

Chương 1. Căn bản về ngôn ngữ lập trình Java

- 1.1. Dùng Netbeans (hoặc Eclipse) soạn thảo và thực thi chương trình HelloWorld dùng để hiển thị ra màn hình câu “Hello Java”.
- 1.2. Thiết lập các biến môi trường để có thể biên dịch và thực thi chương trình Java bằng cơ chế dòng lệnh. Dùng một trình soạn thảo văn bản bất kỳ để biên soạn chương trình Hello Java ở trên, biên dịch và thực thi chương trình bằng cơ chế dòng lệnh.
- 1.3. Nhập vào 2 số nguyên, in ra màn hình số nhỏ hơn và số lớn hơn.
- 1.4. Nhập vào 3 số nguyên, in ra màn hình số nhỏ nhất và số lớn nhất.
- 1.5. Giải và biện luận phương trình bậc 2 ($ax^2 + bx + c = 0$).
- 1.6. Nhập vào một số nguyên là năm, cho biết năm đó có là năm nhuận hay không. Biết rằng: năm nhuận là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100.
- 1.7. Nhập vào một số nguyên là tháng, cho biết tháng đó có bao nhiêu ngày (*nếu là tháng 2 thì yêu cầu nhập thêm năm*).
- 1.8. Nhập vào một số thực là điểm ở thang điểm 10, cho biết giá trị ở thang điểm chữ ABCDF. Biết rằng: 8.5 \rightarrow 10: A, 7.0 \rightarrow 8.4: B, 5.5 \rightarrow 6.9: C, 4.0 \rightarrow 5.4: D, dưới 4: F.
- 1.9. Nhập vào chỉ số điện cũ và chỉ số điện mới. Tính tiền điện phải trả trong tháng, dữ liệu nhập phải đảm bảo chỉ số điện cũ \leq chỉ số điện mới. Cách thức tính tiền điện theo quy định sau:
 - Bậc 1: cho các kWh từ 0 đến 50: 1480 VNĐ/kw;
 - Bậc 2: cho các kWh từ 51 đến 100: 1533 VNĐ/kw;
 - Bậc 3: cho các kWh từ 101 đến 200: 1786 VNĐ/kw;
 - Bậc 4: cho các kWh từ 201 đến 300: 2242 VNĐ/kw;
 - Bậc 5: cho các kWh từ 301 trở lên: 2503 VNĐ/kw;
- 1.10. Kiểm tra xem ngày (*gồm 3 thành phần là ngày, tháng, năm*) cho trước có phải là một ngày hợp lệ. Trong đó, một ngày được gọi là hợp lệ khi:
 - Năm: số nguyên lớn hơn 0;
 - Tháng: từ 1 đến 12;
 - Ngày từ 1 đến số ngày tối đa của tháng – năm tương ứng.
- 1.11. Nhập vào ngày, tháng, năm. Cho biết ngày kế tiếp và ngày kế trước của ngày đó.
- 1.12. Nhập số nguyên dương n. Đếm số lượng các ước số của số n.
- 1.13. Nhập số nguyên dương n. Liệt kê các ước số lẻ của n.
- 1.14. Nhập vào số nguyên dương n. Cho biết n có phải là số nguyên tố hay không?
- 1.15. Nhập vào các số thực khác 0 từ bàn phím. Tính tổng các số đã nhập và in ra màn hình. Giả sử quá trình nhập sẽ dừng khi người dùng nhập số 0.
- 1.16. Nhập số nguyên dương n. Đếm số lượng các chữ số lẻ của n.
- 1.17. Nhập số nguyên dương n. Tính tổng số lượng các chữ số chẵn của n.
- 1.18. Nhập 2 số nguyên dương m, n. Xuất ra hình chữ nhật đặc có kích thước m x n dấu *. Ví dụ: m = 4, n = 5.

```
* * * * *
* * * * *
* * * * *
* * * * *
```

1.19. Nhập 2 số nguyên dương m, n . Xuất ra hình chữ nhật rỗng có kích thước $m \times n$ dấu *. Ví dụ:
 $m = 4, n = 5$.

```

* * * * *
*       *
*       *
* * * * *

```

1.20. Nhập số nguyên dương h là chiều cao. Xuất ra màn hình tam giác cân đặc có chiều cao h . Ví dụ $h = 4$.

```

      *
     * *
    * * *
   * * * *
  * * * * *

```

1.21. Nhập số nguyên dương h là chiều cao. Xuất ra màn hình tam giác cân rỗng có chiều cao h . Ví dụ $h = 4$.

```

      *
     *   *
    *     *
   *       *
  *         *

```

1.22. Viết chương trình tạo mảng một chiều tối đa 500 phần tử (giá trị mỗi phần tử là số nguyên từ -10^6 đến 10^6) và cho phép người dùng chọn thực hiện các chức năng bên dưới, đến khi nào người dùng chọn 0 thì thoát chương trình:

- 1. Nhập giá trị cho n phần tử mảng từ bàn phím ($n \leq 500$);
- 2. Phát sinh giá trị ngẫu nhiên từ -199 đến 199 cho n phần tử mảng ($n \leq 500$);
- 3. Xuất mảng ra màn hình;
- 4. Liệt kê các giá trị âm có trong mảng;
- 5. Liệt kê các số nguyên tố có trong mảng;
- 6. Liệt kê các phần tử có giá trị nằm trong đoạn $[a, b]$ cho trước;
- 7. Tính tổng giá trị các phần tử là số nguyên tố;
- 8. Tính trung bình cộng của các phần tử dương có trong mảng;
- 9. Đếm số phần tử có giá trị lớn hơn x cho trước;
- 10. Đếm số phần tử có giá trị là số nguyên tố trong mảng;
- 11. Kiểm tra mảng có phải là mảng chứa toàn số nguyên tố;
- 12. Kiểm tra mảng có phải là mảng tăng dần;
- 13. Tìm giá trị lớn nhất trong mảng;
- 14. Tìm giá trị nhỏ nhất trong mảng;
- 15. Tìm số âm lớn nhất trong mảng;
- 16. Đảo ngược mảng.

1.23. Viết chương trình tạo một mảng hai chiều (ma trận) tối đa 50×100 phần tử (giá trị mỗi phần tử là số nguyên từ -10^6 đến 10^6) và thực hiện các yêu cầu sau:

- 1. Nhập giá trị cho $m \times n$ phần tử của mảng từ bàn phím ($m \leq 50, n \leq 100$);
- 2. Phát sinh giá trị ngẫu nhiên từ -99 đến 99 cho $m \times n$ phần tử của mảng ($n \leq 100$);
- 3. Xuất giá trị các phần tử ra màn hình;
- 4. Tính tổng giá trị các số lẻ;
- 5. Tính tổng các giá trị dương trên dòng k cho trước;
- 6. Đếm số lượng số dương;

- 7. Đếm số lượng số nguyên tố trên một dòng của ma trận;
- 8. Kiểm tra ma trận có tồn tại số âm hay không;
- 9. Kiểm tra các phần tử trên dòng k cho trước có tăng dần từ trái qua phải hay không;
- 10. Liệt kê dòng có chứa số nguyên tố;
- 11. Liệt kê dòng chứa toàn giá trị chẵn.

1.24. Cài đặt lớp **Diem2D (Điểm trong không gian 2 chiều) gồm:**

- Thuộc tính: x, y là số nguyên.
- Các phương thức bao gồm :
 - + Hàm khởi tạo mặc định: `Diem();`
 - + Hàm khởi tạo có 2 tham số: `Diem(int, int);`
 - + Nhập tọa độ cho điểm từ bàn phím `void nhapDiem ();`
 - + Hiện thị ra màn hình tọa độ điểm theo dạng (x,y): `void hienThi();`
 - + Dời điểm đi 1 độ dời (dx, dy): `void doiDiem(int dx, int dy);`
 - + Lấy ra giá trị hoành độ của điểm: `int giaTriX();`
 - + Lấy ra giá trị tung độ của điểm: `int giaiTriY();`
 - + Tính khoảng cách từ điểm đó đến gốc tọa độ: `float khoangCach();`
 - + Tính khoảng cách từ điểm đó đến 1 điểm khác: `float khoangCach(Diem d);`
- **Viết lớp **SDDiem** có chứa hàm **main()** khai thác lớp vừa định nghĩa:**
 - + Tạo ra điểm A tọa độ (3,4). Hiện thị tọa độ điểm A ra màn hình.
 - + Tạo ra điểm B với giá trị nhập từ bàn phím. Hiện thị tọa độ điểm B ra màn hình.
 - + Tạo ra điểm C đối xứng với điểm B qua gốc tọa độ. Hiện thị tọa độ điểm C ra màn hình.
 - + Hiện thị ra màn hình khoảng cách từ điểm B đến tâm O.
 - + Hiện thị ra màn hình khoảng cách từ điểm A đến điểm B.

1.25. Cài đặt lớp **PhanSo (Phân số) gồm:**

- Các thuộc tính: tử số và mẫu số có kiểu số nguyên.
- Các hàm xây dựng gồm:
 - + Hàm khởi tạo mặc định:
 - + Hàm khởi tạo gồm 2 tham số:
- Các phương thức khác bao gồm:
 - + Hàm nhập giá trị cho 1 phân số. Nếu phân số vừa nhập có mẫu số = 0 thì yêu cầu nhập lại.
 - + Hàm hiện thị phân số theo dạng tu/mau hoặc -tu/mau. Yêu cầu: nếu tử số =0 thì chỉ in ra số 0, nếu mẫu số =1 thì chỉ in ra tử số.
 - + Hàm nghịch đảo phân số (làm thay đổi giá trị phân số) `void nghichDao();`
 - + Hàm tính ra phân số nghịch đảo của 1 phân số (phân số sẽ giữ nguyên nhưng hàm trả về giá trị là phân số nghịch đảo của nó). `PhanSo giaTriNghichDao();`
 - + Hàm tính ra giá trị thực của phân số. Chẳng hạn phân số 1/2 có giá trị là 0.5.

- + Hàm so sánh lớn hơn với phân số a . *boolean lonHon(PhanSo a)*;
- + Hàm cộng, trừ, nhân, chia phân số với 1 phân số a . Kết quả của hàm là 1 phân số.
- + Hàm cộng, trừ, nhân, chia phân số với 1 số nguyên. Kết quả của hàm là 1 phân số.
- Viết lớp SDPhanSo có chứa hàm main() sử dụng lớp PhanSo :
 - + Tạo phân số $a = 3/7$, $b = 4/9$. Hiển thị giá trị của chúng ra màn hình.
 - + Tạo 2 phân số x và y . Nhập giá trị cho x và y từ bàn phím.
 - + Hiển thị giá trị nghịch đảo của phân số x ra màn hình (*không làm thay đổi giá trị của x*).
 - + Tính tổng của $x + y$ và in kết quả ra màn hình.
 - + Nhập vào 1 danh sách gồm n phân số (n : nhập từ bàn phím).
 - + Tính tổng n phân số đó.
 - + Tìm phân số lớn nhất trong danh sách phân số trên.
 - + Sắp xếp danh sách phân số theo thứ tự tăng dần.

1.26. Xây dựng lớp TamGiac (*tam giác trong không gian oxy*) có các thuộc tính và phương thức như sau:

- **Thuộc tính:** 3 đỉnh là 3 điểm trong không gian oxy.
- **Phương thức:**
 - + Phương thức khởi tạo mặc định
 - + Phương thức khởi tạo có đầy đủ tham số
 - + Phương thức khởi tạo sao chép
 - + Các phương thức get/set
 - + Phương thức nhập
 - + Phương thức xuất
 - + Phương thức tính diện tích
 - + Phương thức tính chu vi
- Xây dựng lớp Main có phương thức main, suy nghĩ kịch bản sử dụng các phương thức đã cài đặt của lớp TamGiac.

1.27. Xây dựng lớp DaGiac (*đa giác trong không gian oxy*) có các thuộc tính và phương thức sau:

- **Thuộc tính:** danh sách các đỉnh là các điểm trong không gian oxy
- **Phương thức:**
 - + Phương thức khởi tạo mặc định
 - + Phương thức khởi tạo có đầy đủ tham số
 - + Phương thức khởi tạo sao chép
 - + Các phương thức get/set
 - + Phương thức nhập
 - + Phương thức xuất

- + Phương thức tính chu vi
- + Phương thức tìm đỉnh xa gốc tọa độ nhất

Lưu ý: sử dụng 2 cách (*mảng thông thường và ArrayList*).

Xây dựng phương thức main, suy nghĩ kịch bản sử dụng các phương thức đã cài đặt của lớp DaGiac.

1.28. Cài đặt lớp **DiemMau** (Điểm có màu) thừa kế từ lớp Diem bổ sung thêm:

- Thuộc tính: màu (kiểu String)
- Các phương thức:
 - + Hàm xây dựng: DiemMau(int x, int y, String mau)
 - + Hàm gán giá trị màu cho điểm: void GanMau(String mau)
 - + Hàm nhập, hàm hiển thị thông tin
- Viết lớp SDDiemMau có hàm main() thực hiện các công việc sau:
 - + Tạo 1 điểm màu A có tọa độ là (5, 10) và màu là trắng. Hiển thị thông tin ra màn hình.
 - + Tạo 1 điểm màu tổng quát B. Nhập giá trị từ bàn phím cho điểm B. Dời điểm B đi 1 độ dời (10,8). Hiển thị tọa độ điểm B ra màn hình. Gán màu mới cho điểm B là màu “Vàng”.

1.29. Công ty ABC là công ty sản xuất kinh doanh thú nhồi bông. Công ty có nhiều nhân viên làm việc trong 3 bộ phận khác nhau gồm: bộ phận quản lý, bộ phận sản xuất, bộ phận văn phòng. Việc tính lương cho nhân viên dựa vào các thông tin sau:

- Đối với nhân viên văn phòng:

$$\text{Lương} = \text{LCB} + \text{Số ngày LV} * 100.000 + \text{Trợ cấp}$$
- Đối với nhân viên sản xuất:

$$\text{Lương} = \text{LCB} + \text{Số sản phẩm} * 2.000$$
- Đối với nhân viên quản lý:

$$\text{Lương} = \text{LCB} * \text{Hệ số chức vụ} + \text{Thưởng}$$

Hiện tại công ty áp dụng mức LCB là 1,490,000.

Ngoài ra, công ty cần quản lý các thông tin về nhân viên của mình như: họ tên, năm vào làm và các thông số trên để tính lương của nhân viên trong công ty. Hãy thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

1. Nhập thông tin của các nhân viên để phục vụ cho việc tính lương.
2. Xuất thông tin của các nhân viên.
3. Tính tổng lương của công ty và tổng lương của từng loại nhân viên.

1.30. Một trường học cần xây dựng ứng dụng quản lý điểm các môn học. Biết rằng mỗi môn học đều có các thông tin sau: Mã môn học, tên môn học, số tín chỉ. Các môn học được chia làm 3 loại khác nhau:

- **Môn lý thuyết:** có 3 cột điểm là điểm tiểu luận và điểm cuối kỳ. $\text{ĐTB} = \text{TL} * 0.2 + \text{GK} * 0.3 + \text{CK} * 0.5$.
- **Môn thực hành:** có 4 cột điểm kiểm tra. ĐTB sẽ được tính bằng trung bình cộng các bài kiểm tra.

- **Môn đồ án:** có điểm của giáo viên hướng dẫn (GVHD) và giáo viên phản biện đồ án (GVPB). ĐTB sẽ được tính là trung bình cộng điểm của GVHD và GVPB.

Hãy xây dựng ứng dụng quản lý điểm theo mô tả như trên cho phép nhập xuất danh sách môn học và tính điểm trung bình các môn học.

1.31. Công ty ABC cần quản lý 4 loại đối tượng bất động sản là: đất trống, nhà ở, biệt thự và khách sạn. Biết rằng tất cả các đối tượng này đều có các thông tin: mã số, chiều dài, chiều rộng và phương thức tính giá trị. Biết rằng giá trị được tính như sau:

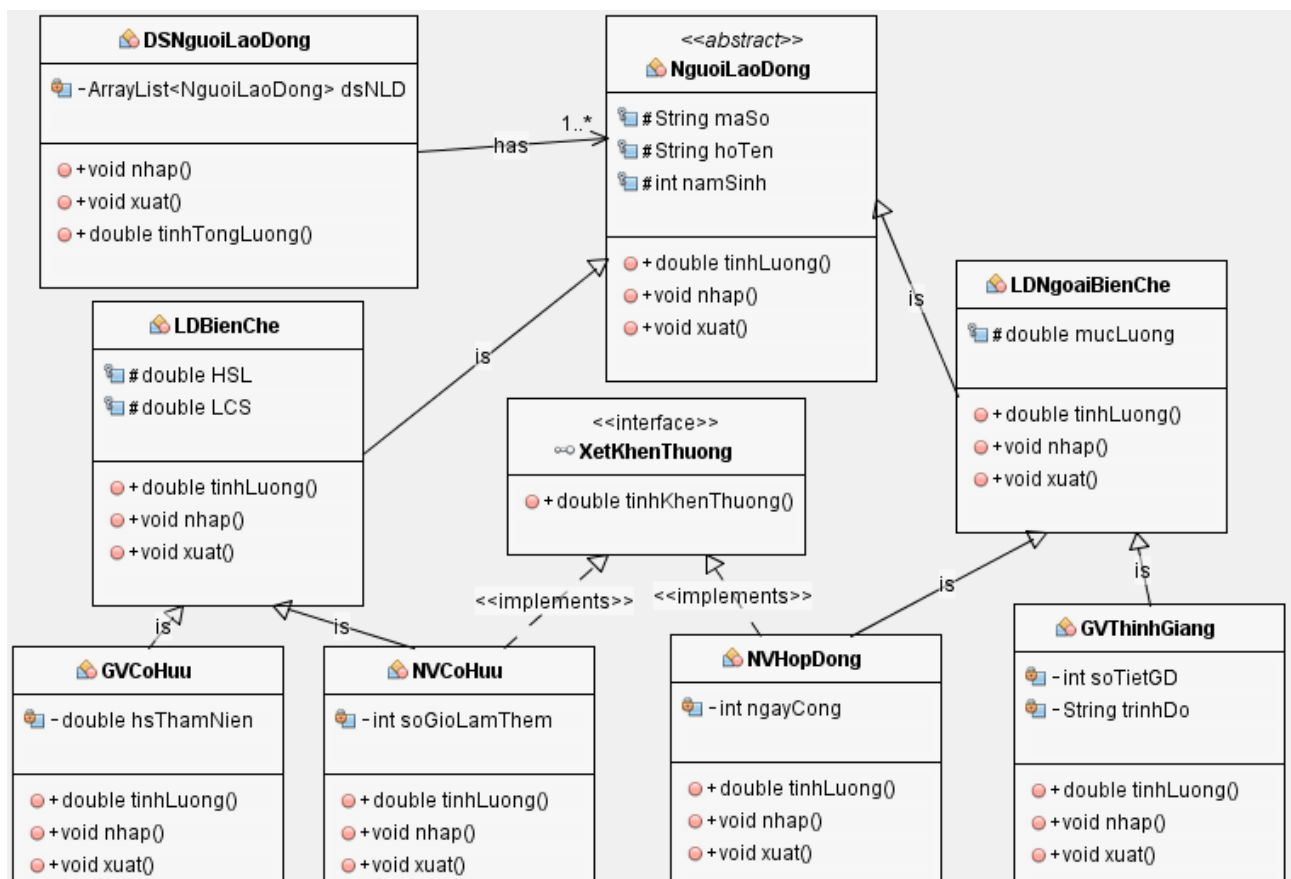
- Đất trống: giá bán = diện tích * 10.000.
- Nhà ở có thêm thông tin số lầu: giá bán = diện tích * 10.000 + số lầu * 100.000.
- Khách sạn có thêm thông tin về số sao: giá bán = 100.000 + số sao * 50.000 .
- Biệt thự: giá bán = diện tích * 400.000.

Tuy nhiên trong 4 loại bất động sản trên thì có 2 loại sản phẩm khi mua khách hàng phải đóng thêm phí kinh doanh là biệt thự và khách sạn. Biết rằng, phí kinh doanh sẽ được tính như sau:

- Biệt thự: Diện tích * 1000
- Khách sạn: Chiều rộng * 5000

Hãy xây dựng ứng dụng quản lý bất động sản theo mô tả như trên cho phép nhập xuất danh sách và tính giá bán của từng loại bất động sản và tổng tất cả bất động sản.

1.32. Xây dựng chương trình theo sơ đồ lớp sau:



Cách tính lương:

- Lương GV cơ hữu = $HSL * LCS * (1 + hsThamNien)$

- Lương NV cơ hữu = $HSL * LCS + soGioLamThem * 50000$
- Lương NV hợp đồng = $mucLuong * ngayCong$
- Lương GV thỉnh giảng = $mucLuong * soTietGD * trinhDo$. Trong đó: ĐH: 1.0; ThS: 1.2; TS: 1.5; PGS: 1.8; GS: 2.0.

Trong đó: LCS giống nhau cho tất cả GV cơ hữu và NV cơ hữu, hiện tại là 1490000.

Xét khen thưởng đối NV cơ hữu và NV hợp đồng:

- NV cơ hữu: nếu $soGioLamThem$ từ 40 trở lên thì thưởng thêm 1000000.
- NV hợp đồng: nếu $ngayCong$ từ 25 trở lên thì thưởng thêm 1500000.

1.33. Xây dựng ứng dụng quản lý khách hàng dùng Text File theo yêu cầu sau:

- Thông tin khách hàng: mã số, họ tên, năm sinh
- Nhập danh sách khách hàng từ bàn phím
- Xuất danh sách khách hàng ra màn hình
- Ghi danh sách khách hàng ra file
- Đọc danh sách khách hàng từ file

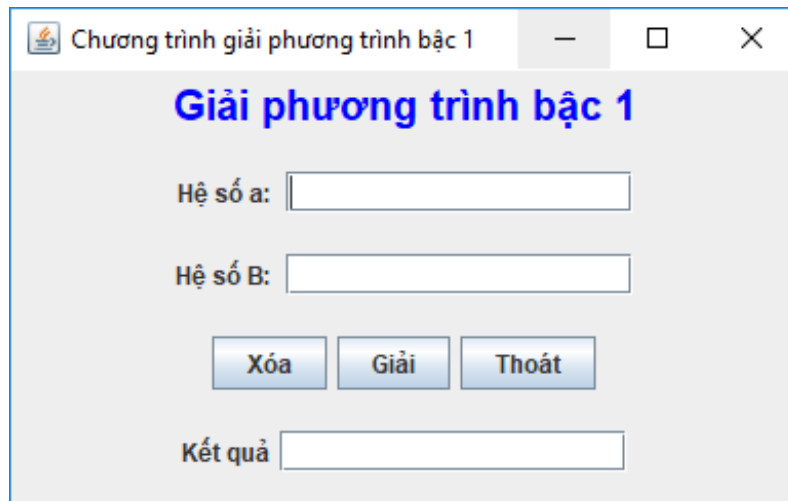
1.34. Xây dựng ứng dụng quản lý danh bạ dùng Serialize File theo yêu cầu sau:

- Thông tin danh bạ: tên, số phone
- Nhập danh sách danh bạ từ bàn phím
- Xuất danh sách danh bạ ra màn hình
- Ghi danh sách danh bạ vào file
- Đọc danh sách danh bạ từ file

1.35. Chỉnh sửa các bài tập từ 1.29 đến 1.32 để thực hiện nhập dữ liệu từ file và xuất kết quả ra file.

Chương 2. Xây dựng giao diện đồ họa với Swing

2.1. Xây dựng ứng dụng giải phương trình bậc 1:



Chương trình giải phương trình bậc 1

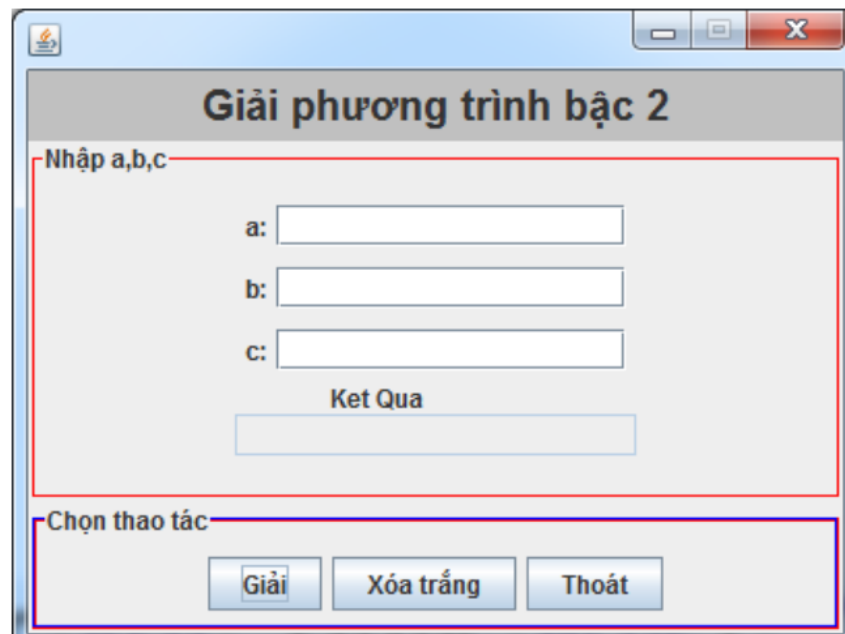
Giải phương trình bậc 1

Hệ số a:

Hệ số B:

Kết quả

2.2. Xây dựng ứng dụng giải phương trình bậc 2



Giải phương trình bậc 2

Nhập a,b,c

a:

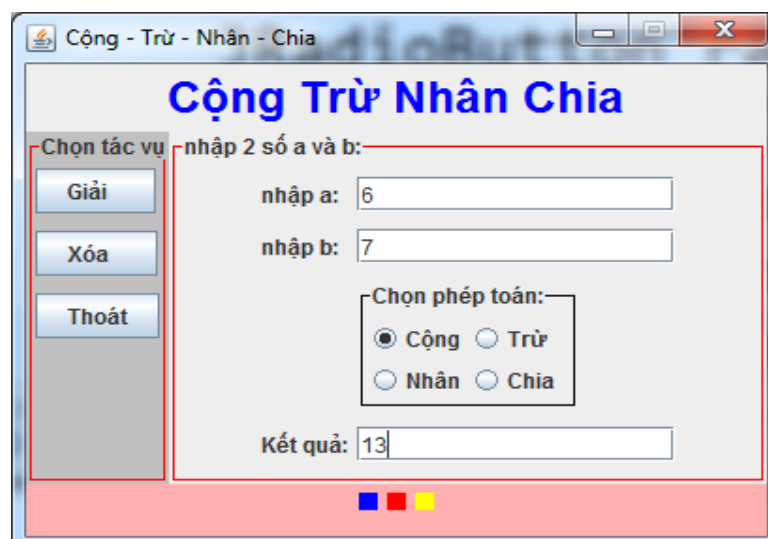
b:

c:

Ket Qua

Chọn thao tác

2.3. Xây dựng ứng dụng tính toán cộng, trừ, nhân, chia như sau:



Cộng - Trừ - Nhân - Chia

Cộng Trừ Nhân Chia

Chọn tác vụ

nhập 2 số a và b:

nhập a:

nhập b:

Chọn phép toán:

☒ Cộng ☐ Trừ

☐ Nhân ☐ Chia

Kết quả:

2.4. Xây dựng máy tính đơn giản:

0				
1	2	3	/	sqrt
4	5	6	*	%
7	8	9	-	1/x
0	+/-	C	+	=

2.5. Thiết kế Frame theo mẫu sau:

Register information

Name:

Date of Birth:

Email:

Sex: ☒ Male ☐ Female

Province:

Operations

Viết code xử lý theo yêu cầu:

- Click nút **Register** thì các thông tin người dùng nhập trên Frame sẽ được ghi vào file “dulieu.txt” nếu thông tin đầy đủ và hợp lệ.
- Click nút **Reset** thì các thông tin người dùng nhập sẽ bị xóa bỏ.

2.6. Xây dựng ứng dụng quản lý nhân viên

Chương trình quản lý nhân viên

Chọn phòng ban:

Danh sách nhân viên

- Nguyễn Văn An
- Lê Thanh Bình

Thông tin chi tiết

Mã số:

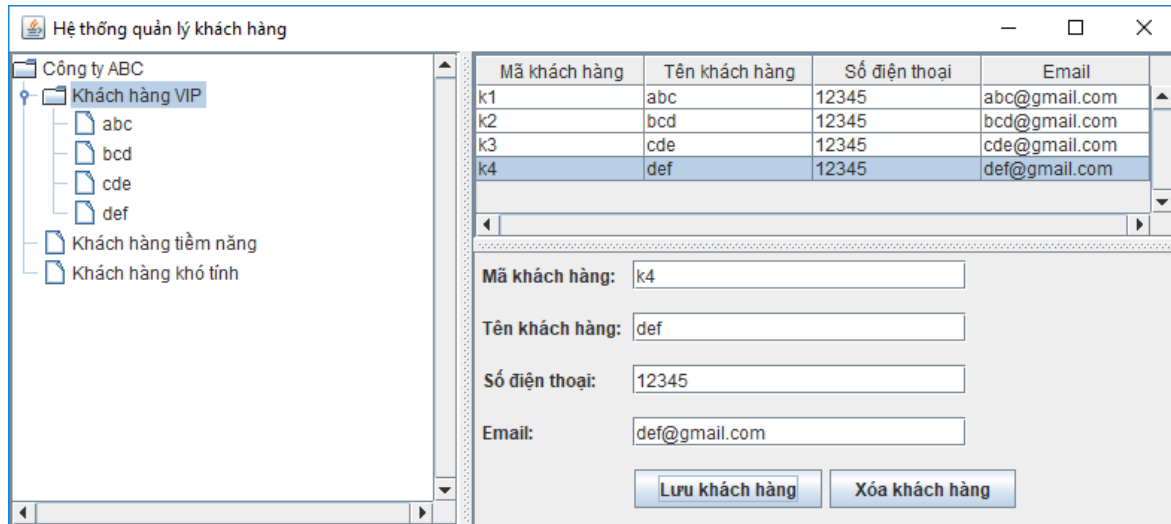
Họ tên:

Ngày VL:

Năm sinh:

Lưu ý: Thông tin phòng ban gồm Mã phòng, tên phòng; nhân viên gồm mã số, họ tên, ngày vào làm, ngày sinh.

2.7. Xây dựng ứng dụng quản lý khách hàng

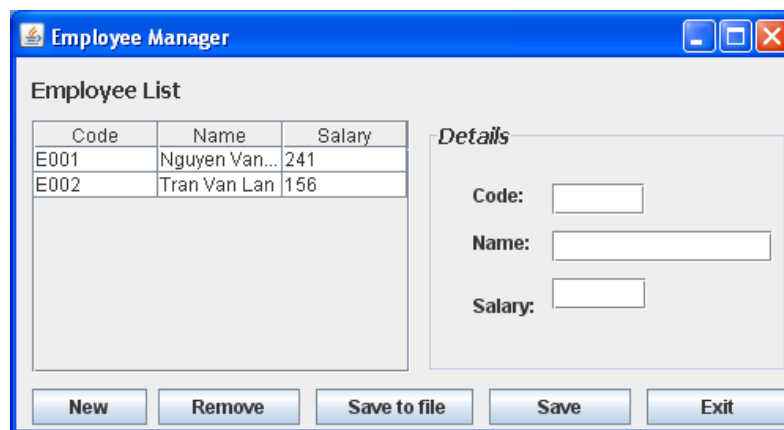


Mã khách hàng	Tên khách hàng	Số điện thoại	Email
k1	abc	12345	abc@gmail.com
k2	bcd	12345	bcd@gmail.com
k3	cde	12345	cde@gmail.com
k4	def	12345	def@gmail.com

Mã khách hàng: k4
Tên khách hàng: def
Số điện thoại: 12345
Email: def@gmail.com

Lưu khách hàng Xóa khách hàng

2.8. Xây dựng ứng dụng quản lý nhân viên như sau:



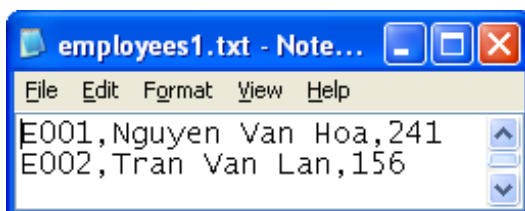
Code	Name	Salary
E001	Nguyen Van...	241
E002	Tran Van Lan	156

Details

Code:
Name:
Salary:

New Remove Save to file Save Exit

Cấu trúc file text lưu thông tin Employee:

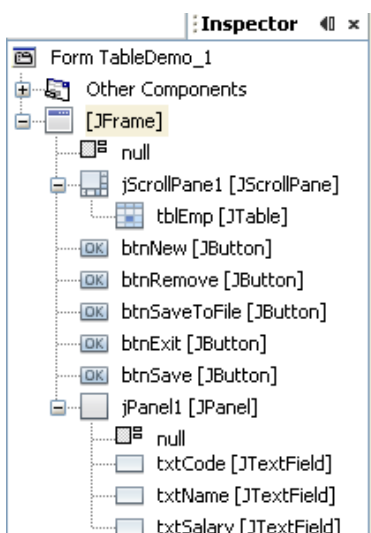


employees1.txt - Note...

File Edit Format View Help

E001,Nguyen Van Hoa,241
E002,Tran Van Lan,156

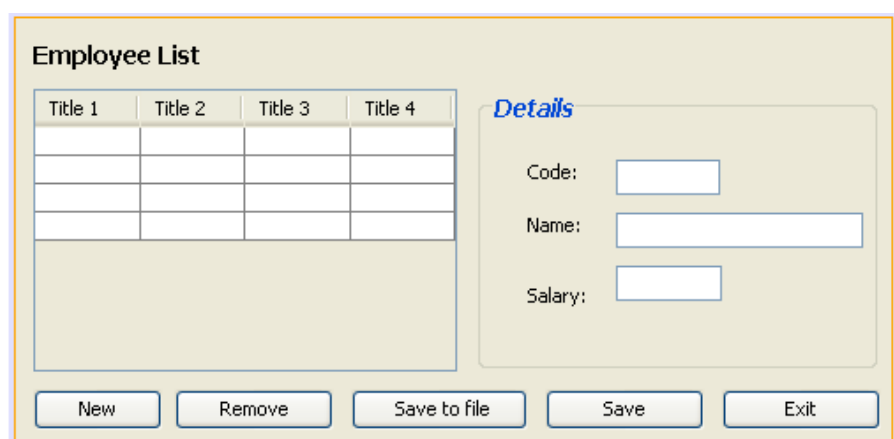
Hướng dẫn:



Inspector

Form TableDemo_1

- Other Components
- [JFrame]
- null
- JScrollPane [JScrollPane]
- tblEmp [JTable]
- btnNew [JButton]
- btnRemove [JButton]
- btnSaveToFile [JButton]
- btnExit [JButton]
- btnSave [JButton]
- JPanel1 [JPanel]
- null
- txtCode [JTextField]
- txtName [JTextField]
- txtSalary [JTextField]



Employee List

Title 1	Title 2	Title 3	Title 4

Details

Code:
Name:
Salary:

New Remove Save to file Save Exit

```

import java.util.Vector; // header and data of the table
import java.util.StringTokenizer; // for splitting string
import java.io.File;
import java.io.FileReader; // reading data from text file
import java.io.BufferedReader;
import java.io.PrintWriter; // write data to text file
import javax.swing.table.DefaultTableModel; // control table
import javax.swing.JOptionPane; // for build-in dialog
public class TableDemo_1 extends javax.swing.JFrame {
    String filename="employees1.txt";
    Vector<String> header= new Vector<String>();
    Vector data= new Vector();
    boolean addNew=false;
    boolean changed=false; // Are data changed?
    /** Creates new form TableDemo_1 */

private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (changed==true)
    { if (JOptionPane.showConfirmDialog(this,"Data changed. Save Y/N?")==
        JOptionPane.OK_OPTION)
        btnSaveToFileActionPerformed(null);
    }
    System.exit(0);
}

```

```

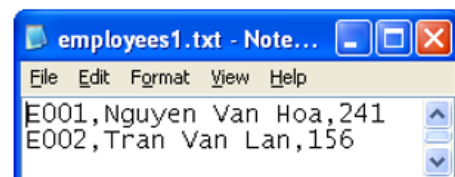
/** Creates new form TableDemo_1 */
public TableDemo_1() {
    initComponents();
    this.setSize(500,270);
    // setup header of the table
    header.add("Code"); header.add("Name"); header.add("Salary");
    loadData();
    DefaultTableModel tblModel;
    tblModel= (DefaultTableModel) this.tblEmp.getModel();
    tblModel.setDataVector(data, header);
}

private void loadData(){
    try
    { File f= new File(filename);
      FileReader fr= new FileReader(f);
      BufferedReader bf = new BufferedReader(fr);
      String aDetails;
      while ((aDetails = bf.readLine()) !=null) {
          StringTokenizer stk= new StringTokenizer(aDetails, ",");
          String code=stk.nextToken();
          String name= stk.nextToken();
          String salaryStr= stk.nextToken();
          Vector<String> v= new Vector<String>();
          v.add(code); v.add(name); v.add(salaryStr);
          data.add(v);
      }
      bf.close(); fr.close();
    }
    catch (Exception e)
    { JOptionPane.showMessageDialog(this, "The file " +
        filename + " not found." );
    }
}

```

Employee List

Code	Name	Salary
E001	Nguyen Van...	241
E002	Tran Van Lan	156



```
private void btnNewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.addNew = true;    // entering the add-new mode
    this.txtCode.setText(""); // preparing input data
    this.txtCode.setEditable(true);
    this.txtName.setText("");
    this.txtSalary.setText("");
    this.txtCode.requestFocus();
}
}
```

```
private void btnSaveToFileActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        File f= new File(filename);
        PrintWriter pf= new PrintWriter(f);
        int n= this.tblEmp.getRowCount();
        for (int i=0;i<n; i++){
            Vector<String> v = (Vector<String>) (data.get(i));
            String S= v.get(0) + "," + v.get(1) + "," + v.get(2);
            pf.println(S);
        }
        pf.close();
        changed=false;
    }
    catch (Exception e){
        JOptionPane.showMessageDialog(this, e);
    }
}
}
```

```
private void btnRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    // Get position of selected employee
    int pos = this.tblEmp.getSelectedRow();
    if (pos>=0)
    {
        String code= (String) (tblEmp.getValueAt(pos,0));
        if (JOptionPane.showConfirmDialog(this,"Delete the " +
            code + " employee ?")== JOptionPane.OK_OPTION)
        {
            data.remove(pos);
            tblEmp.updateUI();
            changed=true;
        }
    }
}
}
```

```
private void tblEmpMouseReleased(java.awt.event.MouseEvent evt) {
    // Turn off the on-table editing mode when user double-clicks on a cell
    if (this.tblEmp.isEditing())
    {
        int row= tblEmp.getSelectedRow();
        int column= tblEmp.getSelectedColumn();
        tblEmp.getCellEditor(row,column).cancelCellEditing();
    }
}
}
```

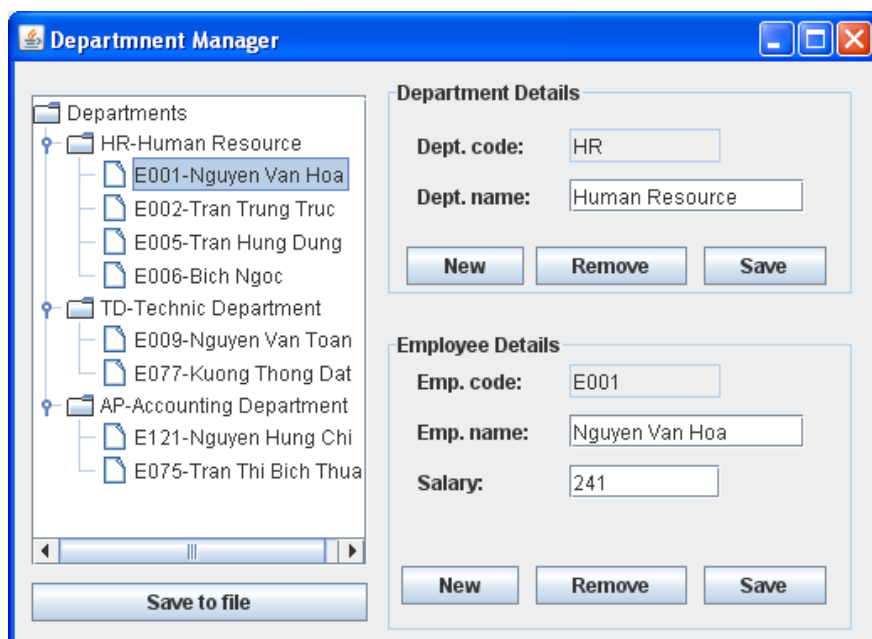
```

private void tblEmpMouseClicked(java.awt.event.MouseEvent evt) {
    // View current employee
    int row= this.tblEmp.getSelectedRow();
    this.txtCode.setText((String) (tblEmp.getValueAt(row,0)));
    this.txtCode.setEditable(false); // user can not update the code
    this.txtName.setText((String) (tblEmp.getValueAt(row,1)));
    this.txtSalary.setText((String) (tblEmp.getValueAt(row,2)));
    addNew = false; // view and update mode
}

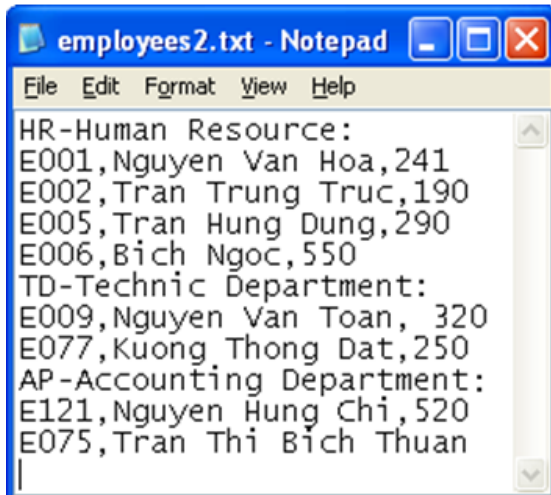
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (!valid()) return;
    String code = txtCode.getText();
    String name = txtName.getText();
    String salaryStr = txtSalary.getText();
    if (addNew)
    { Vector v= new Vector();
      v.add(code); v.add(name); v.add(salaryStr);
      data.add(v);
      addNew=false;
    }
    else
    { int pos= tblEmp.getSelectedRow();
      Vector v= (Vector) data.get(pos);
      v.set(1, name);
      v.set(2, salaryStr);
    }
    tblEmp.updateUI();
    changed=true;
}

```

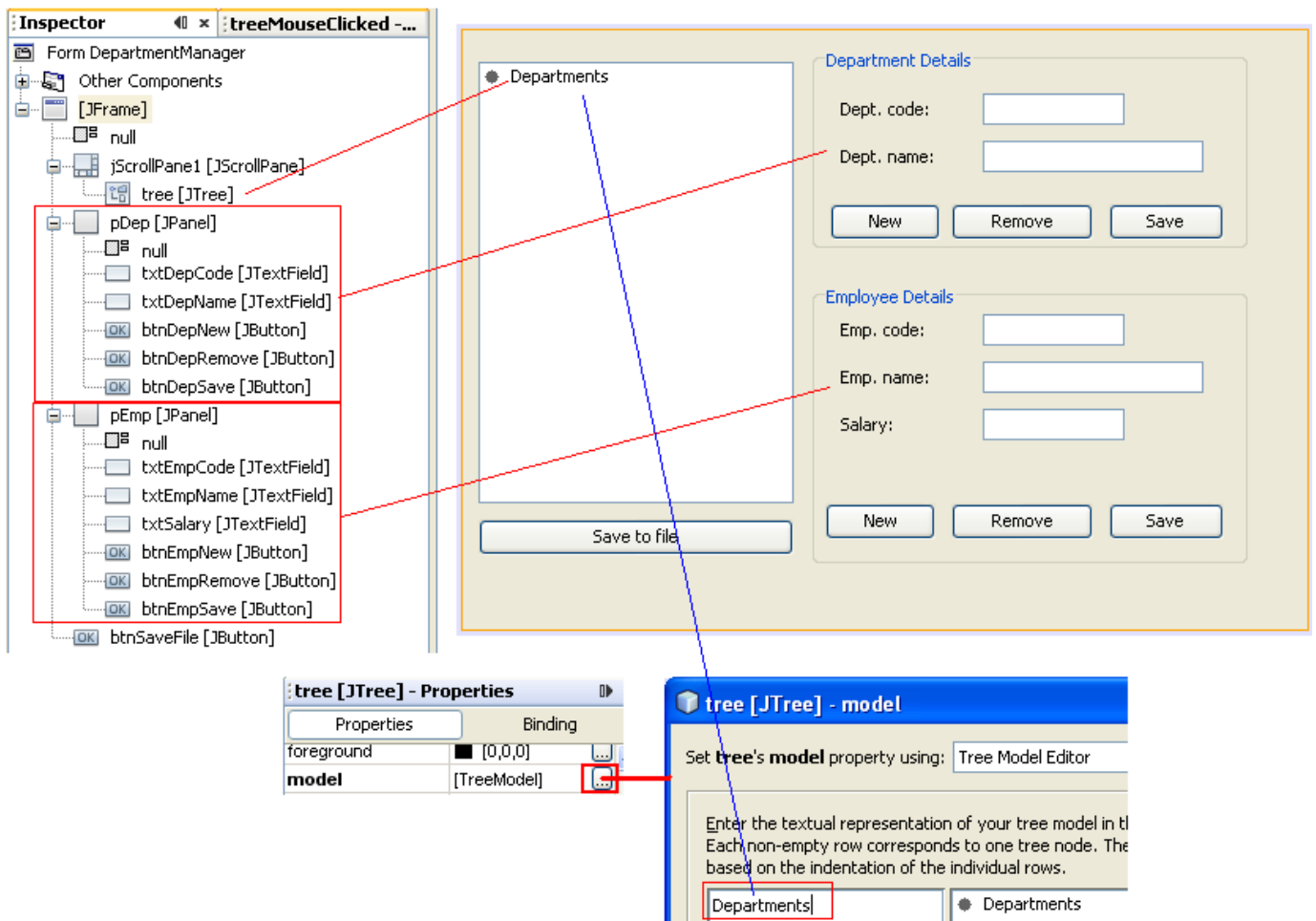
2.9. Xây dựng ứng dụng quản lý nhân viên như sau:



Cấu trúc file text lưu thông tin Employee:



Hướng dẫn:



```
/* Class for a department */
```

```
package treedemo;
```

```
public class Department {
```

```
    String depCode, depName;
```

```
    public Department(String depCode, String depName) {...}
```

```
    public String getDepCode() {...}
```

```
    public void setDepCode(String depCode) {...}
```

```
    public String getDepName() {...}
```

```
    public void setDepName(String depName) {...}
```

```
    public String toString() {
        return depCode + "-" + depName;
    }
}
```

```
/* Class for an employee */
```

```
package treedemo;
```

```
public class Employee {
```

```
    String empCode, empName; int salary;
```

```
    public Employee(String empCode, String empName, int salary){
```

```
        this.empCode = empCode;
```

```
        this.empName = empName;
```

```
        this.salary = salary;
```

```
    }
```

```
    public String getEmpCode() {...}
```

```
    public void setEmpCode(String empCode) {...}
```

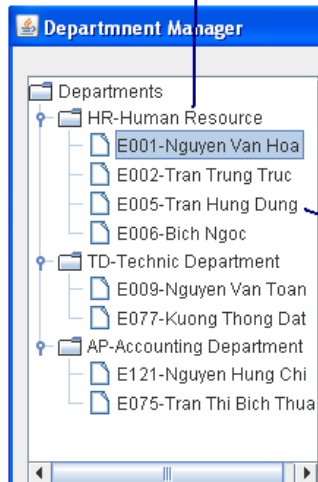
```
    public String getEmpName() {...}
```

```
    public void setEmpName(String empName) {...}
```

```
    public int getSalary() {...}
```

```
    public void setSalary(int salary) {...}
```

```
    public String toString() {
        return empCode + "-" + empName;
    }
}
```



```
DepartmentManager.java *
Source Design
13 import javax.swing.tree.DefaultMutableTreeNode; // a node on a tree
14 import javax.swing.tree.TreePath; // get a path from the root of the tree
15 import javax.swing.JOptionPane; // common dialog
16 import java.io.*; // File processing
17 import java.util.StringTokenizer; // for string splitting
18 import java.util.Enumeration; // interface for enumeration
19 import java.util.Iterator; // for traversing an enumeration
20
21 public class DepartmentManager extends javax.swing.JFrame {
22     String filename= "employees2.txt";
23     DefaultMutableTreeNode root= null; // root of the tree
24     DefaultMutableTreeNode curDepNode= null; // current department
25     DefaultMutableTreeNode curEmpNode= null; // current employee
26     boolean addNewDep= false;
27     boolean addNewEmp= false;
```

employees2.txt - Notepad

```
File Edit Format View Help
HR-Human Resource:
E001,Nguyen Van Hoa,241
E002,Tran Trung Truc,190
E005,Tran Hung Dung,290
E006,Bich Ngoc,550
TD-Technic Department:
E009,Nguyen Van Toan, 320
E077,Kuong Thong Dat,250
AP-Accounting Department:
E121,Nguyen Hung Chi,520
E075,Tran Thi Bich Thuan
```

```
/** Creates new form DepartmentManager */
```

```
public DepartmentManager() {
```

```
    initComponents();
```

```
    this.setSize(520,380);
```

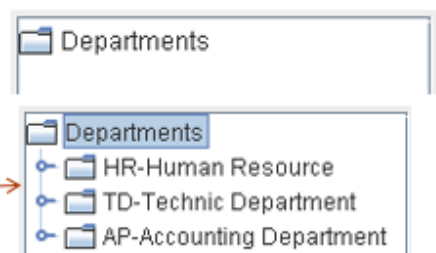
```
    root= (DefaultMutableTreeNode) (this.tree.getModel().getRoot());
```

```
    loadData(); // Loading initial data from file
```

```
    TreePath path=new TreePath(root); // expanding the tree
```

```
    tree.expandPath(path);
```

```
}
```



```
// Load initial data from the employee2.txt file
```

```
private void loadData() {
    String S=""; StringTokenizer stk;
    try {
        FileReader f= new FileReader(filename);
        BufferedReader bf= new BufferedReader (f);
        while ((S=bf.readLine()) !=null) {
            S= S.trim();
            boolean isDept = (S.charAt(S.length()-1)==' ');
            stk= new StringTokenizer(S, "-:");
            String code= stk.nextToken().trim();
            String name= stk.nextToken().trim();
            if (isDept) { // department details
                curDepNode=new DefaultMutableTreeNode(new Department(code, name));
                root.add(curDepNode); // add the node root
            }
            else { // employee details
                int salary= Integer.parseInt(stk.nextToken().trim());
                curEmpNode=new DefaultMutableTreeNode(new Employee(code, name, salary));
                curDepNode.add(curEmpNode); // add to the department
            }
        }
        bf.close(); f.close();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
```

employees2.txt - Notepad

File Edit Format View Help

HR-Human Resource:
E001,Nguyen Van Hoa,241
E002,Tran Trung Truc,190
E005,Tran Hung Dung,290
E006,Bich Ngoc,550
TD-Technic Department:
E009,Nguyen Van Toan, 320
E077,Kuong Thong Dat,250
AP-Accounting Department:
E121,Nguyen Hung Chi,520
E075,Tran Thi Bich Thuan

Departments

- HR-Human Resource
 - E001-Nguyen Van Hoa
 - E002-Tran Trung Truc
 - E005-Tran Hung Dung
 - E006-Bich Ngoc
- TD-Technic Department
 - E009-Nguyen Van Toan
 - E077-Kuong Thong Dat
- AP-Accounting Department
 - E121-Nguyen Hung Chi
 - E075-Tran Thi Bich Thua

```
// Show details of current department and employee
```

```
private void viewDeptAndEmp() {
    Department curDep = null;
    Employee curEmp = null;
    if (curDepNode != null) {
        curDep = (Department) (curDepNode.getUserObject());
    }
    if (curEmpNode != null) {
        curEmp = (Employee) (curEmpNode.getUserObject());
    }
    this.txtDepCode.setText(curDep != null ? curDep.getDepCode() : "");
    this.txtDepName.setText(curDep != null ? curDep.getDepName() : "");
    this.txtEmpCode.setText(curEmp != null ? curEmp.getEmpCode() : "");
    this.txtEmpName.setText(curEmp != null ? curEmp.getEmpName() : "");
    this.txtSalary.setText("" + (curEmp != null ? curEmp.getSalary() : ""));
    this.txtDepCode.setEditable(false);
    this.txtEmpCode.setEditable(false);
}
```



```

private void treeMouseClicked(java.awt.event.MouseEvent evt) {
    // turn of the on-tree editing mode
    tree.cancelEditing();
    // Get the clicked node at the last component of the path
    TreePath path= tree.getSelectionPath();
    if (path==null) return;
    DefaultMutableTreeNode selectedNode = null;
    selectedNode= (DefaultMutableTreeNode) (path.getLastPathComponent());
    // Get the selected object in the model
    Object selectedObj =selectedNode.getUserObject();
    // Checking what is the selected object
    if (selectedNode==root)
        this.curDepNode= this.curEmpNode=null;
    else {
        if (selectedObj instanceof Department) {
            this.curDepNode= selectedNode;
            this.curEmpNode=null;
        }
        else if (selectedObj instanceof Employee){
            curEmpNode= selectedNode;
            curDepNode= (DefaultMutableTreeNode) (selectedNode.getParent());
        }
    }
    viewDeptAndEmp();
}

```

```

private void btnDepNewActionPerformed(java.awt.event.ActionEvent evt) {
    // Make the GUI ready for a new department details entered
    this.addNewDep=true;
    this.txtDepCode.setText("");
    this.txtDepName.setText("");
    this.txtEmpCode.setText("");
    this.txtEmpName.setText("");
    this.txtSalary.setText("");
    this.txtDepCode.setEditable(true);
    this.txtDepCode.requestFocus();
}

```

```

private void btnDepRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    // Removing a department
    if (this.curDepNode.getChildCount(>0) {
        String msg = "Remove all employees before deleting a department.";
        JOptionPane.showMessageDialog(this,msg);
    }
    else {
        int response= JOptionPane.showConfirmDialog(this,"Delete this department- OK?");
        if (response==JOptionPane.OK_OPTION) {
            root.remove(this.curDepNode);
            tree.updateUI();
        }
    }
}

```

curDepNode=curEmpNode=null;
(the department was removed)

```
// checking details of the department is valid or not
private boolean validDepDetails() {
    // your code here
    return true;
}
```

```
private void btnDepSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // Save department details
    String code= this.txtDepCode.getText().trim().toUpperCase();
    txtDepCode.setText(code);
    String name= this.txtDepName.getText().trim();
    txtDepName.setText(name);
    if (! validDepDetails()) return;
    if (addNewDep==true) {
        Department newDep= new Department(code, name);
        root.add(new DefaultMutableTreeNode (newDep));
    }
    else {
        ((Department) curDepNode.getUserObject()).setDepName(name);
    }
    this.tree.updateUI();
    this.addNewDep=false;
    curDepNode=curEmpNode=null;
}
```

```
private void btnEmpNewActionPerformed(java.awt.event.ActionEvent evt) {
    // Make the GUI ready for a new employee details entered
    this.addNewEmp=true;
    this.txtEmpCode.setText("");
    this.txtEmpName.setText("");
    this.txtSalary.setText("");
    this.txtEmpCode.setEditable(true);
    this.txtEmpCode.requestFocus();
}
```

```
private void btnEmpRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    // Removing an employee
    if (this.curEmpNode !=null) {
        int response= JOptionPane.showConfirmDialog(this,"Delete this employee- OK?");
        if (response==JOptionPane.OK_OPTION) {
            curDepNode.remove(this.curEmpNode);
            tree.updateUI();
        }
    }
}
```

```

private void btnEmpSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // Save/ update new employee/ employee details
    String code= this.txtEmpCode.getText().trim().toUpperCase();
    txtEmpCode.setText(code);
    String name= this.txtEmpName.getText().trim();
    txtEmpName.setText(name);
    String salaryStr= this.txtSalary.getText().trim();
    txtSalary.setText(salaryStr);
    int sal= Integer.parseInt(salaryStr);
    if (! validEmpDetails()) return;
    if (addNewEmp==true) {
        Employee newEmp= new Employee(code, name, sal);
        curDepNode.add(new DefaultMutableTreeNode (newEmp));
    }
    else {
        Employee emp= (Employee) (curEmpNode.getUserObject());
        emp.setEmpName (name);
        emp.setSalary(sal);
    }
    this.tree.updateUI();
    this.addNewEmp=false;
}

```

```

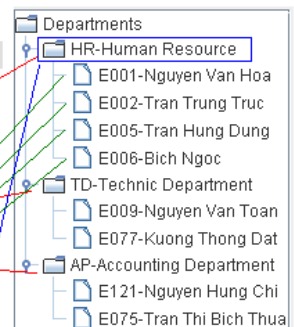
// checking details of the employee is valid or not
private boolean validEmpDetails() {
    // your code here
    return true;
}

```

```

private void btnSaveFileActionPerformed(java.awt.event.ActionEvent evt) {
    // Saving details to the file
    if (root.getChildCount()==0) return;
    String S;
    try {
        FileWriter f= new FileWriter (filename);
        PrintWriter pf= new PrintWriter(f);
        Enumeration depts= root.children();// get departments
        while (depts.hasMoreElements()) {
            DefaultMutableTreeNode depNode= (DefaultMutableTreeNode) depts.nextElement();
            Department dept= (Department) (depNode.getUserObject());
            S = dept.getDepCode() + "-" + dept.getDepName() + ":" ;
            pf.println(S);
            Enumeration emps= depNode.children(); // get employees
            while (emps.hasMoreElements()) {
                DefaultMutableTreeNode empNode= (DefaultMutableTreeNode) emps.nextElement();
                Employee emp= (Employee) (empNode.getUserObject());
                S = emp.getEmpCode() + "," + emp.getEmpName() + "," + emp.getSalary();
                pf.println(S);
            }
        }
        pf.close();f.close();
        JOptionPane.showMessageDialog(this, "Data are saved to the file " + filename);
    }
    catch (Exception e){
        JOptionPane.showMessageDialog(this, e);
    }
}

```



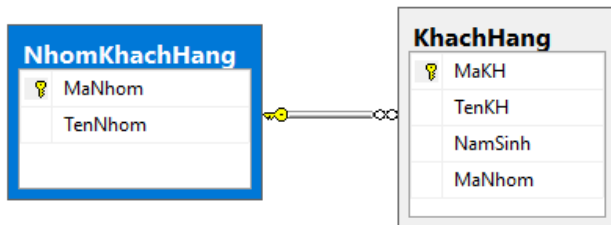
Chương 3. Lập trình ứng dụng Cơ sở dữ liệu

3.1. Xây dựng ứng dụng quản lý khách hàng

Mã khách hàng	Tên khách hàng	Năm sinh
KH1	Nguyễn Văn A	1990
KH2	Lê Thị B	1992
KH3	Trần Thị C	1995
KH4	Lê Văn D	1995
KH5	Nguyễn Thị E	1980
KH6	Lê Thị F	1972

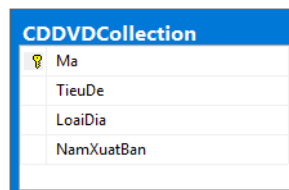
Mã:
Tên:
Năm sinh:

Lưu ý: Tạo cơ sở dữ liệu



3.2. Xây dựng ứng dụng quản lý đĩa CD/DVD

– Cơ sở dữ liệu:



– Màn hình chính

Mã:
Tiêu đề:
Loại:
Năm xuất bản:

Mã	Tiêu đề	Loại	Năm XB
sp1	CD nhạc sến	CD	2015
sp2	DVD học tiếng Anh	DVD	2016
sp3	DVD học kế toán	DVD	2018
sp4	Đĩa phim kiếm hiệp	DVD	2000
sp5	Đĩa nhạc xuân	DVD	2018
sp6	Đĩa học lập trình J...	DVD	2015

- Khi click vào button Tìm kiếm → hiển thị màn hình tìm kiếm

Mã	Tiêu đề	Loại	Năm XB
sp2	DVD học tiếng Anh	DVD	2016
sp3	DVD học kế toán	DVD	2018
sp4	Đĩa phim kiếm hiệp	DVD	2000
sp5	Đĩa nhạc xuân	DVD	2018
sp6	Đĩa học lập trình J...	DVD	2015

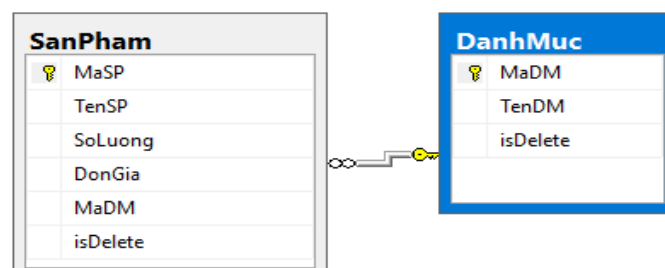
3.3. Viết các Stored Procedures:

- Liệt kê toàn bộ CDDVD;
- Liệt kê danh sách CDDVD theo năm xuất bản;
- Liệt kê danh sách CDDVD theo năm xuất bản và loại đĩa.

Áp dụng các Stored Procedure cho các chức năng trong bài tập 3.2.

3.4. Xây dựng dụng quản lý sản phẩm

- Cơ sở dữ liệu:



- Giao diện ứng dụng:

Mã sản phẩm	Tên sản phẩm	Số lượng	Đơn giá
sp3	Chảo không dính	500	100
sp4	Nồi nấu lẩu	200	14
sp5	Bếp điện từ	200	15

3.5. Xây dựng ứng dụng quản lý nhà sách

1. Cơ sở dữ liệu

Tạo CSDL gồm các table: NgươuDung, LoaiSach, Sach, HoaDon, ChiTietHoaDon, PhieuNhapSach, ChiTietPhieuNhapSach.

1.1. Bảng người dùng (NguoiDung)

	Column Name	Data Type
🔑	ND_ID	int
	TenDangNhap	nvarchar(20)
	MatKhau	nvarchar(45)
	HoTen	nvarchar(255)
	Email	nvarchar(45)
	DiaChi	nvarchar(255)
	NgaySinh	date
	GioiTinh	nvarchar(10)

1.2. Bảng loại sách (LoaiSach)

	Column Name	Data Type
🔑	LS_ID	int
	Ten	nvarchar(100)
	MoTa	nvarchar(255)

1.3. Bảng sách (Sach)

	Column Name	Data Type
🔑	S_ID	int
	TieuDe	nvarchar(255)
	TacGia	nvarchar(255)
	NamXuatBan	int
	TheLoai	int
	MoTa	nvarchar(255)

1.4. Bảng hóa đơn (HoaDon)

	Column Name	Data Type
🔑	HD_ID	int
	Ngay	date
	NhanVien	nvarchar(255)
	TenKhachHang	nvarchar(255)
	GhiChu	nvarchar(255)

1.5. Bảng chi tiết hóa đơn (ChiTietHoaDon)

	Column Name	Data Type
🔑	CTHD_ID	int
	IDHoaDon	int
	IDSach	int
	SoLuong	int
	GiaBan	float

1.6. Bảng phiếu nhập sách

	Column Name	Data Type
🔑	PNS_ID	int
	GhiChu	nvarchar(255)
	Ngay	date
	NhanVien	nvarchar(255)

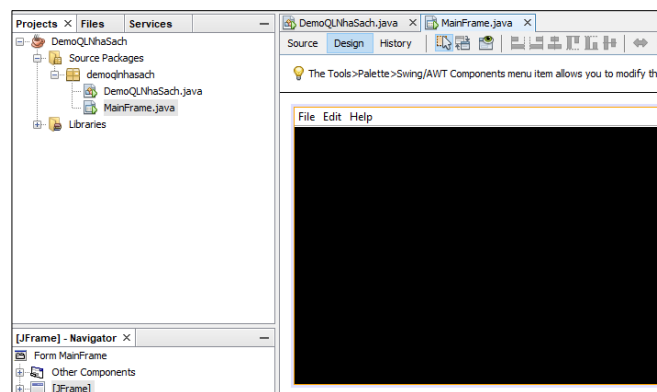
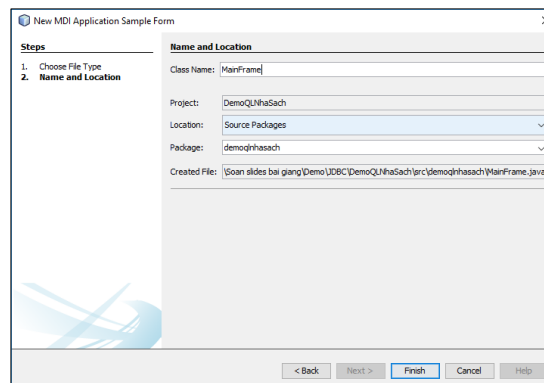
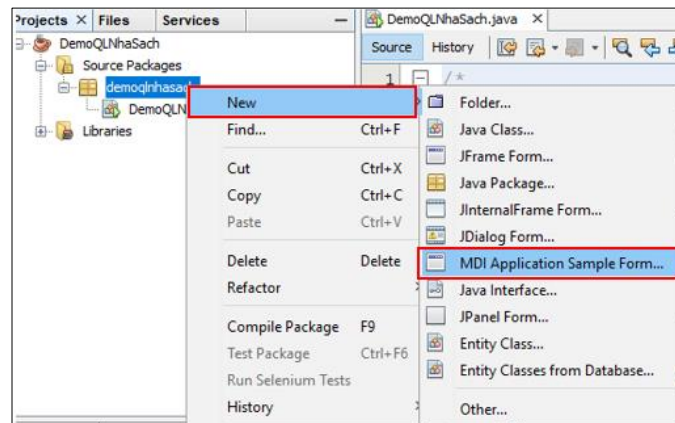
1.7. Bảng chi tiết phiếu nhập sách

	Column Name	Data Type
🔑	CTPNS_ID	int
	IDPhieuNhapSach	int
	IDSach	int
	SoLuong	int
	DonGia	float

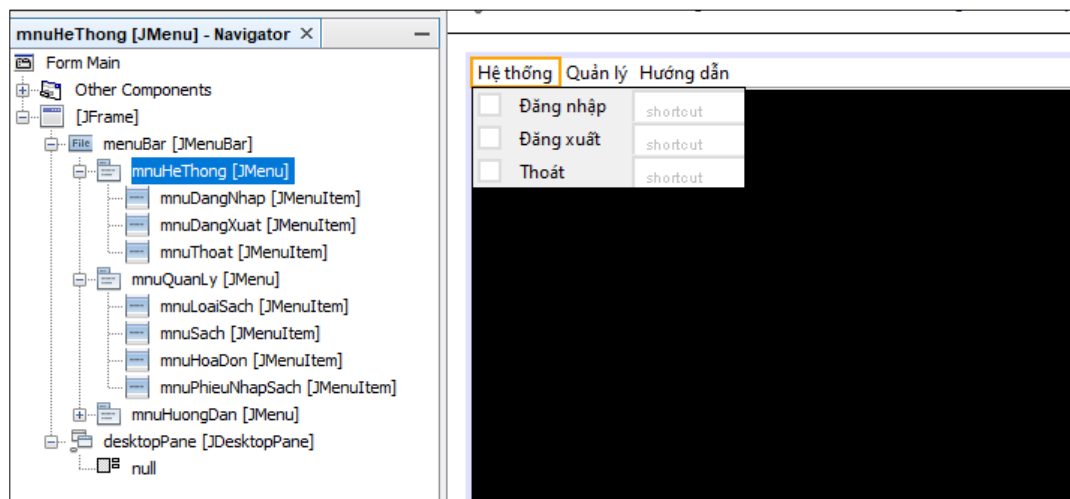
1.8. Sinh viên tự tạo quan hệ giữa các bảng và nhập dữ liệu vào các bảng.

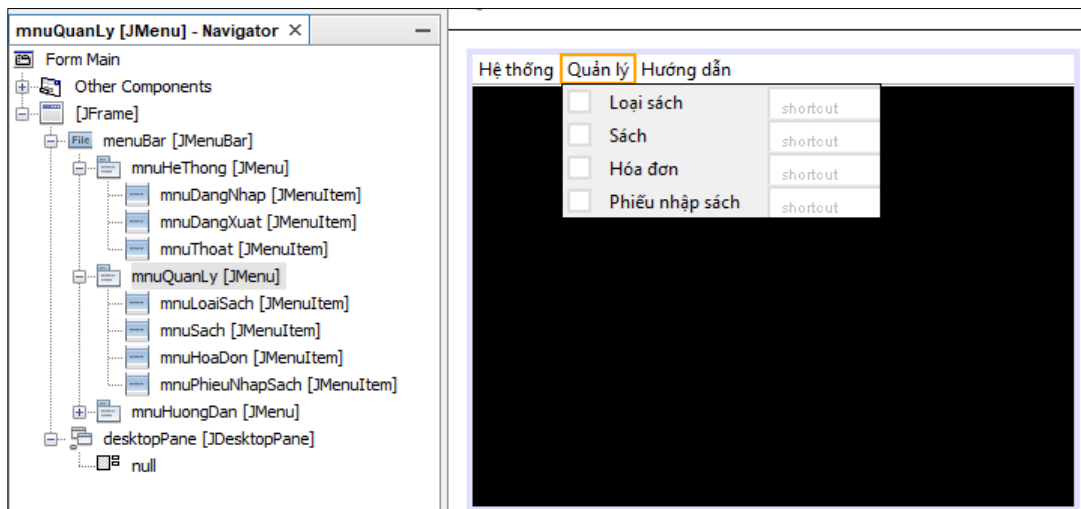
2. Xây dựng ứng dụng

2.1. Tạo project



2.2. Tạo menu

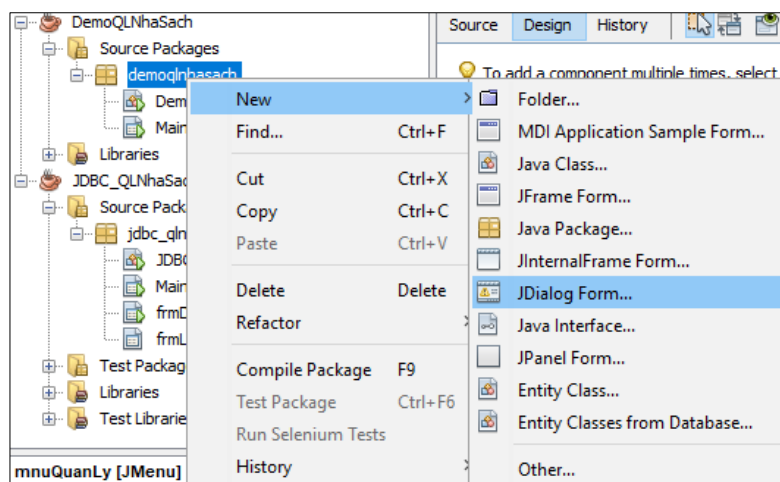




2.3. Xử lý chức năng đăng nhập

2.3.1. Thiết kế màn hình đăng nhập

Chọn *New > JDialog Form*



New JDialog Form

Steps

- Choose File Type
- Name and Location**

Name and Location

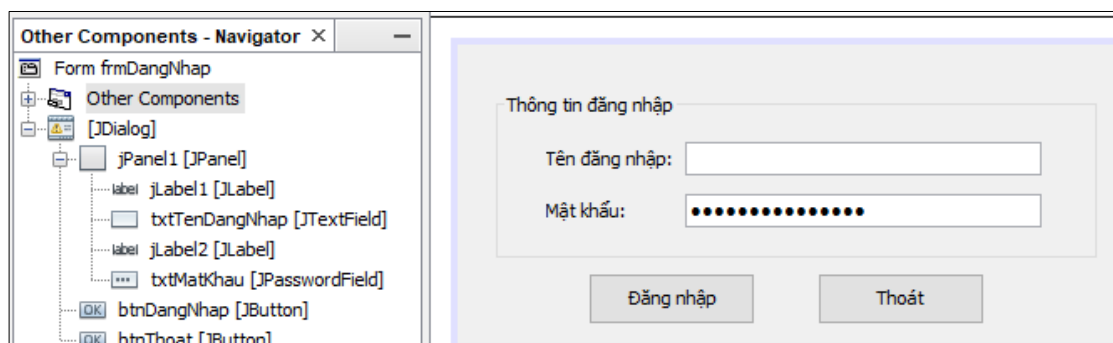
Class Name: frmDangNhap

Project: DemoQLNhaSach

Location: Source Packages

Package: demoqlnhasach

Created File: an slides bai giang\Demo\JDBC\DemoQLNhaSach\src\demoqlnhasach\frmDangNhap.java



2.3.2. Thêm một số thuộc tính và phương thức cho frmDangNhap

```
public class frmDangNhap extends javax.swing.JDialog {

    private String strTenDangNhap;
    private String strMatKhau;
    private String strTenNguoiDung;
    private boolean bKetQuaDangNhap;

    public String getStrMatKhau() {
        return strMatKhau;
    }

    public String getStrTenNguoiDung() {
        return strTenNguoiDung;
    }

    public boolean getKetQuaDangNhap() {
        return bKetQuaDangNhap;
    }

    public frmDangNhap(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }
}
```

2.3.3. Xử lý sự kiện click cho nút đăng nhập

- Trong màn hình Design của frmDangNhap > click phải chuột > *Events* > *Action* > *actionPerformed*.
- Source code xử lý:

```
private void btnDangNhapActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String strServer = "ADMIN";
    String strDatabase = "QL_NhaSach";
    Connection connection = null;
    Statement statement = null;
    ResultSet rs = null;
    try{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        String connectionURL = "jdbc:sqlserver://" + strServer
            + ":1433;databaseName=" + strDatabase
            + ";integratedSecurity = true";
        connection = DriverManager.getConnection(connectionURL);
        statement = connection.createStatement();
        String sql = "SELECT * FROM NguoiDung WHERE "
            + "TenDangNhap = '" + txtTenDangNhap.getText() + "' "
            + "AND MatKhau = '" + txtMatKhau.getText() + "'";
        rs = statement.executeQuery(sql);

        //doc du lieu
        while (rs.next()){
            strTenDangNhap = rs.getString("TenDangNhap");
            strMatKhau = rs.getString("MatKhau");
            strTenNguoiDung = rs.getString("HoTen");
            bKetQuaDangNhap = true;
        }
        connection.close();
        if (bKetQuaDangNhap){
            this.setVisible(false);
        }
        else{
            JOptionPane.showMessageDialog(this, "Tên đăng nhập hoặc mật khẩu không đúng!");
        }
    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}
```

2.3.4. Xử lý sự kiện click menu Đăng nhập trong form chính

- Trong màn hình Design của frmMain > click phải chuột vào menu *Đăng nhập* > *Events* > *Action* > *actionPerformed*.
- Source code xử lý:

```
private void mnuDangNhapActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    frmDangNhap frm = new frmDangNhap(this, true);  
    frm.setVisible(true);  
    if (frm.getKetQuaDangNhap()) {  
        //kích hoạt các chức năng khi đăng nhập thành công  
        this.mnuHoaDon.setEnabled(true);  
        this.mnuLoaiSach.setEnabled(true);  
        this.mnuPhieuNhapSach.setEnabled(true);  
        this.mnuSach.setEnabled(true);  
        this.mnuDangXuat.setEnabled(true);  
        this.mnuDangNhap.setEnabled(false);  
    }  
}
```

2.3.5. Thêm đoạn code xử lý tắt các chức năng trên menu khi chưa đăng nhập Thêm trong phương thức khởi tạo của form frmMain

```
public Main() {  
    initComponents();  
    //Tắt các chức năng khi chưa đăng nhập  
    this.mnuHoaDon.setEnabled(false);  
    this.mnuLoaiSach.setEnabled(false);  
    this.mnuPhieuNhapSach.setEnabled(false);  
    this.mnuSach.setEnabled(false);  
    this.mnuDangXuat.setEnabled(false);  
}
```

2.3.6. Xử lý sự kiện đăng xuất trong form chính

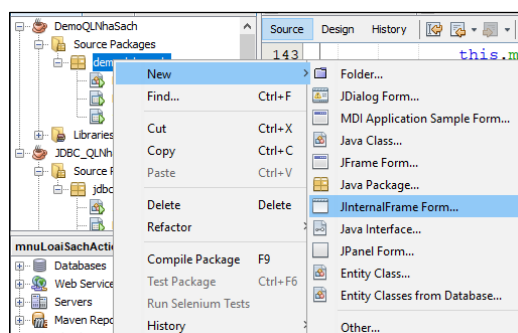
- Trong màn hình Design của frmMain > click phải chuột vào menu *Đăng xuất* > *Events* > *Action* > *actionPerformed*.
- Source code xử lý:

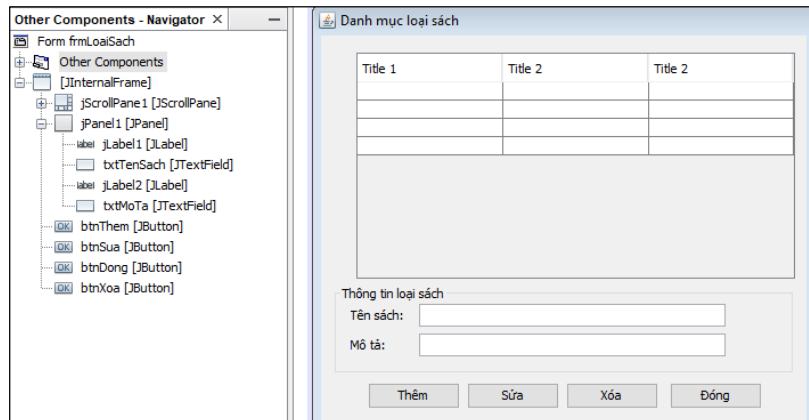
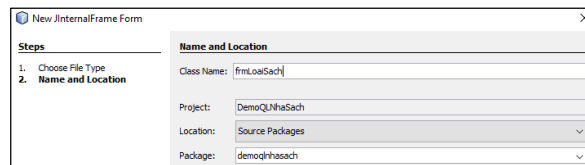
```
//Tắt các chức năng khi đăng xuất  
this.mnuHoaDon.setEnabled(false);  
this.mnuLoaiSach.setEnabled(false);  
this.mnuPhieuNhapSach.setEnabled(false);  
this.mnuSach.setEnabled(false);  
this.mnuDangXuat.setEnabled(false);  
this.mnuDangNhap.setEnabled(true);
```

2.4. Xử lý chức năng quản lý loại sách

2.4.1. Thiết kế màn hình quản lý loại sách

Chọn *New* > *JInternalFrame Form*





2.4.2. Xử lý sự kiện khi chọn menu quản lý loại sách trong form chính

```
frmLoaiSach frm = new frmLoaiSach();
frm.setVisible(true);
this.desktopPane.add(frm);
```

2.4.3. Hiện thị danh mục loại sách khi form quản lý loại sách hiển thị

Viết đoạn code sau trong phương thức khởi tạo của form frmLoaiSach.

```
String strServer = "ADMIN";
String strDatabase = "QL_NhaSach";
Connection connection = null;
Statement statement = null;
ResultSet rs = null;
try{
    Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
    String connectionURL = "jdbc:sqlserver://" + strServer
        + ":1433;databaseName=" + strDatabase
        + ";integratedSecurity = true";
    connection = DriverManager.getConnection(connectionURL);
    statement = connection.createStatement();
    String sql = "SELECT * FROM LoaiSach";
    rs = statement.executeQuery(sql);

    //Đọc dữ liệu
    DefaultTableModel dtm = new DefaultTableModel();
    dtm.addColumn("Mã sách");
    dtm.addColumn("Tên sách");
    dtm.addColumn("Mô tả");
    while (rs.next()){
        //Đọc từng dòng
        Vector vt = new Vector();
        vt.add(rs.getInt("LS_ID"));
        vt.add(rs.getString("Ten"));
        vt.add(rs.getString("MoTa"));
        dtm.addRow(vt); //Đưa vào dtm
    }
    //Gán dtm cho JTable
    this.tblDanhMucLoaiSach.setModel(dtm);
    connection.close();
}
catch(Exception ex){
    JOptionPane.showMessageDialog(this, ex.getMessage());
}
}
```

3. Bài tập

- Hoàn chỉnh phần hiển thị cho các màn hình sách, phiếu nhập sách, hóa đơn.
- Thực hiện các chức năng thêm, xóa, sửa của các màn hình.

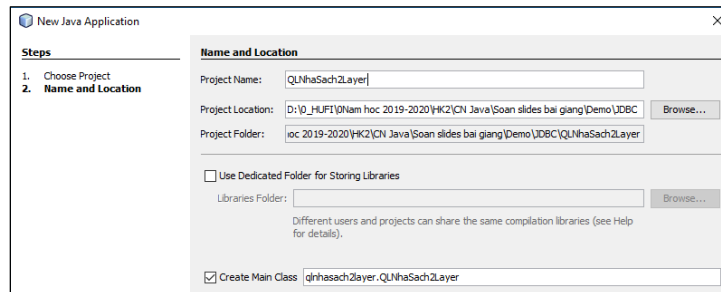
3.6. Xây dựng ứng dụng quản lý nhà sách theo mô hình đa tầng (2 *layer*)

1. Cơ sở dữ liệu

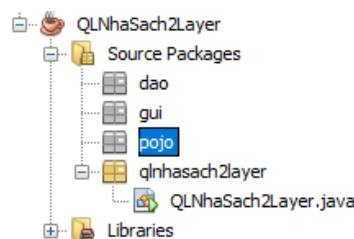
Như Cơ sở dữ liệu quản lý nhà sách ở bài tập 3.5.

2. Xây dựng ứng dụng

2.1. Tạo project



2.2. Tạo các package dao, gui, pojo



2.3. Tạo lớp truy xuất CSDL

2.3.1. Tạo lớp MySQLDataAccess trong package dao

```
public class MySQLDataAccess {
    private Connection connection;
    public void open() {

    }
    public void close() {

    }
    public ResultSet executeQuery(String sql) {
        return null;
    }
    public int executeUpdate(String sql) {
        return 0;
    }
}
```

2.3.2. Viết code cho các phương thức

```
public void open() {
    String strServer = "ADMIN";
    String strDatabase = "QL_NhaSach";
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        String connectionURL = "jdbc:sqlserver://" + strServer
            + ":1433;databaseName=" + strDatabase
            + ";integratedSecurity = true";
        connection = DriverManager.getConnection(connectionURL);
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

```

public void close() {
    try{
        this.connection.close();
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
}

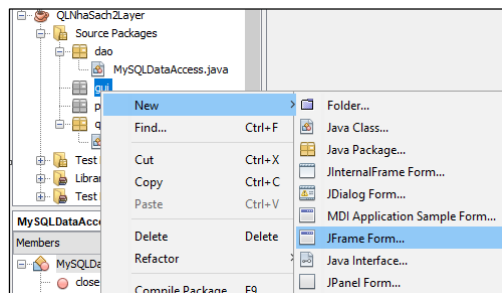
public ResultSet executeQuery(String sql){
    ResultSet rs = null;
    try{
        Statement sm = connection.createStatement();
        rs = sm.executeQuery(sql);
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    return rs;
}

public int executeUpdate(String sql){
    int n = -1;
    try{
        Statement sm = connection.createStatement();
        n = sm.executeUpdate(sql);
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    return n;
}

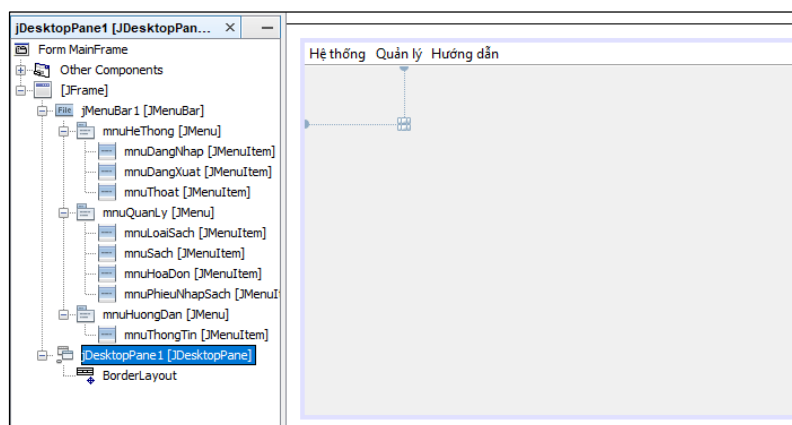
```

2.4. Tạo form chính của chương trình

- Tạo một JFrame Form trong package gui.



- Tạo một JDesktop Pane và hệ thống menu như ở bài tập 3.5.



- Hiển thị form chính khi chạy chương trình

```

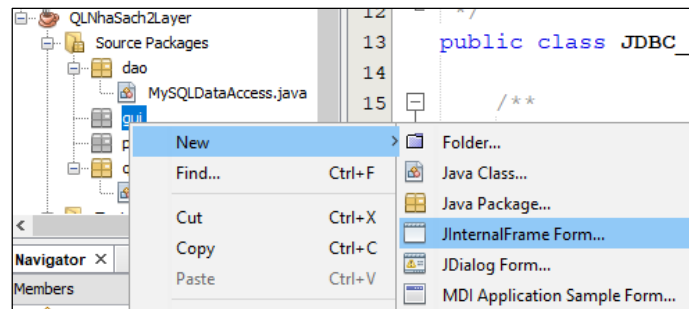
public static void main(String[] args) {
    // TODO code application logic here
    MainForm frm = new MainForm();
    frm.setVisible(true);
}

```

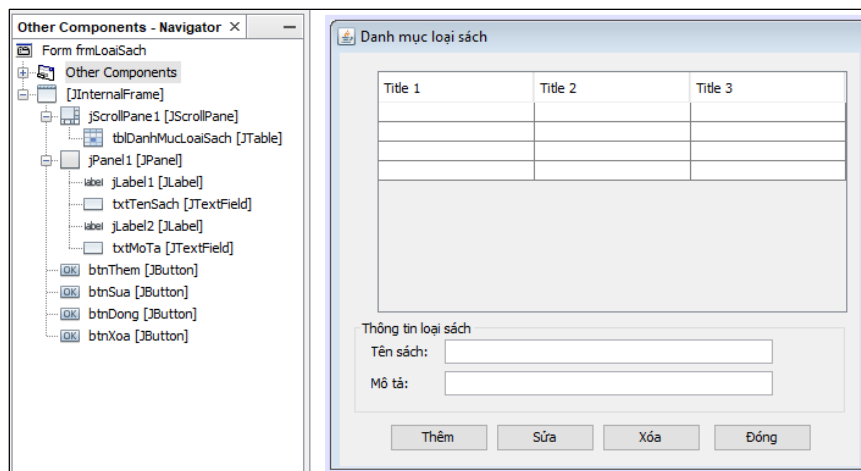
2.5. Xây dựng chức năng quản lý loại sách

2.5.1. Thiết kế màn hình

- Tạo một JFrame Form trong package gui



- Thiết kế màn hình như sau



2.5.2. Xử lý sự kiện chọn menu quản lý loại sách trong form chính

```
private void mnuLoaiSachActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    frmLoaiSach frm = new frmLoaiSach();  
    frm.setVisible(true);  
    this.jDesktopPanel1.add(frm);  
}
```

2.5.3. Hiện thị danh mục loại sách khi form quản lý loại sách hiển thị

- Tạo lớp LoaiSachPOJO trong package pojo

```
public class LoaiSachPOJO {  
  
    private int iID;  
    private String strTen;  
    private String strMoTa;  
  
    public LoaiSachPOJO() { }  
    public LoaiSachPOJO(int id, String ten, String mota) {  
        this.iID = id;  
        this.strTen = ten;  
        this.strMoTa = mota;  
    }  
    public LoaiSachPOJO(LoaiSachPOJO ls) {  
        this.iID = ls.iID;  
        this.strTen = ls.strTen;  
        this.strMoTa = ls.strMoTa;  
    }  
}
```

```

+ public int getiID() {...3 lines }
+ public void setiID(int iID) {...3 lines }
+ public String getStrTen() {...3 lines }
+ public void setStrTen(String strTen) {...3 lines }
+ public String getStrMoTa() {...3 lines }
+ public void setStrMoTa(String strMoTa) {...3 lines }
}

```

- Tạo lớp LoaiSachDAO trong package DAO

```

public class LoaiSachDAO {
    public static ArrayList<LoaiSachPOJO> layDanhSachLoaiSach() {
        ArrayList<LoaiSachPOJO> ds = new ArrayList<LoaiSachPOJO>();
        try{
            String sql = "SELECT * FROM LoaiSach";
            MySQLDataAccess helper = new MySQLDataAccess();
            helper.open();
            ResultSet rs = helper.executeQuery(sql);
            while(rs.next()) {
                LoaiSachPOJO ls = new LoaiSachPOJO();
                ls.setiID(rs.getInt("LS_ID"));
                ls.setStrTen(rs.getString("Ten"));
                ls.setStrMoTa(rs.getString("MoTa"));
                ds.add(ls);
            }
            helper.close();
        }
        catch(Exception ex){
            System.out.println(ex.getMessage());
        }
        return ds;
    }
}

```

- Source code hiển thị danh mục loại sách trên form frmLoaiSach

```

public frmLoaiSach() {
    initComponents();
    hienThi();
}

private void hienThi() {
    ArrayList<LoaiSachPOJO> ds = LoaiSachDAO.layDanhSachLoaiSach();

    DefaultTableModel dtm = new DefaultTableModel();
    dtm.addColumn("Mã sách");
    dtm.addColumn("Tên sách");
    dtm.addColumn("Mô tả");
    dtm.setNumRows(ds.size());

    for (int i = 0; i < ds.size(); i++){
        LoaiSachPOJO ls = ds.get(i);
        dtm.setValueAt(ls.getiID(), i, 0);
        dtm.setValueAt(ls.getStrTen(), i, 1);
        dtm.setValueAt(ls.getStrMoTa(), i, 2);
    }
    this.tblDanhMucLoaiSach.setModel(dtm);
}

```

3. Bài tập

- Hoàn chỉnh phần hiển thị cho các màn hình sách, phiếu nhập sách, hóa đơn.
- Thực hiện các chức năng thêm, xóa, sửa của các màn hình.

Chương 4. Lập trình xử lý đồng thời

4.1. Tạo class MyThread1 thực thi interface Runnable và thực hiện các yêu cầu sau:

- Phương thức run() chứa một vòng lặp in ra 10 số tự nhiên đầu tiên, mỗi lần in cách nhau 500 milliseconds.
- Tạo 2 object từ class MyThread1 là thread1 và thread2 với thứ tự ưu tiên tương ứng là MAX_PRIORITY và MIN_PRIORITY.
- Gọi phương thức start() cho cả 2 thread trên.

4.2. Tạo class MyThread2 và thực hiện các yêu cầu sau:

- Tạo thread1 và thread2.
- Thread1 sẽ in ra các số chẵn trong đoạn từ 1 đến 10, thread2 sẽ in ra các số lẻ trong đoạn từ 1 đến 10.
- Viết code sao cho chương trình chạy xong thread1 thì chạy tiếp thread 2. (thread1.start() → thread1.join() → thread2.start())

4.3. Viết chương trình minh họa việc chuyển khoản ngân hàng. Yêu cầu: đồng bộ giữa các lần rút tiền và gửi tiền.

4.4. Tiếp tục với bài chuyển khoản 4.3, yêu cầu thiết kế giao diện như hình dưới đây để thực hiện chức năng chuyển khoản. Bấm vào Start chương trình sẽ tiến hành chuyển khoản trên giao diện, khi bấm Stop chương trình sẽ dừng chuyển khoản.

