), Delete (Xóa).

Mã SV	Họ tên	Địa chỉ	email
4444	Lê Thanh Tú	q tân Bình	tu@gmail.com
5555	Trần Thị Thu Thủy	Q1	thuy@gmail.com
6666	Trần Thị Lan	Q2	lanthi@gmail.com

#### Bài 4: (Bài tập về nhà)

- Hệ quản trị CSDL SQL Server: Chuẩn bị CSDL theo mô hình sau:



- Nhập liệu:

```

INSERT INTO employees VALUES (1,'Doe','John','john.doe@foo.com', 'HR', 55000.00);
INSERT INTO employees VALUES (2,'Public','Mary','mary.public@foo.com', 'Engineering', 75000.00);
INSERT INTO employees VALUES (3,'Queue','Susan','susan.queue@foo.com', 'Legal', 130000.00);

INSERT INTO employees VALUES (4,'Williams','David','david.williams@foo.com', 'HR', 120000.00);
INSERT INTO employees VALUES (5,'Johnson','Lisa','lisa.johnson@foo.com', 'Engineering', 50000.00);
INSERT INTO employees VALUES (6,'Smith','Paul','paul.smith@foo.com', 'Legal', 100000.00);

INSERT INTO employees VALUES (7,'Adams','Carl','carl.adams@foo.com', 'HR', 50000.00);
INSERT INTO employees VALUES (8,'Brown','Bill','bill.brown@foo.com', 'Engineering', 50000.00);
INSERT INTO employees VALUES (9,'Thomas','Susan','susan.thomas@foo.com', 'Legal', 80000.00);

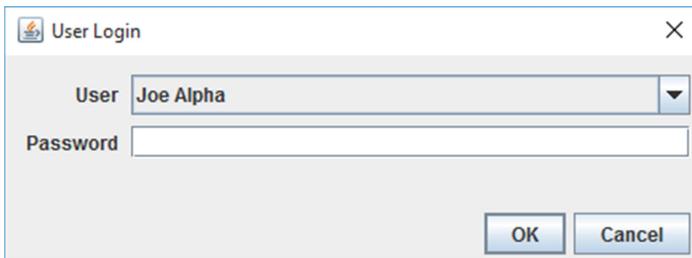
INSERT INTO employees VALUES (10,'Davis','John','john.davis@foo.com', 'HR', 45000.00);
INSERT INTO employees VALUES (11,'Fowler','Mary','mary.fowler@foo.com', 'Engineering', 65000.00);
INSERT INTO employees VALUES (12,'Waters','David','david.waters@foo.com', 'Legal', 90000.00);

INSERT INTO users (id,last_name,first_name,email) VALUES (1,'Alpha','Joe','joe.alpha@foo.com');
INSERT INTO users (id,last_name,first_name,email) VALUES (2,'Beta','Jane','jane.beta@foo.com');
INSERT INTO users (id,last_name,first_name,email) VALUES (3,'Zeta','Becky','becky.zeta@foo.com');

```

Tạo 1 project bằng ngôn ngữ lập trình java thực hiện các công việc sau:

- Khi chạy chương trình: yêu cầu xác định username và password



- Sau khi đăng nhập thành công:

A screenshot of a window titled "Employee Search App". The title bar shows "Logged In: Joe Alpha". Below it is a search interface with a "Enter last name" input field and a "Search" button. A data grid displays employee information with columns: Last Name, First Name, Email, and Salary. The data is as follows:

Last Name	First Name	Email	Salary
Adams	Carl	carl.adams@foo.com	50000.00
Brown	Bill	bill.brown@foo.com	50000.00
Davis	John	john.davis@foo.com	45000.00
Doe	John	john.doe@foo.com	55000.00
Fowler	Mary	mary.fowler@foo.com	65000.00
Johnson	Lisa	lisa.johnson@foo.com	50000.00
Public	Mary	mary.public@foo.com	75000.00
Queue	Susan	susan.queue@foo.com	130000.00

At the bottom are three buttons: "Add Employee", "Update Employee", and "View History".

- Với nút “Search”:

A screenshot of the "Employee Search App" window. The search input field contains "Fowler" and the "Search" button is highlighted. The data grid shows one result:

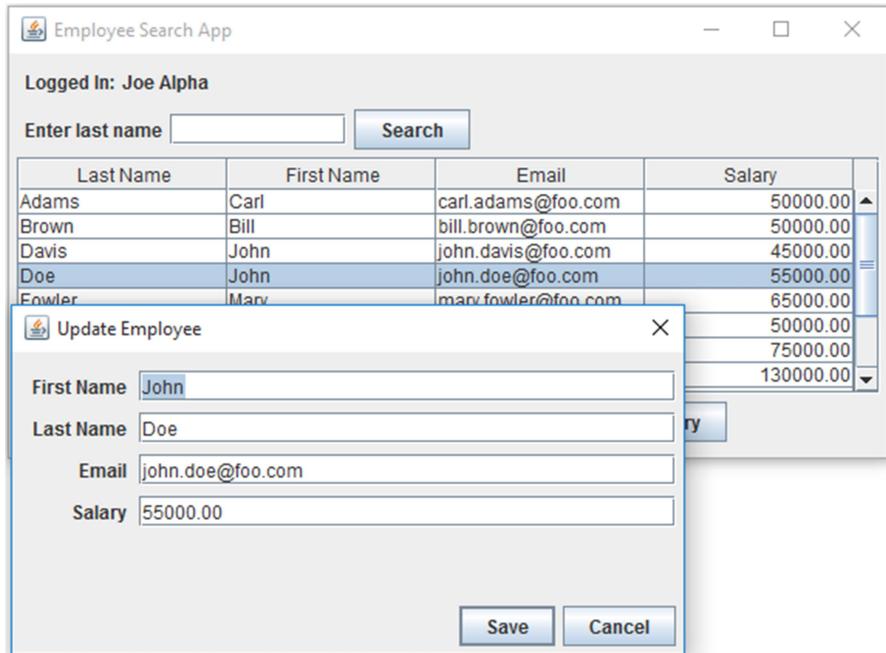
Last Name	First Name	Email	Salary
Fowler	Mary	mary.fowler@foo.com	65000.00

At the bottom are three buttons: "Add Employee", "Update Employee", and "View History".

- Khi click nút “Add Employee”:

A screenshot of a dialog box titled "Add Employee". It has four input fields: "First Name", "Last Name", "Email", and "Salary". At the bottom are "Save" and "Cancel" buttons.

- Khi click nút “Update Employee”:



- Khi click nút “View History”:



## BÀI TẬP THỰC HÀNH TUẦN 9

Nội dung thực hành:

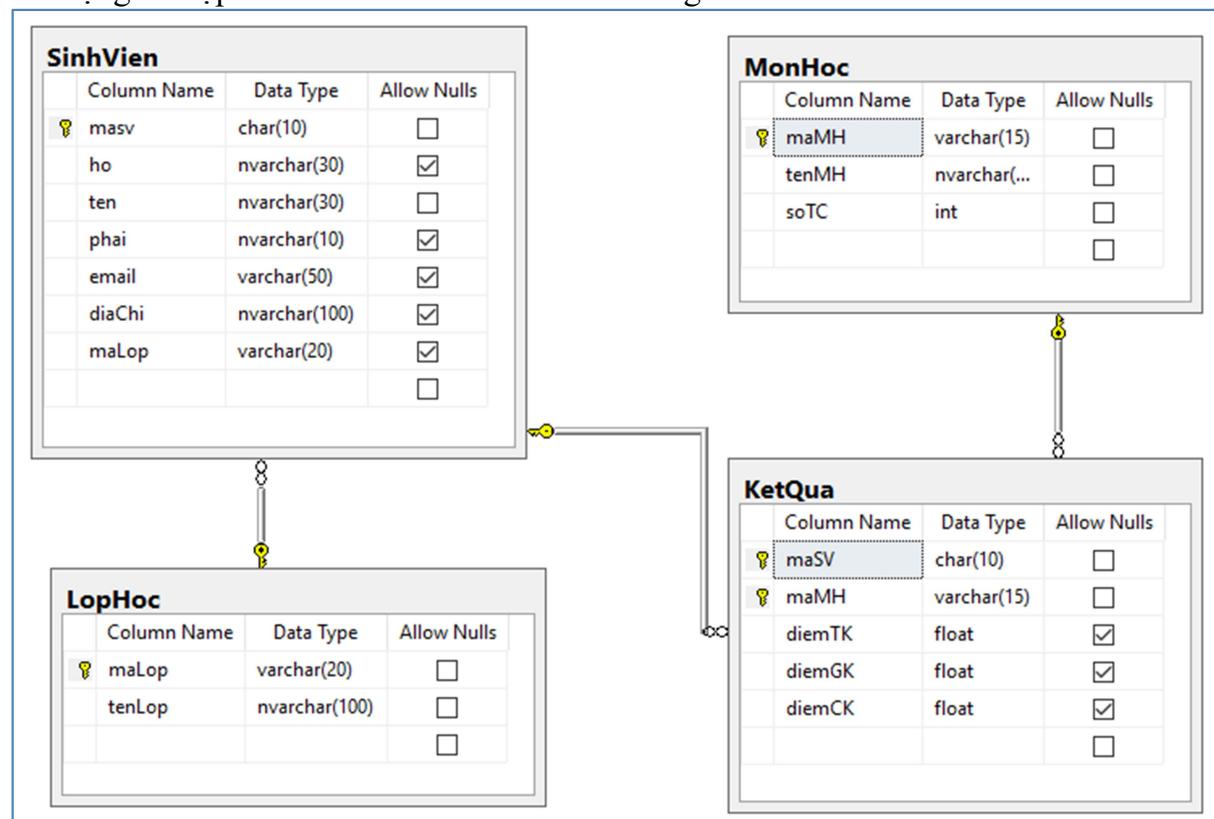
- Swing Menu Components
- Using Callable Statements
- Call Stored Procedures that take parameters.
  - o No parameter
  - o IN parameters
  - o INOUT parameters
  - o OUT parameters
  - o Return a result set
- Transaction

### Bài 1:

- Làm lại bài tập 2 tuần 8: Bằng cách chuyển tất cả các câu SQL sang thủ tục hoặc hàm.
- Đối với nút sửa: Cho phép sửa thông tin 1 đối tượng đã tồn tại, một đối tượng có thể được sửa thông tin nhiều lần. Khi sửa thông tin của đối tượng thì phải lưu trữ mẫu tin cũ trước khi sửa vào bảng Backup nào đó. Bảng Backup có khóa là id (*kiểu int, tự động tăng*) và các thuộc tính của đối tượng. *Lưu ý: Hai công việc phải thực hiện thành công. Nếu 1 trong 2 công việc trên không thành công thì không thực hiện công việc nào cả.*

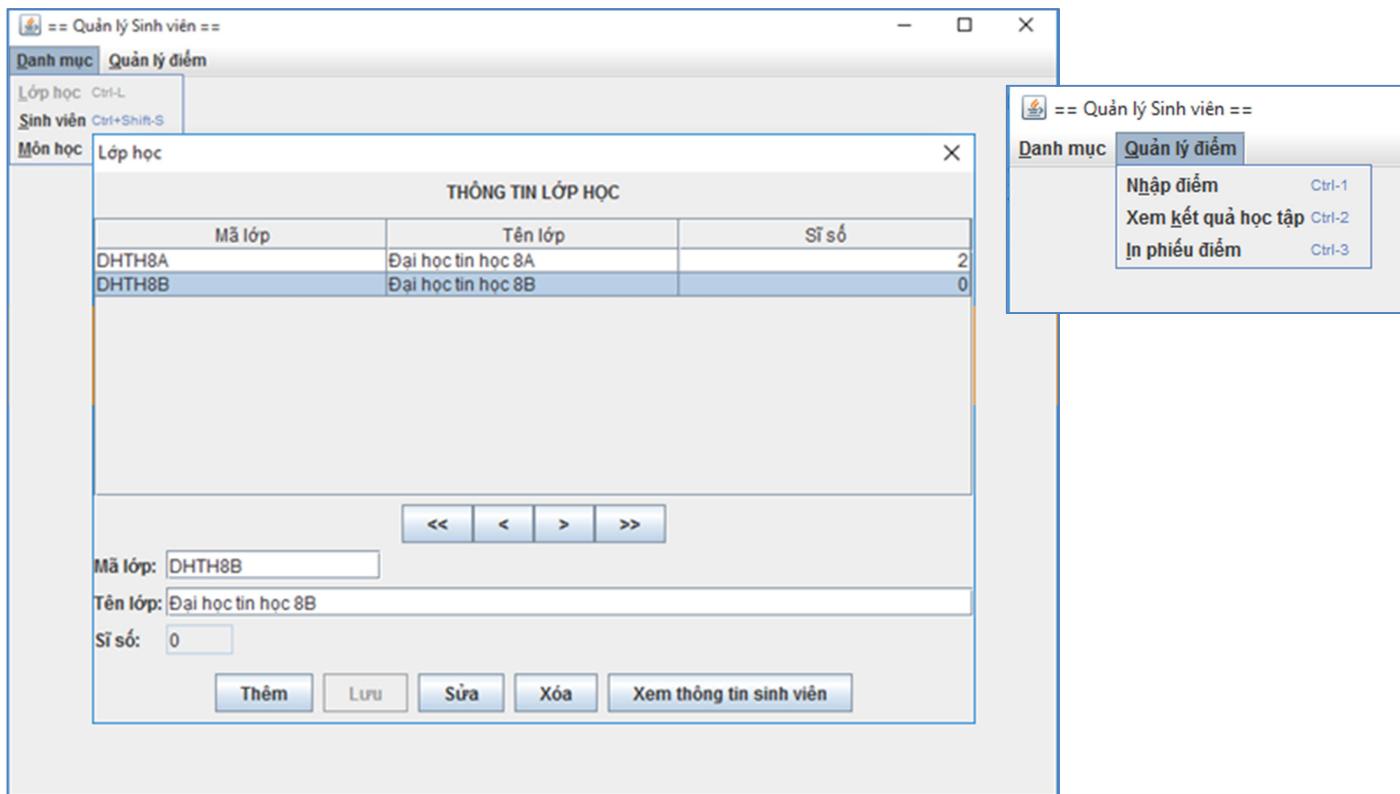
### Bài 2: (Bài tập về nhà)

Mở rộng bài tập 2 của tuần 8. CSDL với các bảng sau:



Tạo 1 project bằng ngôn ngữ lập trình java thực hiện các công việc sau:

- Tạo kết nối với CSDL khi JFrame chính được nạp lên.
- Đóng kết nối với CSDL khi người dùng đóng JFrame này.
- Java Application, với giao diện chính sau:



## **BÀI TẬP THỰC HÀNH TUẦN 10**

*Nội dung thực hành:*

- Using Metadata

### **Bài 1:**

Viết chương trình cho phép kết nối vào 1 database của MS SQL server sau đó lấy danh sách các bảng hiển thị lên 1 JTree. Khi người dùng chọn 1 phần tử trên JTree: nếu là Table thì hiển thị cấu trúc của nó đồng thời hiển thị dữ liệu. Nếu người dùng chọn 1 StoredProcedure thì hiển thị nội dung của câu sql tạo storedproc này. Viết 1 Form cho phép người dùng gõ vào 1 câu SQL và cho thực thi câu SQL này.

## Bài 2: Làm bài kiểm tra thực hành mẫu.

Họ, Tên SV: .....Mã số SV: .....Kí tên:.....Số máy/ Phòng máy: .....

1. Sinh viên điền đầy đủ thông tin, để lại đề thi dưới bàn phím.
2. Sinh viên chỉ được phép dùng tài liệu đã in ra giấy có ghi rõ họ tên.
3. Tạo 1 folder có tên là [họ tên sinh viên]\_[mã số sinh viên] trong ổ đĩa T:. Dùng java IDE, tạo project trong thư mục trên. Tên project **bắt buộc** theo qui ước sau: [họ tên sinh viên]\_[mã số sinh viên]\_[số máy].

Nếu sinh viên không tuân thủ 1 trong các qui định trên thì bài làm của sinh viên bị điểm không.

**Bài toán:** Xử lý *chức năng quản lý các thí sinh* trong một chương trình đào tạo. Mỗi thí sinh có số báo danh là duy nhất, xác định thông tin của 1 thí sinh. Điểm thi lý thuyết và điểm thi thực hành có miền giá trị từ 0 → 10.

- I. Tạo 1 CSDL có tên là tên sinh viên. Tạo cấu trúc bảng cho bên dưới, nhập ít nhất 5 mẫu tin vào bảng này, lưu script vừa tạo vào thư mục sql trong cùng project:

ThiSinh		
Column Name	Data Type	Allow Nulls
soBaoDanh	varchar(10)	<input type="checkbox"/>
hoTen	nvarchar(100)	<input type="checkbox"/>
dienThoai	varchar(50)	<input checked="" type="checkbox"/>
diemLyThuyet	float	<input checked="" type="checkbox"/>
diemThucHanh	float	<input checked="" type="checkbox"/>

- II. Thiết kế giao diện, kết nối với CSDL và hiện thực chương trình

1. (3đ) Khi chương trình chạy, sẽ nạp các thí sinh trong CSDL lên table, giao diện như sau:

The screenshot shows a Windows application window titled "Quản lý điểm". The main title is "THÔNG TIN THÍ SINH". The interface includes input fields for "Số báo danh", "Họ tên", "Điện thoại", "Điểm lý thuyết", and "Điểm thực hành". Below the form is a table displaying student information:

Số báo danh	Họ tên	Điện thoại	Điểm lý thuyết	Điểm thực hành	Điểm trung bình
ACAD.1234	Lê Hùng	0903.456789	5	9	7
ACAD.2345	Lê Nga	0903.345678	10	9	9.5
ACAD.3456	Trần Bình	0903.345678	6	10	8
THA.1234	Nguyễn Thị Hoa	0903.123456	7	8	7.5
THB.1234	Trần Văn Hùng	0903.654321	4	8	6

2. (2đ) Thêm 1 đối tượng Thí sinh mới. Khi click nút “Thêm”, giao diện sẽ như sau:

Quản lý điểm

## THÔNG TIN THÍ SINH

Số báo danh:	<input type="text"/>				
Họ tên:	<input type="text"/>				
Điện thoại:	<input type="text"/>				
Điểm lý thuyết:	<input type="text"/>				
Điểm thực hành:	<input type="text"/>				
<input type="button" value="Hủy"/> <input type="button" value="Lưu"/> <input type="button" value="Sửa"/> <input type="button" value="Xóa"/>					
Số báo danh	Họ tên	Điện thoại	Điểm lý thuyết	Điểm thực hành	Điểm trung bình
ACAD.1234	Lê Hùng	0903.456789	5	9	7

Nhập thông tin một đối tượng Thí sinh, click nút “**Lưu**” sẽ thêm thí sinh vào CSDL.

- (3đ) Sửa thông tin 1 đối tượng thí sinh đã tồn tại, một thí sinh có thể sửa thông tin nhiều lần. Khi click nút “**Sửa**”, giao diện sẽ như sau:

Quản lý điểm

## THÔNG TIN THÍ SINH

Số báo danh:	ACAD.1234				
Họ tên:	Lê Hùng				
Điện thoại:	0903.456789				
Điểm lý thuyết:	5.0				
Điểm thực hành:	9.0				
<input type="button" value="Thêm"/> <input type="button" value="Lưu"/> <input type="button" value="Hủy"/> <input type="button" value="Xóa"/>					
Số báo danh	Họ tên	Điện thoại	Điểm lý thuyết	Điểm thực hành	Điểm trung bình
ACAD.1234	Lê Hùng	0903.456789	5	9	7

Nhập thông tin cần sửa. Click nút “**Lưu**” cập nhật thông tin của thí sinh vào CSDL, và phải lưu trữ mẫu tin cũ trước khi sửa vào bảng Backup. Bảng Backup có khóa là id (*kiểu int, tự động tăng*) và các thuộc tính của thí sinh. *Lưu ý: Hai công việc phải thực hiện thành công. Nếu 1 trong 2 công việc trên không thành công thì không thực hiện công việc nào cả.*

Lưu ý cho câu 2 và câu 3: Click hủy hoặc lưu thành công sẽ khôi phục giao diện như trước.

- (1đ) Khi người dùng chọn 1 dòng trên table, nội dung của dòng đang chọn sẽ hiện thị trên các textfield.
- (1đ) Xóa 1 đối tượng thí sinh đã tồn tại. Khi click nút “**Xóa**”, sẽ xóa trong CSDL dòng dữ liệu đang chọn trong table.

Ghi chú:

- Điểm trung bình =  $(Điểm lý thuyết + Điểm thực hành) / 2$
- Mọi tác vụ sau khi thực hiện thành công, phải cập nhật dữ liệu trên các textfield, hiển thị lại dữ liệu trên table.
- Chương trình phải kiểm tra tính hợp lệ của dữ liệu, thông báo các message cụ thể.
- Câu 3: Nếu chỉ cập nhật được mà không backup được thì điểm tối đa là 1.