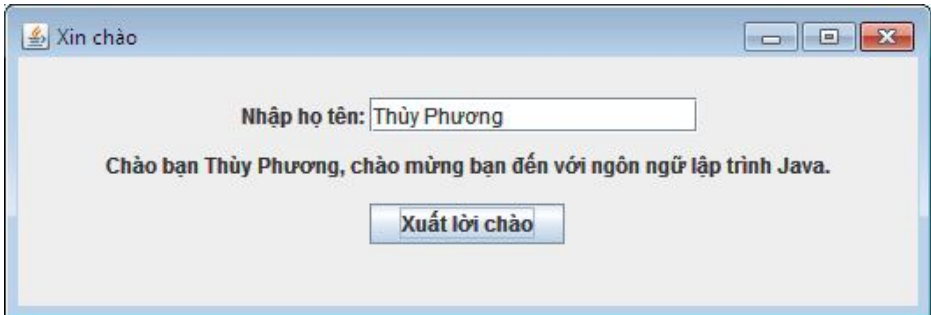


1.1. Xuất câu chào

- ✓ **Yêu cầu:** Xây dựng màn hình Xuất câu chào



- ✓ **Hướng dẫn sử dụng:**
 - Nhập họ tên Sau đó nhấn nút “Xuất câu chào”
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhXuatCauChao: FrmXuatCauChao (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtHoTen: JTextField (nhập liệu)
 - lblCauchao: JLabel (Kết xuất)
 - btnXuatCauChao: JButton
 - **Nhập:**
 - Họ tên
 - **Xuất:**
 - Câu chào
 - **Qui tắc xử lý:**
 - Không có
- ✓ **Thuật giải**
 - Khai báo biến HoTen nhận giá trị của txtHoTen

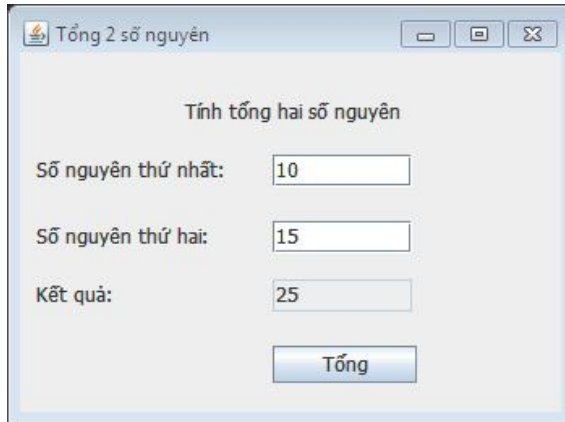
- Kết xuất Câu chào + HoTen

✓ **Hướng dẫn**

- Không có

1.2. Tính tổng hai số nguyên

- ✓ **Yêu cầu:** Xây dựng màn hình viết chương trình tính tổng hai số nguyên



Tổng 2 số nguyên

Tính tổng hai số nguyên

Số nguyên thứ nhất: 10

Số nguyên thứ hai: 15

Kết quả: 25

Tổng

- ✓ **Hướng dẫn sử dụng:**
 - Nhập số nguyên thứ nhất, số nguyên thứ hai sau đó nhấn nút "Tổng"
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhTong2SoNguyen: JFrame
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtSoNguyenThuNhat: JTextField (nhập liệu)
 - txtSoNguyenThuHai: JTextField (nhập liệu)
 - txtKetQua: JTextField (kết xuất, chỉ đọc)
 - btnTong: JButton (xử lý tính tổng hai số nguyên)
 - **Nhập:**
 - Số nguyên thứ nhất
 - Số nguyên thứ hai
 - **Xuất:**

- Kết quả (Tổng hai số nguyên)

- **Qui tắc xử lý:**

- Tổng hai số nguyên = Số nguyên thứ nhất + Số nguyên thứ hai

- ✓ **Thuật giải**

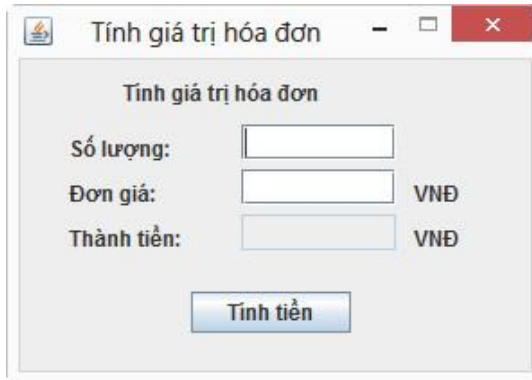
- Khai báo biến soNguyenThuNhat nhận giá trị của txtSoNguyenThuNhat
- Khai báo biến soNguyenThuHai nhận giá trị của txtSoNguyenThuHai
- Chuyển giá trị 2 chuỗi sang 2 số nguyên
- Khai báo biến tongHaiSoNguyen
- Xử lý tính Tổng hai số nguyên ($tongHaiSoNguyen = soNguyenThuNhat + soNguyenThuHai$)
- Kết xuất tongHaiSoNguyen ra txtKetQua

- ✓ **Hướng dẫn**

- Hàm chuyển từ chuỗi sang số nguyên:
Integer.parseInt(String)

1.3. Tính tiền hàng

- ✓ **Yêu cầu:** Xây dựng màn hình tính tiền hàng theo công thức:
Thành tiền = Số lượng * Đơn giá



The screenshot shows a Java Swing window titled "Tính giá trị hóa đơn". Inside the window, there is a label "Tính giá trị hóa đơn" at the top. Below it, there are three rows of input fields: "Số lượng:" followed by a text box, "Đơn giá:" followed by a text box and the unit "VNĐ", and "Thành tiền:" followed by a text box and the unit "VNĐ". At the bottom center of the window is a button labeled "Tính tiền".

- ✓ **Hướng dẫn sử dụng:**
 - Nhập số lượng và đơn giá. Nhấn nút "Tính tiền"
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhTinhTien: FrmTinhTien (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtSoLuong: JTextField (nhập liệu)
 - txtDonGia: JTextField (nhập liệu)
 - txtThanhTien: JTextField (kết xuất, chỉ đọc)
 - btnTinhTien: JButton (Xử lý tính Thành Tiền)
 - **Nhập:**
 - Số lượng
 - Đơn giá
 - **Xuất:**
 - Thành tiền
 - **Qui tắc xử lý:**

- Thành tiền = Số lượng x Đơn giá

✓ Thuật giải

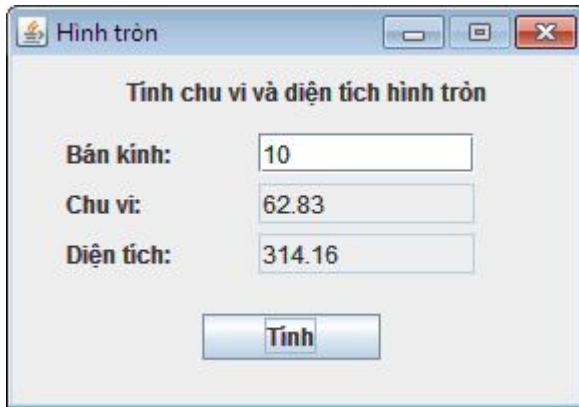
- Khai báo biến SoLuong nhận giá trị của txtSoLuong
- Khai báo biến DonGia nhận giá trị của txtDonGia
- Chuyển đổi dữ liệu từ kiểu chuỗi sang kiểu số
- Khai báo biến ThanhTien
- Xử lý tính thành tiền ($\text{ThanhTien} = \text{SoLuong} * \text{DonGia}$)
- Kết xuất ThanhTien ra txtThanhTien

✓ Hướng dẫn

- Hàm chuyển từ chuỗi sang số nguyên:
Integer.parseInt(String)

2.1. Tính diện tích và chu vi hình tròn

- ✓ **Yêu cầu:** Xây dựng màn hình tính diện tích và chu vi hình tròn



Hình tròn

Tính chu vi và diện tích hình tròn

Bán kính: 10

Chu vi: 62.83

Diện tích: 314.16

Tính

- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào bán kính R. Nhấn nút “Tính” => Diện tích và chu vi hình tròn được hiển thị
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - thManHinhTinhChuViDienTich: FrmTinhChuViDienTich (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtBanKinh: JTextField (nhập liệu)
 - txtChuVi: JTextField (kết xuất, chỉ đọc)
 - txtDienTich: JTextField (kết xuất, chỉ đọc)
 - btnTinh: JButton (Xử lý tính chu vi và diện tích)
 - **Nhập:**
 - Bán kính
 - **Xuất:**
 - Chu vi
 - Diện tích

▪ **Qui tắc xử lý :**

- Chu vi = $2 * \text{Bán kính} * \text{PI}$
- Diện tích = $\text{PI} * \text{Bán kính} * \text{Bán kính}$

✓ **Thuật giải**

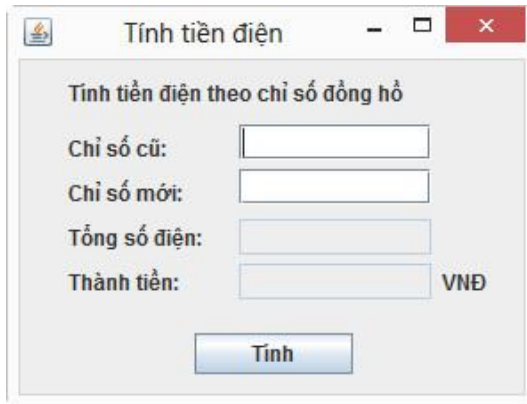
- Khai báo biến banKinh nhận giá trị của txtBanKinh
- Chuyển giá trị chuỗi thành số thực (float)
- Khai báo biến chuVi
- Khai báo biến dienTich
- Xử lý tính chu vi ($2 * \text{Bán kính} * \text{PI}$)
- Xử lý tính diện tích ($\text{PI} * \text{Bán kính} * \text{Bán kính}$)
- Kết xuất dữ liệu. Kết quả chu vi và diện tích là số thực (float), để xuất giá trị có 2 số thập phân ta phải format chuỗi xuất ra

✓ **Hướng dẫn**

- Giá trị PI: Java cung cấp hằng số Math.PI
- Hàm chuyển từ chuỗi sang số thực:
Float.parseFloat(String)
- Format chuỗi:
`String.format("%.2f", i ChuVi);`

2.2. Tính tiền điện

- ✓ **Yêu cầu:** Xây dựng màn hình tính tiền điện



- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào chỉ số cũ, chỉ số mới. Nhấn nút “Tính” => Hiển thị tổng số điện và tổng tiền phải trả.
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhTinhTienDien: FrmTinhTienDien(extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtChiSoCu: JTextField (nhập liệu)
 - txtChiSoMoi: JTextField (nhập liệu)
 - txtTongSoDien: JTextField (kết xuất, chỉ đọc)
 - txtThanhTien: JTextField (kết xuất, chỉ đọc)
 - btnTinh: JButton (Xử lý tính chu vi và diện tích)
 - **Nhập:**
 - Chỉ số cũ
 - Chỉ số mới
 - **Xuất:**

- Tổng số điện
- Thành tiền
- **Qui tắc xử lý :**
 - Tổng số điện = Chỉ số mới – chỉ số cũ
 - 50 số đầu tiên: 500vnd/số
 - Từ số 51 đến số 100: 1000vnd/số
 - Từ số 101 đến số 200: 2000vnd/số
 - Từ 201 đến số 300: 2500vnd/số
 - Trên 300: 4000vnd/số

✓ Thuật giải

- Khai báo biến chiSoCu nhận giá trị của txtChiSoCu
- Khai báo biến chiSoMoi nhận giá trị của txtChiSoMoi
- Chuyển giá trị kiểu chuỗi sang số thực (Float)
- Khai báo biến tongSoDien
- Xử lý tính tổng số điện (chiSoMoi - chiSoCu)
- Xử lý tính thành tiền (theo quy tắc xử lý)
- Kết xuất dữ liệu

✓ Hướng dẫn

- Tính thành tiền: kiểm tra điều kiện tổng số điện lần lượt theo các mức tiêu thụ 300, 200, 100, 50

```
private Float tinhTien(Float tongSoDien){
```

```
    float i TongSoDien = tongSoDien;
```

```
    float i ThanhTien = 0f;
```

```
    if(i TongSoDien > 300){
```

```
        i ThanhTien += (i TongSoDien - 300) * 4000;
```

```
        i TongSoDien = 300;
```

```
    }
```

```
if(i TongSoDi en > 200){  
    i ThanhTi en += (i TongSoDi en - 200) * 2500;  
    i TongSoDi en = 200;  
}
```

```
if(i TongSoDi en > 100){  
    i ThanhTi en += (i TongSoDi en - 100) * 2000;  
    i TongSoDi en = 100;  
}
```

```
if(i TongSoDi en > 50){  
    i ThanhTi en += (i TongSoDi en - 50) * 1000;  
    i TongSoDi en = 50;  
}
```

```
i ThanhTi en += i TongSoDi en * 500;
```

```
return i ThanhTi en;  
}
```

2.3. Giải Phương trình bậc I

- ✓ **Yêu cầu:** Xây dựng màn hình giải phương trình bậc nhất $ax + b = 0$, với $a \neq 0$



The screenshot shows a Java Swing window titled "Giải phương trình bậc I". Inside the window, there is a label "Giải phương trình $ax + b = 0$ ". Below this label, there are three input fields: "Nhập a:", "Nhập b:", and "Kết quả:". At the bottom of the window, there are two buttons: "Tính" and "Nhập lại".

- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào a, b. Nhấn nút "Giải PT" => Hiện thị nghiệm của phương trình.
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhGiaiPhuongTrinh: FrmGiaiPhuongTrinh JFrame
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtGiaTriA: JTextField (nhập liệu)
 - txtGiaTriB: JTextField (nhập liệu)
 - txtKetQua: JTextField (kết xuất, chỉ đọc)
 - btnTinh: JButton (Xử lý giá trị biểu thức)
 - btnNhapLai: JButton (Xử lý xóa hết giá trị trong các field)
 - **Nhập:**
 - Giá trị a
 - Giá trị b

- **Xuất:**

- Kết quả

- **Qui tắc xử lý :**

- $x = -b/a$

- ✓ **Thuật giải**

- Khai báo biến giaTriA nhận giá trị của txtGiaTriA
- Khai báo biến giaTriB nhận giá trị của txtGiaTriB
- Chuyển giá trị kiểu chuỗi thành số (Float)
- Khai báo biến ketQua
- Xử lý tính kết quả ($x = -b / a$)
- Kết xuất dữ liệu

- ✓ **Hướng dẫn**

- Không có

2.4. Tính giai thừa của một số

- ✓ **Yêu cầu:** Xây dựng màn hình tính giai thừa của một số



- ✓ **Hướng dẫn sử dụng:**

- Nhập vào x. Nhấn nút "Tính giai thừa" => Hiển thị kết quả

- ✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng:**

- frmManHinhTinhGiaiThua: FrmTinhGiaiThua (extends từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtGiaTriX: JTextField (nhập liệu)
- txtKetQua: JTextField (kết xuất, chỉ đọc)
- btnTinh: JButton (Xử lý giá trị biểu thức)
- btnNhapLai: JButton (Xử lý xóa hết giá trị trong các field)

- **Nhập:**

- Nhập giá trị x, là số nguyên dương

- **Xuất:**

- Kết quả tính giai thừa của x

- **Qui tắc xử lý :**

- $x! = 1 * 2 * 3 * \dots * (x - 1) * x$
- $0! = 1$

✓ Thuật giải

- Khai báo biến giaTriX nhận giá trị của txtGiaTriX
- Chuyển giá trị chuỗi sang số nguyên
- Khai báo biến ketQua
- Xử lý tính kết quả
- Kết xuất dữ liệu

✓ Hướng dẫn

```
private Long tinhGiaiThua(int x){
    Long kq = 1l;

    if(x < 0){
        return -1l;
    }

    for(int i = 1; i <= x; i++){
        kq = kq * i;
    }

    return kq;
}
```

2.5. Tách họ và tên trong chuỗi họ tên

- ✓ **Yêu cầu:** Xây dựng màn hình tách họ tên trong chuỗi họ tên



The screenshot shows a Java Swing window titled "Tách chuỗi họ tên". Inside the window, there is a label "Tách họ tên". Below this label, there are three input fields: "Nhập họ tên:", "Họ:", and "Tên:". At the bottom of the window, there is a button labeled "Tính".

- ✓ **Hướng dẫn sử dụng:**

- Nhập họ tên vào ô nhập liệu. Nhấn nút "Tính" => tách họ và tên và hiển thị ở ô "Họ" và ô "Tên"

- ✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng:**

- frmManHinhTachHoTen: FrmTachHoTen (extends từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtHoTen: JTextField (nhập liệu)
- txtHo: JTextField (kết xuất, chỉ đọc)
- txtTen: JTextField (kết xuất, chỉ đọc)
- btnTinh: JButton (Xử lý giá trị biểu thức)
- btnNhapLai: JButton (Xử lý xóa hết giá trị trong các field)

- **Nhập:**

- Họ tên

- **Xuất:**

- Họ

- Tên

- **Qui tắc xử lý :**

- Tìm khoảng trắng cuối cùng trong chuỗi, tách chuỗi họ tên theo vị trí khoảng trắng

- ✓ **Thuật giải**


- Khai báo biến hoTen nhận giá trị của txtHoTen
- Khai bao bien ho
- Khai bao bien ten
- Xử lý tính kết quả theo qui tắc xử lý và gán giá trị cho biến ho và biến ten
- Kết xuất dữ liệu biến ho, ten ra txtHo và txtTen

- ✓ **Hướng dẫn**

- Không có

2.6. Tìm kiếm chuỗi

- ✓ **Yêu cầu:** Xây dựng màn hình Tìm kiếm chuỗi



- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào chuỗi thứ nhất và chuỗi thứ hai. Nhấn nút “Tìm kiếm” => Xuất ra kết quả cho biết chuỗi thứ hai có nằm trong chuỗi thứ nhất hay không.
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhTimKiemChuoi: FrmTimKiemChuoi (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtChuoi1: JTextField (nhập liệu)
 - txtChuoi2: JTextField (nhập liệu)
 - txtKetQua: JTextField (kết xuất, chỉ đọc)
 - btnTiemKiem: JButton (Xử lý giá trị biểu thức)
 - **Nhập:**
 - Chuỗi 1
 - Chuỗi 2
 - **Xuất:**
 - Kết quả

- **Qui tắc xử lý :**

- ✓ **Thuật giải**

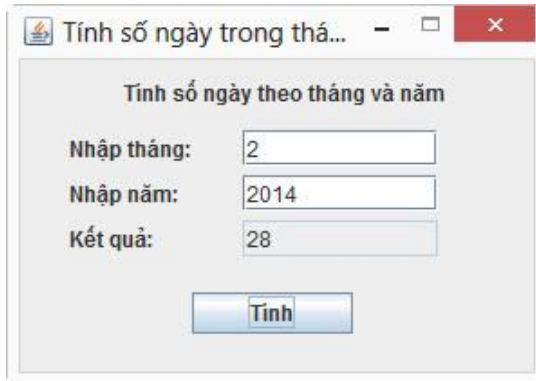
- Khai báo biến `chuoi1` nhận giá trị của `txtChuoil`
- Khai báo biến `chuoi2` nhận giá trị của `txtChuoil2`
- Khai báo biến `ketQua`
- Xử lý tìm kiếm chuỗi và gán giá trị cho biến `ketQua`
- Kết xuất dữ liệu `ketQua` ra `txtKetQua`

- ✓ **Hướng dẫn**

- Dùng hàm `contains` của lớp `String` để xử lý

2.7. Tính số ngày trong tháng

- ✓ **Yêu cầu:** Xây dựng màn hình Tính số ngày trong tháng



The screenshot shows a Java Swing window titled "Tính số ngày trong tháng...". Inside the window, there is a label "Tính số ngày theo tháng và năm". Below this label, there are three text input fields: "Nhập tháng:" with the value "2", "Nhập năm:" with the value "2014", and "Kết quả:" with the value "28". At the bottom center of the window is a button labeled "Tính".

- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào tháng và năm. Nhấn nút "Tính" => Hiện thị số ngày trong tháng
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhTinhNgayTrongThang: FrmTinhNgayTrongThang (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtThang: JTextField (nhập liệu)
 - txtNam: JTextField (nhập liệu)
 - txtKetQua: JTextField (kết xuất, chỉ đọc)
 - btnTinh: JButton (Xử lý giá trị biểu thức)
 - **Nhập:**
 - Tháng
 - Năm
 - **Xuất:**
 - Ngày trong tháng

▪ Qui tắc xử lý :

- Tháng 1, 3, 5, 7, 8, 10, 12 có 31 ngày
- Tháng 4, 6, 9, 11 có 30 ngày
- Tháng 2 năm thường có 28 ngày, năm nhuận có 29 ngày
- Năm nhuận là năm chia hết cho 4 và không chia hết cho 100 hoặc là năm chia hết cho 400

✓ Thuật giải

- Khai báo biến thang nhận giá trị của txtThang
- Khai báo biến nam nhận giá trị của txtNam
- Kiểm tra giá trị nhập
- Chuyển giá trị chuỗi sang số nguyên
- Khai báo biến ketQua
- Xử lý tính ngày theo tháng và năm sau đó gán giá trị cho biến ketQua. Lưu ý trường hợp năm nhuận.
- Kết xuất dữ liệu ra txtKetQua

✓ Hướng dẫn

Tính ngày trong tháng

```
private int tinhNgayTrongThang(int thang, int nam){  
    switch (thang) {  
        case 1: case 3: case 5: case 7: case 8: case 10: case  
12:  
            return 31;  
        case 4: case 6: case 9: case 11:  
            return 30;  
        case 2:  
            if(laNamNhuon(nam)){  
                return 29;  
            }  
            return 28;  
    }  
    return 0;  
}
```

Kiểm tra năm nhuận

```
private boolean laNamNhuon(int nam){
```

```
if((nam % 4 == 0 && nam % 100 != 0) || nam % 400 == 0){  
    return true;  
}  
return false;  
}
```

2.8. Tính ngày sinh

- ✓ **Yêu cầu: Xây dựng chương trình Tính ngày sinh**



Tính ngày sinh

Tính ngày sinh

Nhập ngày: 11

Nhập tháng: 1

Nhập năm: 2014

Hôm nay là ngày sinh của bạn. Chúc mừng sinh nhật!

Tính

- ✓ **Hướng dẫn sử dụng:**

- Nhập vào ngày, tháng, năm sinh. Nhấn nút “Tính” => hiển thị một trong các kết quả: Còn n ngày nữa là đến ngày sinh của bạn” hoặc “Ngày sinh của bạn đã qua n ngày” hoặc “Hôm nay là ngày sinh của bạn. Chúc mừng sinh nhật.”

- ✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng:**

- frmManHinhTinhNgaySinh: FrmTinhNgaySinh (extends từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtNgay: JTextField (nhập liệu)
- txtThang: JTextField (nhập liệu)
- txtNam: JTextField (nhập liệu)
- txtKetQua: JTextField (kết xuất, chỉ đọc)
- btnTinh: JButton (Xử lý giá trị biểu thức)

- **Nhập:**

- Ngày
- Tháng

- Năm
- **Xuất:**
 - Câu thông báo
- **Qui tắc xử lý :**
 - So sánh ngày user nhập với ngày của hệ thống

✓ Thuật giải

- Khai báo biến ngay nhận giá trị của txtNgay
- Khai báo biến thang nhận giá trị của txtThang
- Khai báo biến nam nhận giá trị của txtNam
- Kiểm tra giá trị nhập hợp lệ
- Chuyển chuỗi sang số nguyên
- Khai báo biến ketQua
- Xử lý tính ngày sinh so với ngày hệ thống sau đó gán giá trị vào biến ketQua
- Kết xuất dữ liệu vào txtKetQua

✓ Hướng dẫn

- Dùng đối tượng Calendar để thao tác ngày.
- Lấy ngày hệ thống:

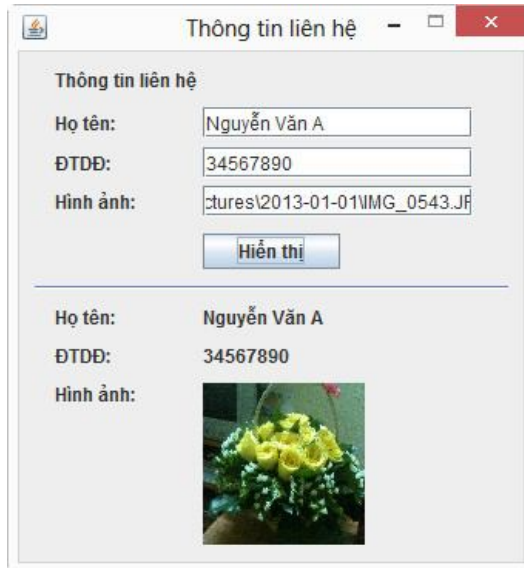

```
Calendar now = Calendar.getInstance();
```
- Tính ra số ngày tính từ năm 1200.


```
int nowDays = (int)(now.getTimeInMillis() / (1000 * 60 * 60 * 24));
```


3.1. Thông tin Liên hệ

- ✓ **Yêu cầu: Thiết kế màn hình hiển thị thông tin Liên hệ có các thông tin sau:**

- Tên
- ĐTDĐ
- Hình ảnh



The screenshot shows a Java Swing window titled "Thông tin liên hệ". It has a standard title bar with minimize, maximize, and close buttons. The window is divided into two main sections. The top section contains three text input fields with labels "Họ tên:", "ĐTDĐ:", and "Hình ảnh:". The fields contain the values "Nguyễn Văn A", "34567890", and "ctures\2013-01-01\VMG_0543.Jf" respectively. Below these fields is a button labeled "Hiển thị". The bottom section, separated by a horizontal line, displays the entered information: "Họ tên: Nguyễn Văn A", "ĐTDĐ: 34567890", and "Hình ảnh:" followed by a small image of a bouquet of yellow flowers.

- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào các ô nhập liệu các thông tin: tên, điện thoại di động, hình ảnh. Nhấn nút "Hiển thị" => Các thông tin vừa nhập sẽ được hiển thị.
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - ThManHinhHienThi: JFrame
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtTen: JTextField (Nhập liệu)
 - txtDtdd: JTextField (Nhập liệu)

- txtHinhAnh: JTextField (Nhập liệu)
- lblHienThiTen: JLabel (Read Only)
- lblHienThiDtdd: JLabel (Read Only)
- lblHienThiHinhAnh: JLabel (Read Only) (phải hiển thị hình ảnh)
- btnHienThi: JButton
- **Nhập:**
 - Tên
 - Điện thoại di động
 - Hình ảnh
- **Xuất:**
 - Tên
 - Điện thoại di động
 - Hình ảnh
- **Qui tắc xử lý:**
 - Không có

✓ Thuật giải

- Khai báo biến ten nhận giá trị của txtTen
- Khai báo biến dtdd nhận giá trị của txtDtdd
- Khai báo biến hinh nhận giá trị của txtHinhAnh
- Kiểm tra giá trị nhập hợp lệ
- Xử lý button hiển thị. Hiển thị hình ảnh dạng icon của JLabel
- Kết xuất dữ liệu

✓ Hướng dẫn

- Load hình ảnh:


```
I mageI con i con = new I mageI con("đường dẫn đến i mage");
I bl Hi nhAnh. setI con(i con);
```


3.2. Thiết kế màn hình Thêm mới Liên hệ

- ✓ **Yêu cầu:** Thiết kế màn hình Thêm mới Liên hệ có các thông tin sau:

- Tên
- ĐTDĐ
- Hình ảnh



The screenshot shows a standard Java Swing window with a title bar containing a small icon, the text 'Thông tin liên hệ', and standard window controls (minimize, maximize, close). The main content area has a light gray background. It features three labels on the left: 'Thông tin liên hệ' (in bold), 'Họ tên:', 'ĐTDD:', and 'Hình ảnh:'. To the right of each label is a white text input field. The 'Hình ảnh:' field is followed by a small blue button with three dots '...'. At the bottom center of the window is a blue button labeled 'Thêm'.

- ✓ **Hướng dẫn sử dụng:**
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhThem: FrmThemLienHe (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtTen: JTextField (nhập liệu)
 - txtDtdd: JTextField (nhập liệu)
 - fchHinhAnh: JFileChooser (chọn)
 - btnThem: JButton
 - **Nhập:**
 - Không có
 - **Xuất:**
 - Không có

▪ **Qui tắc xử lý :**

- Không có

✓ **Thuật giải**

- Không có

✓ **Hướng dẫn**

- Không có

3.3. Thiết kế màn hình Cập nhật liên hệ

- ✓ **Yêu cầu: Thiết kế màn hình Cập nhật Liên hệ có các thông tin sau:**

- Tên
- ĐTDĐ
- Hình ảnh

The screenshot shows a Java Swing window titled "Thông tin liên hệ". It has a standard title bar with minimize, maximize, and close buttons. The window is divided into two main sections by a horizontal line. The top section contains a label "Họ tên:" followed by a text input field and a "Tìm kiếm" button. The bottom section contains three labels: "Họ tên:", "ĐTDĐ:", and "Hình ảnh:", each followed by a text input field. The "Hình ảnh:" field has a small "..." button to its right. A "Thêm" button is located at the bottom center of the window.

- ✓ **Hướng dẫn sử dụng:**

- ✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng:**

- frmManHinhCapNhatLienHe: FrmCapNhatLienHe (extends từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTenTim: JTextField (nhập liệu)
- btnTim: JButton (Xử lý tìm)
- txtTen: JTextField (cập nhật)
- txtDtdd: JTextField (cập nhật)
- lblHinhAnh: JLabel (read only)
- fchHinhAnh: JFileChooser (chọn)
- btnCapNhat: JButton (Xử lý cập nhật)

- **Nhập:**

- Không có

- **Xuất:**

- Không có

- **Qui tắc xử lý:**

- Không có

- ✓ **Thuật giải**

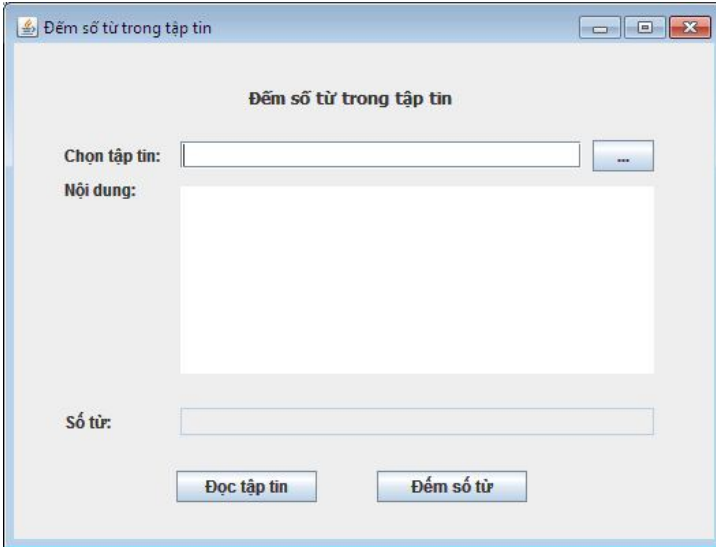
- Không có

- ✓ **Hướng dẫn**

- Không có

4.1. Đếm số từ trong tập tin văn bản

- ✓ **Yêu cầu:** xây dựng chương trình **Đếm số từ trong tập tin văn bản**



- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào đường dẫn của một tập tin văn bản. Nhấn nút “Đọc tập tin” => Hiển thị nội dung tập tin. Nhấn nút “Đếm số từ” => Hiển thị số từ của nội dung tập tin.
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhDocTapTin: FrmDocTapTin (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtTenTapTin: JTextField (nhập liệu) hoặc JFileChooser (chọn tập tin)
 - txtNoiDung: JTextArea (kết xuất, chỉ đọc)
 - txtSoTu: JTextField (kết xuất, chỉ đọc)
 - btnDoc: JButton (xử lý đọc và hiển thị)
 - btnDemSoTu: JButton (xử lý đếm và hiển thị)

- **Nhập:**
 - Chọn tập tin dạng text
- **Xuất:**
 - Nội dung tập tin
 - Số từ trong tập tin
- **Quy tắc xử lý :**
 - Đọc nội dung tập tin dạng text
 - Đếm các từ trong nội dung, các từ được phân cách bởi ký tự khoảng trắng, dấu phẩy, dấu chấm, dấu chấm phẩy, dấu hai chấm.

✓ Thuật giải

- Hiển thị đường dẫn file được chọn trong txtTenTapTin
- Khai báo biến nhận giá trị từ txtTenTapTin
- Đọc nội dung file, kết quả lưu vào biến noiDung
- Hiển thị noiDung lên txtNoiDung
- Xử lý đếm số từ theo quy tắc xử lý
- Kết quả số từ hiển thị trong txtSoTu

✓ Hướng dẫn

Đọc file dạng text

```
private String docNoiDungFile(File file) throws IOException{
    FileReader fr = new FileReader(file);
    char[] cbuf = new char[(int)file.length()];
    fr.read(cbuf);
    String noiDung = String.valueOf(cbuf);
    fr.close();
    return noiDung;
}
```

Dùng hàm split của String để cắt các từ trong một chuỗi

```
private int demSoTu(String text){
    String[] ls =
text.split("[\\s\\.\\,\\:\\\\;\\/\\\\\\\\\\\\\\\\]+");
    if(ls != null){
        return ls.length;
    }
    return 0;
}
```

4.2. Thêm Liên hệ vào tập tin

- ✓ **Yêu cầu:** Xây dựng chương trình Thêm mới Liên hệ có các thông tin sau:

- Tên
- ĐTDĐ
- Hình ảnh



- ✓ **Hướng dẫn sử dụng:**

- Nhập vào các ô nhập liệu các thông tin: tên, điện thoại di động, hình ảnh. Nhấn nút "Thêm" => Các thông tin vừa nhập sẽ được lưu vào tập tin LienHe.txt.

- ✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng (sử dụng giao diện đã thiết kế ở bài Giao diện 1):**

- frmManHinhThemLienHe: FrmThemLienHe (extends từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTen: JTextField (nhập liệu)
- txtDtdd: JTextField (nhập liệu)
- fchHinhAnh: JFileChooser (chọn)
- btnThem: JButton (Xử lý thêm)

- **Nhập:**

- Tên
- ĐTDĐ
- Hình ảnh...
- **Xuất:**
 - “Thông tin Liên hệ đã được thêm vào tập tin” nếu thêm được hoặc “Không thể thêm Liên hệ”
- **Qui tắc xử lý:**
 - Tạo tập tin LienHe
 - Lấy thông tin được nhập và ghi vào tập tin LienHe.txt

✓ Thuật giải

- Hiển thị đường dẫn file được chọn trong txtHinhAnh
- Khai báo biến hoTen nhận giá trị từ txtHoTen
- Khai báo biến dtdd nhận giá trị từ txtDtdd
- Khai báo biến hinhAnh nhận giá trị từ txtHinhAnh
- Xử lý ghi vào file các thông tin hoTen, dtdd, hinhAnh
- Xuất kết quả dưới dạng dialog thông báo

✓ Hướng dẫn

Ghi nội dung vào file

```
private void ghiVaoFile(String filePath,
    String noiDung) throws IOException{
    FileWriter fw = new FileWriter(filePath);
    fw.write(noiDung);
    fw.close();
}
```

Hiển thị thông báo

```
JOptionPane.showMessageDialog(null, "Thông tin Liên hệ đã
được thêm vào tập tin");
```

5.1. Tính tổng các phần tử trong mảng

- ✓ **Yêu cầu:** Viết chương trình Tính tổng các phần tử trong mảng có n phần tử, mỗi phần tử có giá trị ngẫu nhiên



- ✓ **Hướng dẫn sử dụng:**
 - Nhập vào số phần tử trong mảng n. Nhấn nút “Tính tổng”
=> Hiển thị mảng và tổng các phần tử trong mảng
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhTinhTong: FrmTinhTong (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtSoPhanTu: JTextField (nhập liệu)
 - txtMang: JTextArea (kết xuất, chỉ đọc)
 - txtTong: JTextField (kết xuất, chỉ đọc)
 - btnTinhTong: JButton (Xử lý tính tổng và hiển thị)
 - **Nhập:**
 - Nhập giá trị n
 - **Xuất:**
 - Mảng số nguyên ngẫu nhiên

- Xuất tổng
- **Qui tắc xử lý :**

- Không có

✓ **Thuật giải**

- Khai báo biến nhận giá trị nhập từ field txtSoPhanTu
- Chuyển giá trị nhập từ chuỗi sang số
- Phát sinh mảng ngẫu nhiên int[]
- Xuất giá trị chuỗi của mảng vào txtMang
- Tính tổng các phần tử trong mảng
- Hiển thị kết quả tính lên txtTong

✓ **Hướng dẫn**

Dùng đối tượng Random để phát sinh số ngẫu nhiên.

```
private int[] phatSi nhMangInt(int n){  
    int[] rs = new int[n];  
    Random random = new Random();  
    for(int i = 0; i < n; i++){  
        rs[i] = random.nextInt(100);  
    }  
  
    return rs;  
}
```

Xuất chuỗi của mảng

```
private String xuatMang(int[] mang){  
    String rs = "";  
    for (int i : mang) {  
        rs += i + " ";  
    }  
  
    return rs;  
}
```

Hàm tính tổng

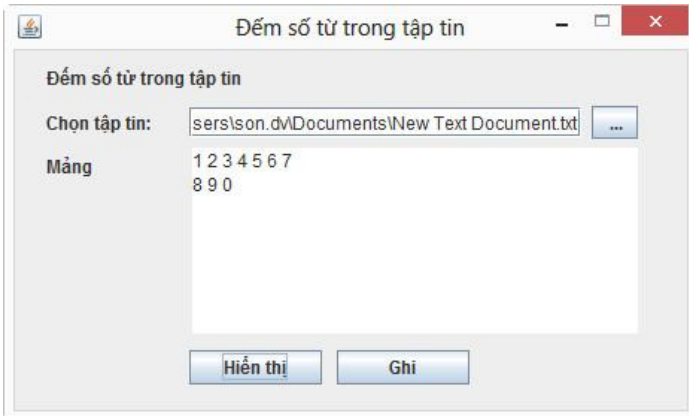
```
private long tingTong(int[] mang){  
    long tong = 0;  
    for (int i : mang) {
```

```
        tong += i;  
    }  
    return tong;  
}
```

5.2. Đọc và ghi mảng 1 chiều

✓ Yêu cầu: Viết chương trình Đọc và ghi mảng 1 chiều

- Cấu trúc file như sau: mảng được ghi vào tập tin, mỗi phần tử cách nhau bằng một khoảng trắng



✓ Hướng dẫn sử dụng:

- Chọn đường dẫn của tập tin chứa mảng đang có. Nhấn "Hiển thị" => Mảng được tạo từ tập tin sẽ hiển thị, kèm theo là thông báo "Mảng có ... phần tử"
- Nhập một mảng mới. Chọn tập tin sẽ chứa mảng mới. Nhấn "Ghi"

✓ Tóm tắt yêu cầu

▪ Thiết kế giao diện người dùng:

- ThManHinhTinhTong: JFrame

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTenTapTin: JTextField (nhập liệu)

- txtMang: JTextArea (Read Only)
- fchChonTapTin: JFileChooser
- btnHienThi: Jbutton (Xử lý hiển thị)
- btnGhi: Jbutton (Xử lý ghi file)
- **Nhập:**
 - Chọn file
 - Nhập giá trị mảng mới
- **Xuất:**
 - Hiển thị nội dung file
 - Ghi giá trị mảng vào file
- **Qui tắc xử lý :**
 - Đọc nội dung file, tách các phần tử trong nội dung theo khoảng trắng. Kiểm tra tính hợp lệ của phần tử (là số nguyên)
- ✓ **Thuật giải**
 - Hiển thị đường dẫn file được chọn vào txtTenTapTin
 - Đọc file được chọn
 - Tách các phần tử trong mảng, đếm phần tử
 - Hiển thị nội dung lên txtNoiDung
 - Hiển thị số phần tử dạng message box
 - Xử lý lưu nội dung vào file
- ✓ **Hướng dẫn**
 - Dùng hàm split của String để cắt các phần tử

```
private int kiemTraPhanTu(String text){
    String[] ls
    text.split("[\\s\\.,\\:\\;\\|\\\\\\\\]+");
    if(ls != null){
        for (String string : ls) {
            try{
```

```
        Integer.parseInt(string);
    }catch(Exception ex){
        return -1;
    }
}
return ls.length;
}
return 0;
}
```

5.3. Thêm Liên hệ vào tập tin (nâng cao)

✓ **Yêu cầu: Viết chương trình thêm Liên hệ vào tập tin chứa mảng các liên hệ**

- Cấu trúc file như sau: các phần tử Liên hệ sẽ được ghi vào tập tin có cấu trúc mảng như sau:
 - 1 phần tử là một chuỗi chứa 3 nội dung, mỗi nội dung cách nhau bằng một dấu "|"
 - Các phần tử khi ghi vào tập tin sẽ cách nhau bằng dấu enter xuống dòng



✓ **Hướng dẫn sử dụng:**

- Nhập các thông tin Tên, Điện thoại di động, Hình ảnh. Nhấn "Ghi"

✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng (sử dụng giao diện đã thiết kế ở bài Giao diện 1):**

- frmManHinhThemLienHe: FrmThemLienHe JFrame

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTen: JTextField (nhập liệu)
- txtDtdd: JTextField (nhập liệu)
- fchHinhAnh: JFileChooser (chọn)
- btnThem: JButton (Xử lý thêm)

- **Nhập:**

- Họ tên
- Điện thoại di động
- Hình ảnh

- **Xuất:**

- Ghi thông tin liên hệ vào file

- **Qui tắc xử lý :**

- Với mỗi thông tin nhập vào tạo thành một chuỗi (là 1 phần tử của mảng) có cấu trúc như đã mô tả ở trên.
- Lưu chuỗi này vào tập tin chứa mảng các Liên Hệ.

✓ **Thuật giải**

- Khai báo biến hoTen nhận giá trị từ txtHoTen
- Khai báo biến dtdd nhận giá trị từ txtDtdd
- Khai báo biến hìnhAnh nhận giá trị từ txtHinhAnh
- Tạo chuỗi kết hợp từ các thành phần hoTen, dtdd, hìnhAnh để lưu vào file
- Đọc nội dung file hiện tại
- Thêm nội dung mới vào cuối file
- Lưu nội dung vào file

✓ **Hướng dẫn**

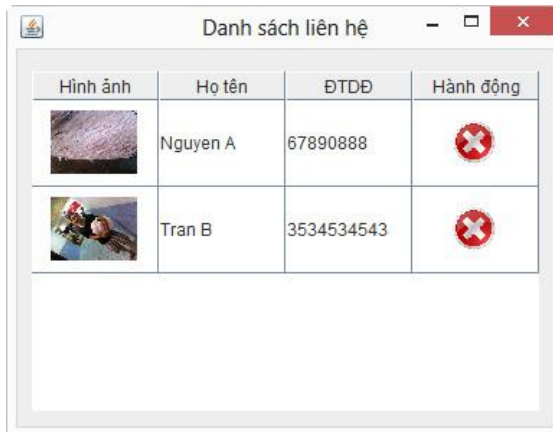
Hàm tạo chuỗi để thêm vào file

```
private String taoNoiDungGhiFile(String hoTen, String dtdd,
    String hinhAnh){
    String content = "";
    content += hoTen;
    content += "|";
    content += dtdd;
    content += "|";
    content += hinhAnh;

    return content;
}
```


6.1. Hiện thị danh sách các liên hệ

- ✓ **Yêu cầu: Viết chương trình hiện thị danh sách các liên hệ lên trên lưới.**
 - Gồm các thông tin sau: hình ảnh, tên, điện thoại di động, đằng sau mỗi liên hệ là button xóa, cho phép xóa liên hệ khi chọn.



- ✓ **Hướng dẫn sử dụng**
 - Không có
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmManHinhDanhSach: FrmDanhSachLienHe (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - tblDanhSachLienHe: JTable
 - btnXoa: Button
 - **Nhập:**
 - Không có
 - **Xuất:**

- Danh sách liên hệ
- **Qui tắc xử lý :**

- Không có

✓ **Thuật giải**

- Đọc nội dung file
- Phân tách các thành phần thành trong file
- Tạo nội dung hiển thị cho table, nội dung dưới dạng ma trận (Object[][]) mỗi dòng là một liên hệ, các thành phần mỗi dòng lần lượt là "Hình ảnh", "Họ tên", "ĐTĐD", "Hành động"
- Hiển thị hình ảnh bằng cách tạo TableCellRenderer cho cột "hình ảnh" và "hành động"

✓ **Hướng dẫn**

Tạo Render như sau:

```
public class ImageTableCellRenderer extends
DefaultTableCellRenderer {
    private static final long serialVersionUID =
        -7959113581100753271L;

    private int imageWidth;
    private int imageHeight;

    public ImageTableCellRenderer(int imageWidth, int
imageHeight) {
        this.imageWidth = imageWidth;
        this.imageHeight = imageHeight;
    }

    @Override
    public Component getTableCellRendererComponent(JTable
table,
        Object value, boolean isSelected, boolean
hasFocus,
        int row, int column) {

        super.getTableCellRendererComponent(table, value,
            isSelected, hasFocus, row, column);
```

```

        try {
            BufferedImage image = null;
            URL url = new URL("file:" + value);
            image = getScaledImages(ImageIO.read(url),
                                    imageWidth, imageHeight);
            setIcon(new ImageIcon(image));
            setHorizontalAlignment(JLabel.CENTER);
            setText("");
        } catch (Exception e) {
            setText("[No image]");
            e.printStackTrace();
        }

        return this;
    }
    ...

```

//

```

private BufferedImage getScaledImages(BufferedImage in,
                                       int WIDTH, int HEIGHT) {
    BufferedImage out = new BufferedImage(WIDTH, HEIGHT,
                                           BufferedImage.TYPE_INT_RGB);
    Graphics2D g2 = out.createGraphics();
    g2.setColor(Color.white);
    g2.fillRect(0, 0, WIDTH, HEIGHT);
    double width = in.getWidth();
    double height = in.getHeight();
    double xScale = WIDTH / width;
    double yScale = HEIGHT / height;
    double scale = 1.0;
    scale = Math.min(xScale, yScale); // scale to fit

    double x = (WIDTH - width * scale) / 2;
    double y = (HEIGHT - height * scale) / 2;
    AffineTransform at =
        AffineTransform.getTranslateInstance(x, y);
    at.scale(scale, scale);
    g2.drawRenderedImage(in, at);
    g2.dispose();

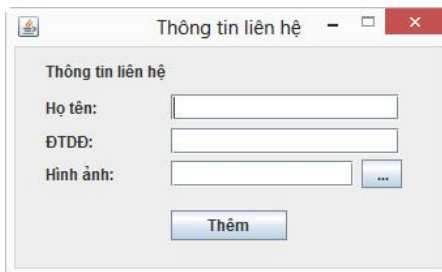
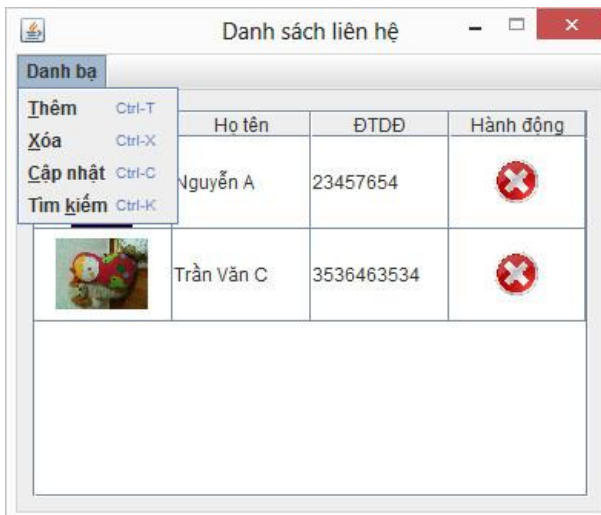
    return out;
}

```


}

6.2. Quản lý danh bạ điện thoại

- ✓ **Yêu cầu: Viết chương trình Quản lý danh bạ điện thoại.**
- Tương tự như bài trên như có thêm một menu “Danh bạ”, trong đó có các item: Thêm một liên hệ, Xóa một liên hệ, Tìm kiếm liên hệ, Cập nhật liên hệ.



✓ **Hướng dẫn sử dụng:**

- Xem thông tin các liên hệ trên màn hình chính.
- Chọn item nào trong menu Danh bạ thì sẽ hiển thị màn hình tương ứng.

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

- frmManHinhDanhSach: JFrame

* (Các thể hiện phía dưới đều nằm trong Frame)

- mnuDanhmuc: JMenu
- mnuIThem: JMenuItem
- mnuIXoa: JMenuItem
- mnuICapNhat: JMenuItem
- mnuITimKiem: JMenuItem
- tblDanhSachLienHe: JTable

▪ **Nhập:**

- Không có

▪ **Xuất:**

- Danh Sách liên hệ

▪ **Qui tắc xử lý :**

- Không có

✓ Thuật giải

- Đọc nội dung file
- Phân tách các thành phần thành trong file
- Tạo nội dung hiển thị cho table, nội dung dưới dạng ma trận (Object[][]) mỗi dòng là một liên hệ, các thành phần mỗi dòng lần lượt là "Hình ảnh", "Họ tên", "ĐTĐD", "Hành động"
- Hiển thị hình ảnh bằng cách tạo TableCellRender cho cột "hình ảnh" và "hành động"
- Chọn menu Thêm, hiển thị màn hình thêm liên hệ
- Xử lý thêm liên hệ, cập nhật nội dung file
- Chọn menu Xóa, thực hiện xóa liên hệ, cập nhật nội dung file
- Chọn menu Cập nhật, hiển thị màn hình cập nhật liên hệ
- Xử lý cập nhật liên hệ, cập nhật nội dung file
- Chọn menu Tìm kiếm, hiển thị màn hình tìm kiếm
- Xử lý tìm kiếm liên hệ

✓ Hướng dẫn

- Không có

7.1. Thực hiện tính toán hai phân số

- ✓ **Yêu cầu:** Xây dựng chương trình Tính toán hai phân số

Gồm có:

- phân số thứ nhất (tử số, mẫu số) và phân số thứ hai (tử số, mẫu số)
- combobox 4 phép tính: +, -, *, / cho người dùng chọn

The screenshot shows a Java Swing window titled "Phân số" (Fractions). It contains two columns for inputting fractions. The first column, labeled "Phân số 1", has a numerator field with "1" and a denominator field with "2". The second column, labeled "Phân số 2", has a numerator field with "1" and a denominator field with "3". Between the columns is a combobox with a "+" sign selected. Below these inputs is a "Kết quả:" (Result) field displaying "5/6". At the bottom of the window are two buttons: "Tính" (Calculate) and "Làm lại" (Reset).

- ✓ **Hướng dẫn sử dụng:**

- Nhập vào phân số thứ nhất (tử số, mẫu số) và phân số thứ 2 (tử số, mẫu số), chọn 1 trong 4 phép tính. Nhấn nút "Tính"
=> Hiển thị kết quả lên phân số kết quả (tử số, mẫu số)

- ✓ **Tóm tắt yêu cầu**

- **Thiết kế giao diện người dùng:**

- frmManHinhTinhToanPhanSo: FrmTinhToanPhanSo (extend từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtTuSo1: JTextField (nhập liệu)

- txtMauSo1: JTextField (nhập liệu)
- txtTuSo2: JTextField (nhập liệu)
- txtMauSo2: JTextField (nhập liệu)
- cmbPhepTinh: JComboBox (chọn)
- txtKetQua: JTextField (nhập liệu)
- btnTinh: JButton (Xử lý tính)
- btnLamLai: JButton (Xử lý làm lại)
- **Nhập:**
 - Tử số phân số 1
 - Mẫu số phân số 1
 - Tử số phân số 2
 - Mẫu số phân số 2
 - Chọn phép tính
- **Xuất:**
 - Kết quả tính toán 2 phân số
- **Qui tắc xử lý :**
 - Xây dựng class PhanSo
 - Xử lý tính toán trên hai đối tượng có kiểu là PhanSo
- ✓ **Thuật giải**
 - Khai báo biến tuSo1 nhận kết quả từ txtTuSo1
 - Khai báo biến mauSo1 nhận kết quả từ txtMauSo1
 - Khởi tạo phanSo1 (kiểu PhanSo) từ tuSo1, mauSo1
 - Khai báo biến tuSo2 nhận kết quả từ txtTuSo2
 - Khai báo biến mauSo2 nhận kết quả từ txtMauSo2
 - Khởi tạo phanSo2 (kiểu PhanSo) từ tuSo2, mauSo2
 - Khai báo biến phepTinh nhận giá trị từ cmbPhepTinh
 - Tính toán 2 phân số theo phép tính

✓ Hướng dẫn

Hàm tính toán 2 phân số theo phép tính

```
private PhanSo tinh(String phepToan, PhanSo phanSo1,
    PhanSo phanSo2){
    if("+".equals(phepToan)){
        return tinhTong(phanSo1, phanSo2);
    }else if("-".equals(phepToan)){
        return tinhHi eu(phanSo1, phanSo2);
    }else if("*".equals(phepToan)){
        return tinhTi ch(phanSo1, phanSo2);
    }else if("/".equals(phepToan)){
        return tinhThuong(phanSo1, phanSo2);
    }

    return null;
}
```

Các Hàm tính toán 2 phân số

```
private PhanSo tinhTong(PhanSo phanSo1, PhanSo phanSo2){

    int kqTuSo = phanSo1.getTuSo() * phanSo2.getMauSo()
        + phanSo2.getTuSo() * phanSo1.getMauSo();
    int kqMauSo = phanSo1.getMauSo() * phanSo2.getMauSo();

    PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
    return kq;
}

private PhanSo tinhHi eu(PhanSo phanSo1, PhanSo phanSo2){
    int kqTuSo = phanSo1.getTuSo() * phanSo2.getMauSo()
        - phanSo2.getTuSo() * phanSo1.getMauSo();
    int kqMauSo = phanSo1.getMauSo() * phanSo2.getMauSo();

    PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
    return kq;
}

private PhanSo tinhTi ch(PhanSo phanSo1, PhanSo phanSo2){
    int kqTuSo = phanSo1.getTuSo() * phanSo2.getTuSo();
    int kqMauSo = phanSo1.getMauSo() * phanSo2.getMauSo();

    PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
}
```

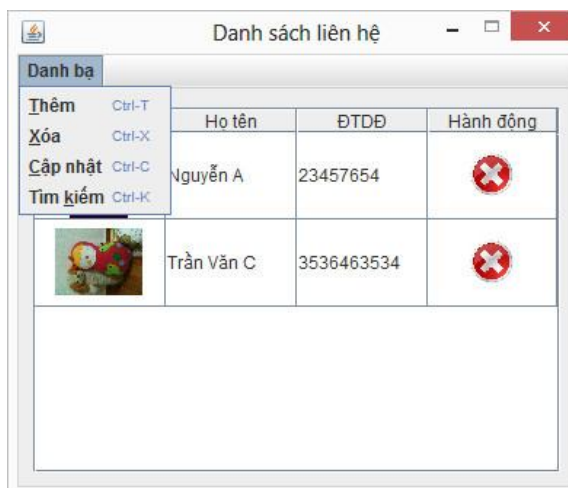
```
        return kq;
    }

    private PhanSo tinhThuong(PhanSo phanSo1, PhanSo phanSo2){
        int kqTuSo = phanSo1.getTuSo() * phanSo2.getMauSo();
        int kqMauSo = phanSo1.getMauSo() * phanSo2.getTuSo();

        PhanSo kq = new PhanSo(kqTuSo, kqMauSo);
        return kq;
    }
```


7.2. Quản lý Danh bạ thoại

- ✓ **Yêu cầu: Xây dựng chương trình Quản lý danh bạ điện thoại**
 - Tương tự như bài “Quản lý danh bạ điện thoại” của chương 6, nhưng nội dung các phần tử trong mảng Liên hệ của tập tin khi được đọc sẽ tái cấu trúc và gán vào một đối tượng của class LienHe.
 - Tất cả các liên hệ sẽ được quản lý bằng mảng các phần tử có kiểu class LienHe



- ✓ **Hướng dẫn sử dụng:**
 - Không có
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - Tương tự 6.2
 - **Nhập:**
 - Không có
 - **Xuất:**
 - Danh sách liên hệ

- **Qui tắc xử lý :**

- Xây dựng class LienHe
- Xử lý các đối tượng của class LienHe trong mảng.

- ✓ **Thuật giải**

- Không có

- ✓ **Hướng dẫn**

- Không có

7.3. Hóa đơn bán hàng

- ✓ **Yêu cầu:** Xây dựng chương trình nhập xuất thông tin hóa đơn bán hàng của một siêu thị

- Thông tin hóa đơn gồm có: số hóa đơn, ngày lập hóa đơn, danh sách hàng hóa được mua trong hóa đơn
- Thông tin hàng hóa gồm có: tên hàng hóa, đơn vị tính, đơn giá, số lượng mua
- Xây dựng chương trình có giao diện như sau, nhập thông tin hóa đơn và xuất thông tin hóa đơn ra màn hình

The screenshot shows a software window titled "Hóa đơn bán hàng" (Sales Invoice). It is divided into two main sections: "Hóa đơn bán hàng" on the left and "Thông tin hàng hóa" (Goods Information) on the right. The left section contains three input fields: "Số hóa đơn:" (Invoice Number) with the value "HD01", "Ngày hóa đơn:" (Invoice Date) with the value "17/01/2014", and "Số lượng hàng hóa:" (Quantity of goods) with the value "3". Below these fields are two buttons: "Thêm hóa đơn" (Add invoice) and "Thêm hàng hóa" (Add goods). The right section contains three input fields: "Tên hàng hóa:" (Goods name), "Đơn vị tính:" (Unit), and "Đơn giá:" (Unit price), each followed by an empty input box. Below these is another empty input box labeled "Số lượng:" (Quantity). At the bottom of the window, there is a table with 5 columns: "Tên HH" (Goods name), "Đơn vị tính" (Unit), "Đơn giá" (Unit price), "Số lượng" (Quantity), and "Thành tiền" (Amount). The table contains three rows of data: "Sữa tươi" (Fresh milk) with unit "Hộp" (Box), price 6000, quantity 10, and amount 60000; "Bánh mì" (Bread) with unit "ổ" (Loaf), price 3000, quantity 5, and amount 15000; and "Phô mai" (Cheese) with unit "Hộp" (Box), price 30000, quantity 1, and amount 30000. Below the table are two buttons: "Xuất hóa đơn" (Print invoice) and "Làm lại" (Reset).

Tên HH	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Sữa tươi	Hộp	6000	10	60000
Bánh mì	ổ	3000	5	15000
Phô mai	Hộp	30000	1	30000

Số HĐ: HD01

Ngày HĐ: 17/01/2014

Danh sách hàng hóa:

Tên HH	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Sữa tươi	Hộp	6000	10	60000
Bánh mì	ổ	3000	5	15000
Phô mai	Hộp	30000	1	30000

OK

✓ Hướng dẫn sử dụng:

- Nhập thông tin hóa đơn và thông tin các hàng hóa trong hóa đơn. Nhấn nút "Xuất hóa đơn" để xem thông tin của hóa đơn.
- Nhấn nút "Làm lại" để xóa các thông tin trên màn hình

✓ Tóm tắt yêu cầu

▪ Thiết kế giao diện người dùng:

- Thiết kế màn hình Nhập hóa đơn bán hàng

frmManHinhNhapHoaDon: FrmNhapHoaDon (extend từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- txtSoHoaDon: JTextField (nhập liệu)
- txtNgayHoaDon: JTextField (nhập liệu)
- txtSoLuongHoaDon: JTextField (nhập liệu)
- txtTenHangHoa: JTextField (nhập liệu)
- txtDonViTinh: JTextField (nhập liệu)
- txtDonGia: JTextField (nhập liệu)
- txtSoLuong: JTextField (nhập liệu)
- btnThemHoaDon: JButton (xử lý thêm hóa đơn)
- btnThemHangHoa: JButton (xử lý thêm hàng hóa)
- btnXuatHoaDon: JButton (xử lý xuất hóa đơn)

- btnLamLai: JButton (xử lý làm lại)
- Thiết kế màn hình Thông tin hóa đơn
- dlgHienThiHoaDon:DlgHienThiHoaDon (extend từ JDialog)
- * (Các thể hiện phía dưới đều nằm trong Frame)
 - lblSoHoaDon: JLabel (hiển thị)
 - lblNgayHoaDon: JLabel (hiển thị)
 - tblDSHangHoa: JTable (hiển thị)
 - btnOK: JButton (xử lý đóng màn hình)
- **Nhập:**
 - Nhập các thông tin hóa đơn, hàng hóa
- **Xuất:**
 - Thông tin hóa đơn
- **Qui tắc xử lý :**
 - Xây dựng class HoaDon và class HangHoa
 - Xử lý class.
- ✓ **Thuật giải**
 - Không có
- ✓ **Hướng dẫn**
 - Không có

8.1. Đổi ngoại tệ

✓ Yêu cầu: Viết chương trình **Đổi ngoại tệ**

- Gọi sử dụng Dịch vụ “Tỷ giá” trên Internet

Mã	Quốc kỳ	Giá mua	Giá bán
CNY		3370.0	3330.0
JPY		200.7	203.0
SGD		16430.0	16600.0
HKD		2700.0	2730.0
PNJ_DAB		3276000.0	3316000.0
CAD		19100.0	19320.0
AUD		18400.0	18600.0
NZD		0.0	0.0

Nhập số tiền:

Loại ngoại tệ: CNY ▼

Thành tiền VNĐ: Mua

Bán

✓ Hướng dẫn sử dụng:

- Nhập vào số tiền (USD, Yên, Euro), chọn Tỷ giá. Nhấn “Tính tiền” => Hiển thị thành tiền VNĐ tương ứng với số tiền đã nhập.

✓ Tóm tắt yêu cầu

▪ Thiết kế giao diện người dùng:

frmManHinhTyGia: FrmTyGia (extend từ JFrame)

* (Các thể hiện phía dưới đều nằm trong Frame)

- tblDsTyGia: JTable (hiển thị)
- txtNhapTien: JTextField (nhập liệu)

- cmbLoaiTienTe: JComboBox (chọn giá trị)
- txtThanhTienVNDMua: JTextField (hiển thị)
- txtThanhTienVNDBan: JTextField (hiển thị)
- txtTinhTien: JButton (xử lý tính tiền VND)
- **Nhập:**
 - Giá trị tiền cần qui đổi
 - Chọn loại tiền tệ
- **Xuất:**
 - Giá mua, giá bán theo loại tiền tệ
- **Qui tắc xử lý :**
 - Thành tiền = giá ngoại tệ * giá trị nhập

✓ Thuật giải

- Lấy thông tin tỷ giá từ <http://dongabank.com.vn/exchange/export>
- Xử lý kết quả trả về, khai báo biến mapTyGia chứa kết quả xử lý
- Hiển thị kết quả lên table
- Khai báo biến nhận giá trị từ txtNhapGiaTri
- Khai báo biến loiTienTe nhận giá trị từ cmbLoaiTienTe
- Xử lý tính giá mua, giá bán theo loại tiền tệ

✓ Hướng dẫn

Hàm lấy thông tin từ request url

```
private String layTyGia() throws IOException {
    URL tyGiaUrl = new
    URL("http://dongabank.com.vn/exchange/export");
    URLConnection yc = tyGiaUrl.openConnection();
    BufferedReader in = new BufferedReader(new
    InputStreamReader(yc.getInputStream()));
    String input = in.readLine();

    in.close();
}
```

```
    return input;  
}
```