

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO MÔN NGÔN NGỮ LẬP TRÌNH JAVA**  
**ĐỀ TÀI: ỨNG DỤNG XỬ LÝ ẢNH TRONG JAVA**

**Lớp:** SE330.I21

**Giảng viên hướng dẫn:** ThS. Huỳnh Tuấn Anh

**Sinh viên thực hiện:** Lý Bảo Khang – 15520343  
Mai Quốc Kiệt - 15520400

# LỜI CẢM ƠN

Chúng em xin trân trọng cảm ơn thầy Huỳnh Tuấn Anh, giảng viên môn ngôn ngữ lập trình Java, đã tận tình giúp đỡ và hướng dẫn chúng em trong suốt quá trình thực hiện đồ án. Tuy nhiên, trong quá trình làm bài, do thời gian và khả năng của bản thân còn hạn chế, nên chúng em không thể tránh khỏi sai sót. Vì vậy, chúng em mong được sự góp ý bổ sung của thầy để hoàn thiện đồ án của mình tốt hơn.

**Chúng em xin chân thành cảm ơn thầy và chúc thầy gặp nhiều thành công trong cuộc sống!**

# Mục lục

<b>CHƯƠNG 1: GIỚI THIỆU.....</b>	<b>4</b>
1. Tóm tắt về đồ án: .....	4
2. Mục đích và phạm vi sử dụng:.....	4
3. Ý tưởng và khó khăn: .....	4
<b>CHƯƠNG 2: TỔ CHỨC ĐỒ ÁN .....</b>	<b>5</b>
1. Giao tiếp, truyền thông:.....	5
2. Cấu trúc nhóm: .....	5
3. Vai trò: .....	5
<b>CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ .....</b>	<b>6</b>
1. Phần mềm và thư viện sử dụng: .....	6
2. Các chức năng trong chương trình: .....	6
3. Cách xử lý các chức năng:.....	6
<b>CHƯƠNG 4: THỰC HIỆN.....</b>	<b>7</b>
1. Giới thiệu giao diện:.....	7
2. Các thành phần trong giao diện: .....	7
2.1. MenuBar: .....	7
2.2. Phần để xử lý ảnh bằng bộ lọc: .....	8
2.3. Phần thể hiện ảnh và làm việc với ảnh:.....	10
2.4. Phần thêm chi tiết vào ảnh:.....	10
<b>CHƯƠNG 5: KẾT LUẬN.....</b>	<b>12</b>
1. Kết quả:.....	12
2. Một số mặt hạn chế:.....	12
3. Đánh giá thành viên:.....	12
4. Tài liệu tham khảo: .....	12

# CHƯƠNG 1: GIỚI THIỆU

## 1. Tóm tắt về đồ án:

Trong quá trình học hỏi và tìm hiểu về chuyên ngành, chúng em thấy vấn đề xử lý ảnh là rất thú vị và cũng khá phổ biến trong cuộc sống hằng ngày.

Vì vậy chúng em phát triển một phần mềm bằng ngôn ngữ Java để có thể xử lý ảnh với các bộ lọc cơ bản, cắt, ... cùng với một số chức năng cơ bản như của Microsoft Paint để phục vụ cho việc xử lý cũng như trang trí ảnh.

## 2. Mục đích và phạm vi sử dụng:

Mục đích sử dụng: giúp cho người dùng có thể thực hiện các chức năng xử lý ảnh cơ bản như cắt, lọc, chèn chữ ... trên ảnh

Phạm vi sử dụng: phù hợp với mọi phạm vi người dùng

## 3. Ý tưởng và khó khăn:

Ý tưởng: sử dụng thư viện Graphics2d để thực hiện các chức năng và áp dụng các kiến thức đã học về xử lý ảnh cũng như các kiến thức mới về Java

Khó khăn:

Kiến thức các thành viên trong nhóm còn có hạn, nhất là chỉ mới làm quen với ngôn ngữ lập trình Java, nên code còn khá tốn thời gian, chưa được tối ưu cũng như có thể có bug không mong muốn

Các thành viên trong nhóm có thời gian học tập khác nhau, nên tìm được ít khoảng thời gian hợp lý để có thể họp nhóm

Lịch trình:

Nhóm tổ chức thành các giai đoạn như sau:

Giai đoạn	Thời gian
Tìm hiểu	1 tuần
Thiết kế giao diện	2 tuần
Thực hiện các chức năng	4 tuần
Kiểm tra và sửa lỗi	1 tuần
Hoàn thành đồ án và viết báo cáo	2 ngày

## CHƯƠNG 2: TỔ CHỨC ĐỒ ÁN

### 1. Giao tiếp, truyền thông:

STT	Nội dung	Hình thức	Ghi chú
1	Họp nhóm để tổng kết sau các giai đoạn và lên kế hoạch cho giai đoạn tiếp theo	Gặp mặt	Sau khi hoàn thành từng giai đoạn Họp 2-4h/lần
2	Báo cáo tiến độ công việc mỗi ngày	Online	Trao đổi trên Facebook
3	Báo cáo tổng kết tuần	Gặp mặt	Sáng thứ 7 mỗi tuần

### 2. Cấu trúc nhóm:

MSSV	Họ và tên	Email	Chức vụ
15520343	Lý Bảo Khang	15520343@gm.uit.edu.vn	Nhóm trưởng
15520400	Mai Quốc Kiệt	15520400@gm.uit.edu.vn	Thành viên

### 3. Vai trò:

Thành viên	Vai trò
<b>Lý Bảo Khang</b>	Là người quản lý các hoạt động của nhóm, bao gồm: lập kế hoạch thực hiện, tìm kiếm tài liệu, phân chia công việc, tổ chức các buổi họp nhóm, xác định các chức năng, lập trình một số chức năng
<b>Mai Quốc Kiệt</b>	Nghe theo chỉ dẫn của nhóm trưởng, tìm hiểu các tài liệu, thực hiện công việc theo phân công và các công việc khác theo nhóm trưởng, lập trình giao diện và các chức năng còn lại

## CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ

### 1. Phần mềm và thư viện sử dụng:

Ngôn ngữ lập trình: Java

Phần mềm sử dụng: NetBean IDE

Thư viện sử dụng: Graphics2D, Swing

### 2. Các chức năng trong chương trình:

Xử lý ảnh bằng các bộ lọc: Gaussian, Edges, Sharpen, Emboss và một bộ lọc Custom cho phép người dùng tự nhập vào bộ lọc (dành cho người có kiến thức về xử lý ảnh)

Vẽ hình oval, hình chữ nhật, cắt, chèn chữ, cho phép người dùng chọn kích cỡ cây cọ để vẽ cũng như chọn màu của cây cọ

Cho phép người dùng load ảnh từ một vị trí trong máy để xử lý và lưu ảnh đã xử lý vào trong máy, cũng như các chức năng như undo (trở về trạng thái trước) và redo trong (đi tới trạng thái sau) như trong Microsoft Paint

### 3. Cách xử lý các chức năng:

Tạo giao diện:

Phác họa giao diện sao cho giao diện trực quan và dễ dàng cho người sử dụng

Giao diện sẽ bao gồm: một JPanel để thể hiện ảnh đang được xử lý, các Button để người dùng thực hiện các chức năng trên ảnh, một Jslider để lựa chọn kích cỡ cây cọ, cũng như các JTextField để cho người dùng nhập bộ lọc ảnh tùy chọn của họ và một Menu Bar để thực hiện các chức năng về File, Edit như save, load ảnh, undo, redo, ...

Xử lý các chức năng:

Chức năng dùng bộ lọc: áp dụng thư viện Graphics2D, chỉ cần ta đưa vào ma trận kernel, thư viện sẽ hỗ trợ ta xử lý. Nếu ta tự code thì thời gian thực thi sẽ lâu hơn do không tối ưu bằng

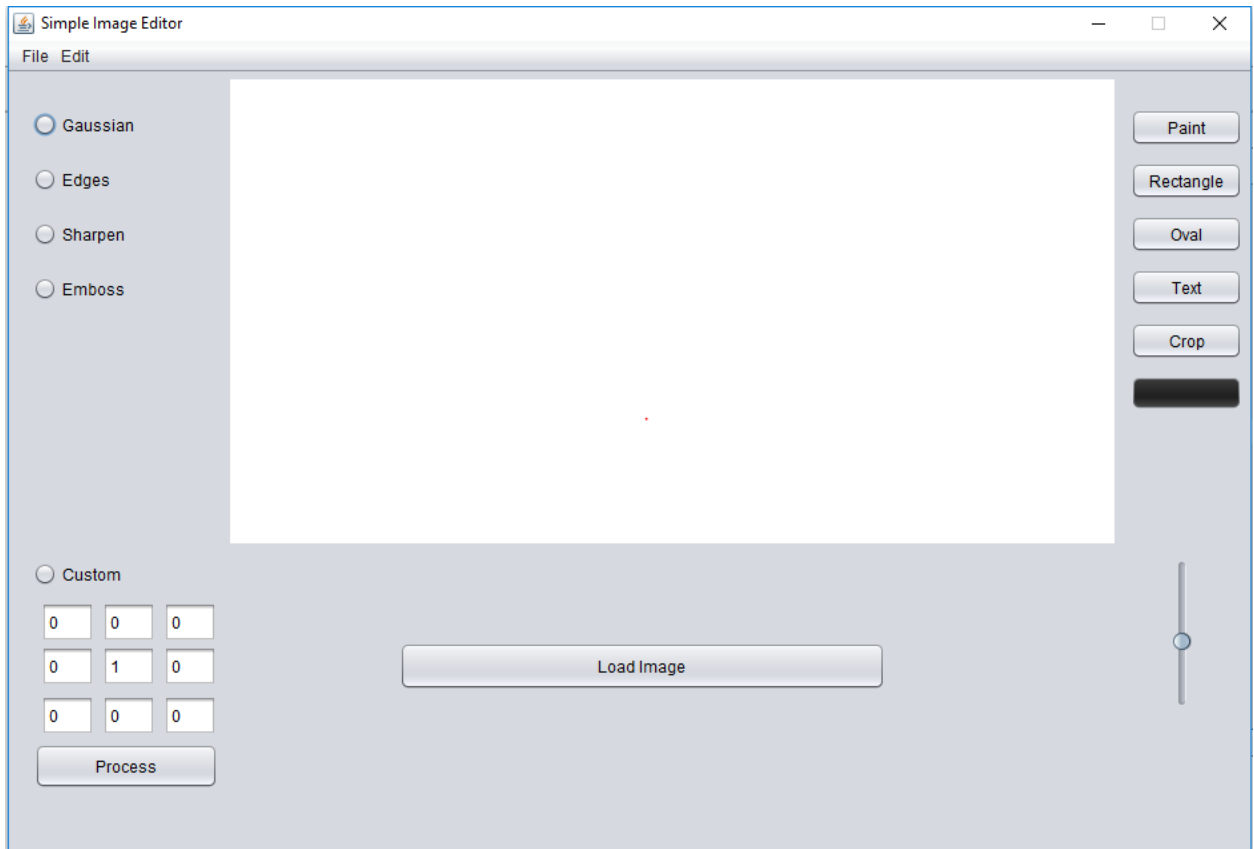
Chức năng vẽ đường thẳng, vẽ hình chữ nhật, vẽ hình oval, vẽ chữ: sử dụng các hàm draw như drawLine(), drawRect(), drawOval(), drawText()

Chức năng load ảnh và save ảnh: sử dụng JFileChooser

Chức năng Redo và Undo: sử dụng một ArrayList để lưu các trạng thái

# CHƯƠNG 4: THỰC HIỆN

## 1. Giới thiệu giao diện:



Giao diện được xây dựng bằng cách kéo thả các thành phần và đặt vào một JFrame

## 2. Các thành phần trong giao diện:

### 2.1. MenuBar:

MenuBar gồm 2 MenuItem: File, Edit

Trong Menu File có các chức năng:

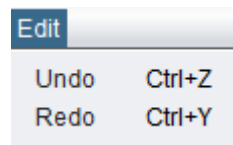
File	Edit
Open	Ctrl+O
Save	Ctrl+S
Exit	Alt+F4

Open (sử dụng bằng tổ hợp phím Ctrl + O): dùng để chọn và load ảnh để xử lý. Chức năng này được xử lý qua hàm load() bằng việc áp dụng JFileChooser hỗ trợ bởi thư viện Swing. Sau khi được load, ảnh sẽ được đưa lên một JPanel để thể hiện lên trên frame

Save (sử dụng bằng tổ hợp phím Ctrl + S): dùng để save ảnh đã xử lý vào ổ cứng máy tính. Chức năng này được xử lý bằng việc áp dụng JFileChooser hỗ trợ bởi thư viện Swing

Exit (sử dụng bằng tổ hợp phím Alt + F4): dùng để thoát ra khỏi chương trình. Chức năng này được xử lý bằng hàm dispose() để hủy bỏ frame

Trong Menu Edit có các chức năng:

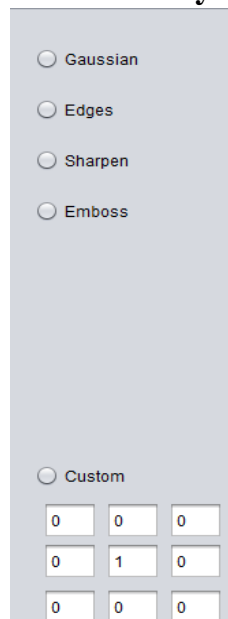


Undo (sử dụng bằng tổ hợp phím Ctrl + Z): Trở lại trạng thái xử lý trước của bức ảnh

Redo (sử dụng bằng tổ hợp phím Ctrl + Y): Trở lại trạng thái xử lý sau của bức ảnh (nếu người dùng có sử dụng Undo trước đó)

Để xử lý hai chức năng này, ta sử dụng một ArrayList để lưu trạng thái của bức ảnh qua các lần xử lý.

## 2.2. Phần để xử lý ảnh bằng bộ lọc:





Các thành phần trong xử lý ảnh bằng bộ lọc:

JRadioButton Gaussian: khi người dùng chọn vào sẽ áp dụng bộ lọc Gaussian Blur vào ảnh đang được xử lý, dùng để làm mờ ảnh. Bộ lọc Gaussian Blur có giá trị là:

$$\begin{bmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{bmatrix}$$

JRadioButton Edges: khi người dùng chọn vào sẽ áp dụng bộ lọc Edges vào ảnh đang được xử lý, dùng để tìm cạnh của ảnh. Bộ lọc Edges có giá trị là:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

JRadioButton Sharpen: khi người dùng chọn vào sẽ áp dụng bộ lọc Sharpen vào ảnh đang được xử lý, dùng để làm sắc ảnh. Bộ lọc Sharpen có giá trị là:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

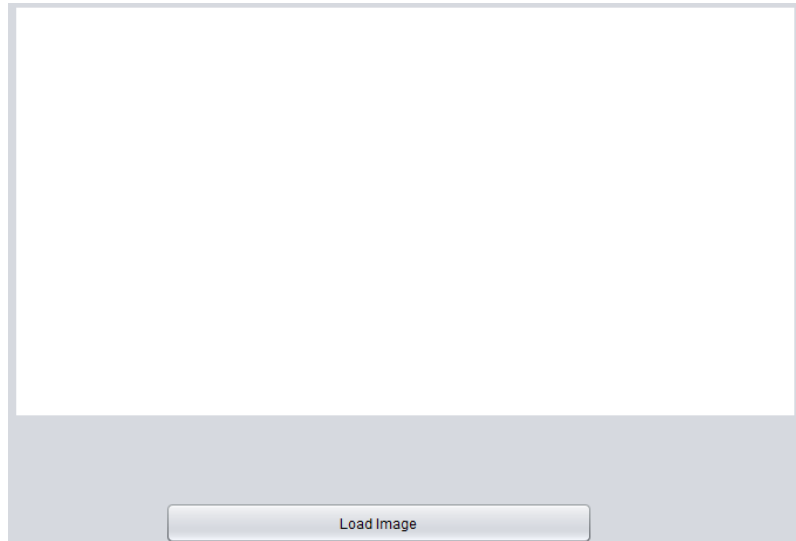
JRadioButton Emboss: khi người dùng chọn vào sẽ áp dụng bộ lọc Emboss vào ảnh đang được xử lý, dùng để làm nổi cạnh. Bộ lọc Emboss có giá trị là:

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

JRadioButton Custom: khi người dùng chọn vào sẽ áp dụng bộ lọc do người dùng nhập vào và áp dụng vào ảnh đang được xử lý khi bấm vào nút process.

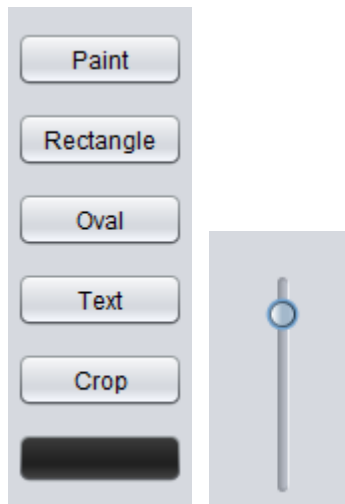
Sau khi chọn bộ lọc, ta nhân kernel với ảnh bằng cách tạo new ConvolveOp() để cho ra ảnh kết quả

### 2.3. Phần thể hiện ảnh và làm việc với ảnh:



Hình được chọn và load sẽ được thể hiện ở đây. Ngoài cách sử dụng cách Menu → File → Open để mở ảnh, ta cũng có thể sử dụng nút Load Image.

### 2.4. Phần thêm chi tiết vào ảnh:



JSslider ở bên phải cho phép người dùng chọn kích cỡ nét vẽ của cây cọ. Càng kéo lên cao thì nét vẽ của cây cọ càng cao

JButton ở dưới cùng dùng để chọn màu cho nét vẽ. Màu của button sẽ là màu mà bạn đang chọn

JButton Paint: dùng để vẽ nét lên ảnh. Khi chọn, ta sẽ vào chế độ vẽ lên ảnh. Sau đó, khi ta kéo chuột lên bức ảnh trên JPanel thì sẽ hiện ra nét vẽ với kích cỡ và màu đã lựa chọn. Chức năng này được xây dựng thông qua hàm drawLine(). Độ dài đường này được xác định bởi điểm đầu old và điểm cuối current của MouseDragged event

**JButton Rectangle:** dùng để tạo 1 hình chữ nhật lên ảnh. Khi chọn, ta sẽ vào chế độ tạo hình chữ nhật. Sau đó khi ta kéo chuột lên bức ảnh sẽ tạo ra một hình chữ nhật với kích cỡ và màu đã chọn. Chức năng này được xây dựng thông qua hàm `drawRect()`. Diện tích phần hình chữ nhật được tạo ra được xác định qua điểm đầu `old` của `MousePressed` event, cập nhật lại liên tục qua `MouseDragged` event và điểm cuối `current` của `MouseReleased` event

**JButton Oval:** dùng để tạo 1 oval lên ảnh. Khi chọn, ta sẽ vào chế độ tạo hình oval. Sau đó khi ta kéo chuột lên bức ảnh sẽ tạo ra một oval với kích cỡ và màu đã chọn. Diện tích phần hình oval tạo ra được xác định qua điểm đầu `old` của `MousePressed` event, cập nhật lại liên tục qua `MouseDragged` event và điểm cuối `current` của `MouseReleased` event

**JButton Text:** dùng để tạo 1 text lên ảnh. Khi chọn, ta sẽ vào chế độ thêm chữ vào ảnh. Sau đó khi ta kéo chuột lên bức ảnh sẽ hiện lên một khoảng `TextArea` để nhập chữ vào và sau khi bấm ra ngoài `TextArea` đó thì phần chữ sẽ được ghép vào hình. Chức năng này được xây dựng thông qua hàm `drawText()`. Diện tích phần `TextArea` tạo ra được xác định qua điểm đầu `old` của `MousePressed` event, cập nhật lại liên tục qua `MouseDragged` event và điểm cuối `current` của `MouseReleased` event

**JButton Crop:** dùng để cắt ra một phần ảnh từ ảnh lớn. Khi chọn, ta sẽ vào chế độ cắt ảnh. Sau khi kéo chuột lên bức ảnh và thả ra, ta sẽ cắt phần đó ra khỏi bức ảnh gốc. Diện tích phần được cắt ra được xác định qua điểm đầu `old` của `MousePressed` event, cập nhật lại liên tục qua `MouseDragged` event và điểm cuối `current` của `MouseReleased` event

## CHƯƠNG 5: KẾT LUẬN

### 1. Kết quả:

Nhóm đã hoàn thành về mặt giao diện, chức năng của chương trình theo những gì nhóm đã đề ra. Mặc dù vẫn có một số hạn chế, nhưng các thành viên đều rất cố gắng làm và hoàn thành trách nhiệm của mình

### 2. Một số mặt hạn chế:

Giao diện còn chưa đẹp mắt

Còn một số lỗi không mong muốn

### 3. Đánh giá thành viên:

Tiêu chuẩn đánh giá	Lý Bảo Khang	Mai Quốc Kiệt
Đóng góp về nội dung, chất lượng đồ án	100%	100%
Giải quyết vấn đề	100%	100%
Hợp tác tốt	100%	100%
Chuyên cần	100%	100%

### 4. Tài liệu tham khảo:

Chuỗi bài viết về thư viện Graphics2D trên Oracle:

<https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics2D.html>

Chuỗi bài viết về thư viện Swing trên Oracle:

<https://docs.oracle.com/javase/7/docs/api/javafx/swing/package-summary.html>

Cách giải quyết các vấn đề trên stackoverflow:

<https://stackoverflow.com/>