1. **MỤC TIÊU:**

* Tìm hiểu các thành phần giao diện cơ bản trong java:
  + JTextField.
  + JButton
  + JCheckbox
  + JRadioButton – ButtonGroup
  + JComboBox
  + JList – DefaultListModel
  + JTable – DefaultTableModel
  + JOptionPane

1. **TÓM TẮT LÝ THUYẾT:**
2. **JTextField:** 

* setText(String s): gán giá trị cho textbox
* getText(): lấy giá trị từ textbox

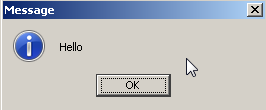
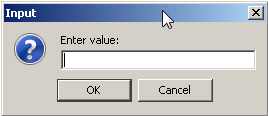
1. **JButton:** 

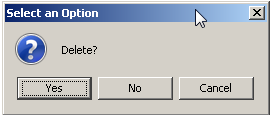
* Sự kiện **ActionListener**: xảy ra khi nút lệnh được nhấn

public void actionPerform(ActionEvent e) {

//Code xử lý sự kiện

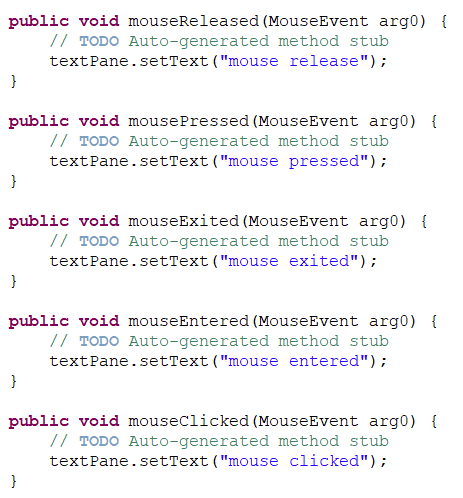
}

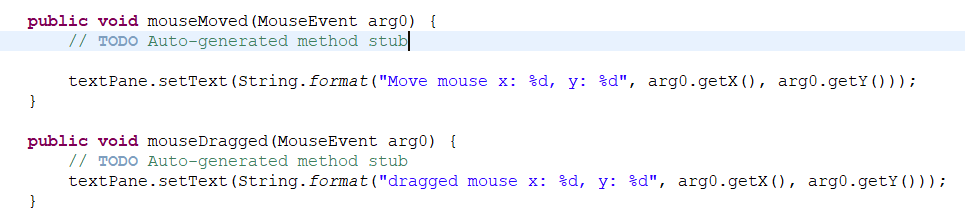
1. **JCheckBox:** 
   * setSelected(boolean b): check vào checkbox.
   * getSelected(): lấy trạng thái của checkbox : true/false
2. **JRadioButton – ButtonGroup:** 
   * setSelected(boolean b): check vào radiobutton.
   * getSelected(): lấy trạng thái của radiobutton: true/false
   * Các radiobutton phải được nhóm lại bằng đối tượng **ButtonGroup**  thì mới có thể chọn được.
     + add (AbstractButton button);
3. **JComboBox:** 
   * Hiển thị một danh sách các giá trị, chỉ được chọn một
   * getSelectedIndex(): trả về vị trí được chọn, 0 là vị trí đầu tiên, -1 là không có phần tử nào được chọn.
   * setSeletectedIndex(int index): thiết lập vị trí được chọn trong danh sách.
4. **JList – DefaultListModel:** 
   * Hiển thị một danh sách các giá trị, có thể chọn một hoặc nhiều
   * Hiện thực theo cơ chế MVC: View🡪JList, Model🡪DefaultListModel
   * JList:
     + setModel(ListModel model): gắn kết list với model.
     + getSelectedIndex(): trả về vị trí được chọn, 0 là vị trí đầu tiên, -1 nếu không có phần tử nào được chọn
     + setSelectedIndex(): đặt vị trí được chọn.
     + getSelectedItem(): trả về đối tượng được chọn
   * DefaultListModel:
     + add(Object o): thêm phần tử vào danh sách
     + remove(int index): xóa phần tử tại vị trí index
5. **JTable - DefaultTableModel:** 
   * Hiển thị một danh sách dưới dạng bảng, có thể chọn một hoặc nhiều
   * Hiện thực theo cơ chế MVC: View🡪JTable, Model🡪DefaultTableModel
   * JTable:
     + setModel(TableModel model): gắn kết table với model.
     + getSelectedRow(): trả về dòng được chọn, 0 là vị trí đầu tiên, -1 nếu không có dòng nào nào được chọn
   * DefaultTableModel:
     + addColoumn(Object o): thêm cột vào table.
     + addRow(Object[] rowData): thêm dòng dữ liệu vào table
     + removeRow(int index): xóa dòng tại vị trí index.
6. **JOptionPane:**
   * showMessageDialog():
   * showInputDialog():



* + showConfirmDialog():

1. **Mouse listener và mouse move event**
   * Để thêm event và move event trong swing ta sử dụng hàm addMouseListener và addMouseMotionListener và viết đè các hàm của lớp cha

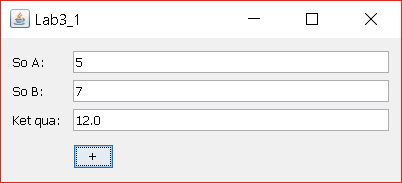




1. **NỘI DUNG THỰC HÀNH:**

Sinh viên import file lab3.zip, thực hiện các bài lab sau:

1. Mở **Lab3\_1.java:**

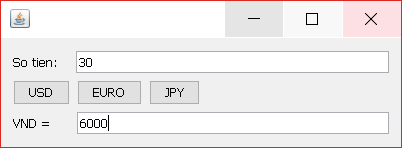


**Yêu cầu:**

* + Chạy chương trình, quan sát kết quả.
  + Hãy thiết kế thêm 3 button -, x, / . Viết code xử lý sự kiện cho các nút đó và xuất kết quả trong textbox.

**Bài tập 1:**

* + Thiết kế Frame như hình, viết code xử lý cho các nút trên frame. Biết rằng 1USD = 22800 VND, 1 EURO = 27800 VND, 1 JPY = 200 VND



1. Mở **Lab3\_2.java**:

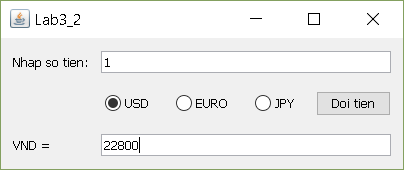


**Yêu cầu:**

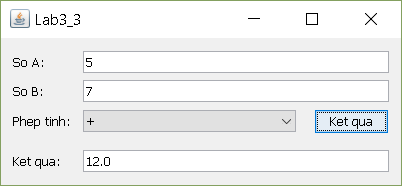
* + Chạy chương trình, quan sát kết quả.
  + Hãy viết code xử lý cho tùy chọn -, \*, /.

**Bài tập 2:**

* + Thiết kế Frame như hình, viết code xử lý cho nút Đổi tiền.



1. Mở **Lab3\_3.java:**

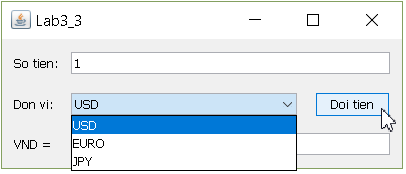


**Yêu cầu:**

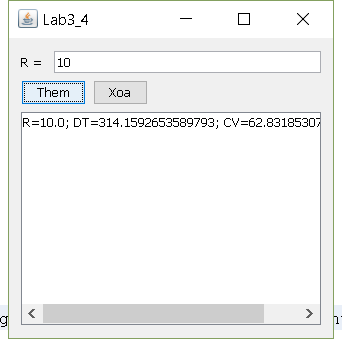
* + Chạy chương trình, quan sát kết quả
  + Hãy thêm vào các phép tính -, \*, / vào combobox, viết code xử lý cho các phép tính trên.

**Bài tập 3:**

* + Thiết kế Frame như hình, viết code xử lý cho nút Đổi tiền.



1. Mở **Lab3\_4.java**:



**Yêu cầu:**

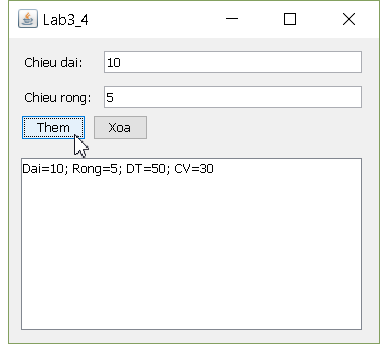
* + Chạy chương trình, quan sát kết quả
  + Hãy viết lệnh cho nút Xoa: xóa item đã chọn trong list.

*Hướng dẫn:*

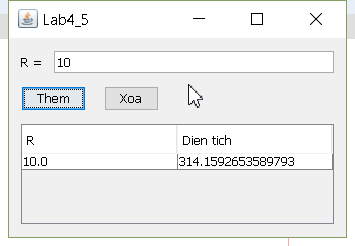
* + Dùng phương thức **getSelectedIndex**() của Jlist lấy vị trí item được chọn.
  + Dùng phương thức **remove**() của model xóa item tại vị trí được chọn.

**Bài tập 4:**

* + Tạo lớp **HinhChuNhat** có 2 thuộc tính, chiều dài, chiều rộng, có các phương thức getDientich(), getChuvi().
  + Thiết kế Frame như hình, viết code xử lý cho nút Them: thêm đối tượng HinhChuNhat vào JList và Xoa: xóa đối tượng đã chọn.



1. Mở **Lab3\_5.java**:



**Yêu cầu:**

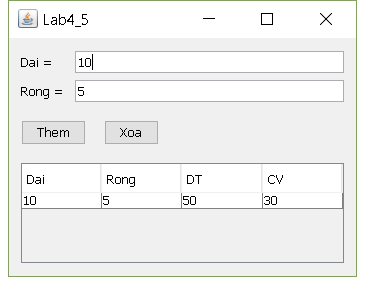
* + Chạy chương trình, quan sát kết quả
  + Hãy thêm cột Chu vi hiển thị chu vi đường tròn, viết lệnh cho nút Xoa: xóa item đã chọn trong list.

*Hướng dẫn:*

* + Dùng phương thức **getSelectedRow**() của Jlist lấy vị trí item được chọn.
  + Dùng phương thức **removeRow**() của model xóa item tại vị trí được chọn.
  + Dùng phương thức **remove()** của list để xóa dữ liệu trong List.

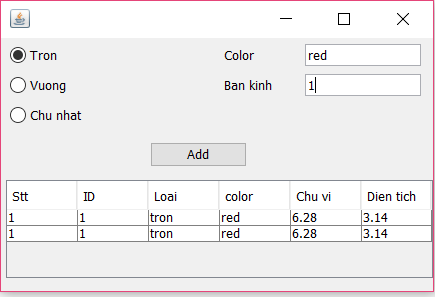
**Bài tập 5:**

* + Sử dụng lại lớp HinhChuNhat ở bài tập 4.
  + Thiết kế Frame như hình, viết code xử lý cho nút **Them**: thêm đối tượng HinhChuNhat vào JTable, nút **Xoa**: xóa đối tượng đã chọn.

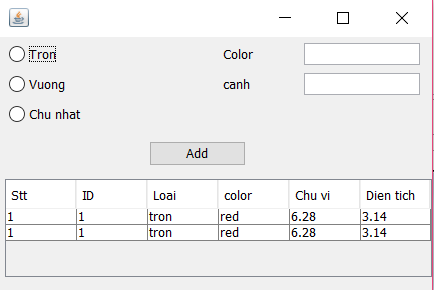


**Bài tập 6:** xây dựng chương trình có giao diện như sau:

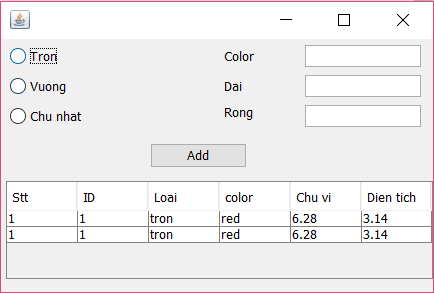
* Khi nhấn vào radiobutton “Tron” thì giao diện như hình



* Khi nhấn vào radio “Vuong” thì giao diện

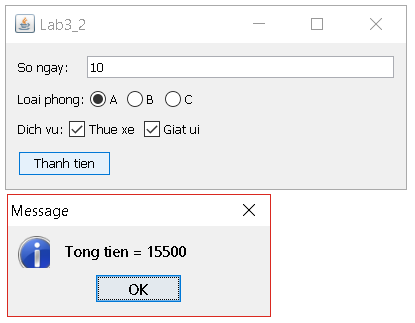


* Khi nhấn vào radio “Chu nhat” thì giao diện như sau:



* Khi nhấn button Add thì thêm vào Array<đối tượng> với :
* Stt tăng dần
* Id random
* Loại tùy thuộc vào radio button
* Màu lấy từ editText Color
* Chu vi là hàm tính toán biết chu vi của tròn, vuông, chữ nhật có công thức :2pi\*r, 4\*a, 2\*(dai+rong)
* Diện tích tương tự là: pi\*r^2,a^2,dai\*rong
* Viết các class cần thiết để viết chương trình này

1. **BÀI TẬP LÀM THÊM:**
2. Thiết kế frame như hình:



**Yêu cầu:**

* + Cách tính tiền phòng như sau:

*Loại phòng:* Loại A: 1500; Loại B: 1200; Loại C: 1000

*Dịch vụ:* 300 thuê xe, 200 giặt ủi

**Thành tiền = Giá loại phòng x số ngày + các dịch vụ đi kèm**

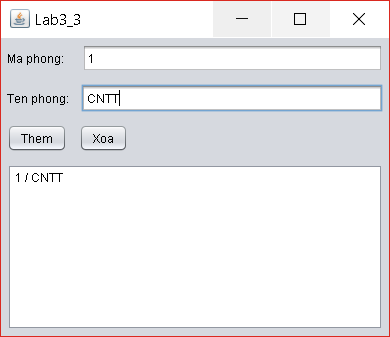
1. Tạo lớp PhongBan như sau:

**PhongBan**

- mapb: int

- tenphong: String

* + Thiết kế Frame như hình:



**Yêu cầu:**

* + Khi kích vào nút **Add**: tạo đối tượng PhongBan, thêm vào JList
  + Khi chọn đối tượng PhongBan trong Jlist, kích nút Xoa: xoa đối tượng đã chọn.

1. Tạo sơ đồ lớp như sau:

**PhongBan**

- mapb: int

- tenphong: String

**NhanVien**

- manv: int

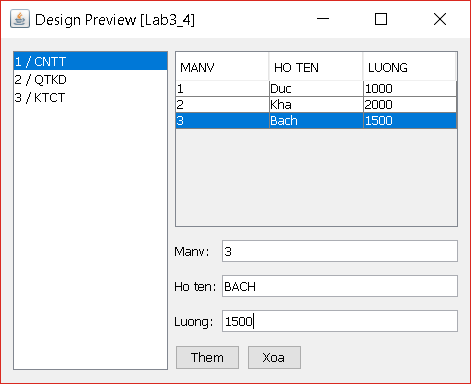
- hoten: String

- luong: double

**1**

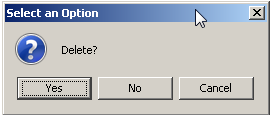
**n**

* + Thiết kế Frame như hình:



**Yêu cầu:**

* + Tạo danh sách PhongBan nạp vào Jlist
  + Khi chọn đối tượng PhongBan, nạp danh sách NhanVien vào Jtable
  + Khi kích nút **Them**: tạo đối tượng NhanVien, thêm vào PhongBan đã chọn.
  + Khi kích nút **Xoa** :
    - Xuất hiện cảnh báo trước khi xóa



* + - Nếu chọn Yes: xóa NhanVien đã chọn.

1. Một chương trình game võ lâm truyền kỳ có các lớp nhân vật như sau:
   * + Hệ kim, mộc, thủy, hỏa, thổ theo nguyên lý ngũ hành tương sinh tương khắc
     + Lớp nhân vật có chung những thuộc tính như sau: tên nhân vật, id, ngày tạo, đã xóa, chỉ số sức mạnh, thân pháp, nội công, sinh khí. Các chỉ số này lúc bắt đầu đều là 25
     + Các lớp hệ kim, mộc, thủy, hỏa, thổ có những thuộc tính của lớp nhân vật nhưng khác nhau về baselv. Chỉ số baselv như sau:
       - Hệ kim: mỗi lần lên lv có chỉ số baselv sức mạnh, thân pháp, nội công, sinh khí là: 2,1,1,1( nghĩa là mỗi lần lên lv thì sức mạnh +2, thân pháp +1, nội công + 1, sinh khí +1
       - Hệ mộc: baselv là (1,2,1,1)
       - Hệ thủy: baselv là (1,1,2,1)
       - Hệ hỏa : baselv là (1,1,1,2)
       - Hệ thổ : baselv là (1,1,2,2)
     + Viết hàm lên cấp với tham số đầu vào là 1-n level
     + Chỉ số tấn công của một lớp nhân vật gồm theo từng lớp nhân vật được tính như sau:
       - Kim tấn công :sức mạnh\*1+ 0.5\* thân pháp
       - Mộc tấn công :sức mạnh \*0.5 +1\*thân pháp
       - Thủy tấn công: nội công \*1 +sức manh \*0.5
       - Hỏa tấn công: sinh khí \*0.5+sức mạnh \*0.5 + thân pháp \*0.5
       - Thổ tấn công: sinh khí \*0.5 + sức manh \*0.5 +nội công \*0.5
       - Các nhân vật đánh vào lớp kị hệ bị giảm 60% đame và lớp nhân vật bị kị tăng 60% ( kim đánh vào mộc tăng 60%, kim đánh vào hỏa giảm 60% đame, vào hệ thủy, thổ đame là sức tấn công)
       - Viết 3 hàm tính đame tấn công tương ứng với một lớp nhân vật
     + Chỉ số máu của các lớp nhân vật như sau
       - Kim số máu : sinh khí \*8
       - Mộc số máu: sinh khí \*7
       - Thủy số máu: sinh khí \*6
       - Hỏa số máu:sinh khí \*7
       - Thổ số máu: sinh khí \*5 + nội công \*2
       - Viết hàm tính máu từng lớp nhân vật
     + Viết hàm tấn công với tham số đầu vào là id nhân vật khác trả về giá trị true nếu chiến thằng và false nếu thua, cách tính thắng thua như sau:
       - Một lần tấn công của lớp A và lớp B sẽ mất máu bằng dame tấn công(snếu kị hệ tăng 60%, bị kị hệ thì giảm 60%)
       - Số máu còn lại của nhân vật bị tấn công bằng máu còn lại = máu – dame(của nhân vật tấn công)
       - Khi máu nhân vật nào về 0 trước thì nhân vật đó thua- ngược lại thì thắng
       - Số lần tấn công phụ thuộc vào thân pháp của nhân vật, nhân vật nào thân pháp cao hơn thì tấn công trước và thân pháp cộng dồn như sau:

Ví dụ: nhân vật A tấn công 50, máu 1000, thân pháp 14

Nhân vật B tấn công 55, máu 1000, thân pháp 10

Lần 1 A tấn công B mất 50 máu, B tấn công A mất 55 máu , số máu còn lại A và B là 945,950. Thân pháp của A>B thân pháp còn lại của A = 14-10 =4

Lần 2 A tấn công B mất 50 máu, B tấn công A mất 55 máu, số máu còn lại của A và B là 890,900. Thân pháp của A = thân pháp còn lại lần trước + thân pháp = 4+14 =18, thân pháp của B =10 thân pháp còn lại là A =18 -10 =8 < thân pháp B =10

Lần 3 A tấn công B mất 50 máu, B tấn công A mất 55 máu, số máu còn lại A và B là 835,850. Thân pháp của A là thân pháp còn lại + thân pháp = 8 +14=22 > thân pháp B =10, thân pháp còn lại =22-10 =12>10(thân pháp B) A tấn công thêm lần nữa A tấn công B mất 50 máu số máu còn lại A và B là 835 và 800 thân pháp còn lại của A là 12 -10 =2 và tiếp tục cho đến khi một nhân vật có máu về 0. Giao diện xem hình dưới

