# BÁO CÁO CUỐI KÌ
# PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

Giáo viên hướng dẫn: ThS. Nguyễn Ngọc Long

Phan Quang Khánh - 1611146
Lý Phi Long - 1611146

Ngày 27 tháng 12 năm 2019

# GIỚI THIỆU CHUNG

## Đề bài

Thiết kế và cài đặt một ứng dụng GUI cho phép vẽ các đối tượng hình học 2D. Các đối tượng 2D có thể được vẽ đường biên và được tô. Người sử dụng vẽ hình bằng cách thao tác trên chuột. Người sử dụng có thể thay đổi thuộc tính của hình như màu sắc, kích thước, vị trí bằng chuột. Người sử dụng có thể tạo nên những đối tượng 2D mới bằng các phép toán hợp, hiệu, giao ; mỗi đối tượng được tạo mới như vậy được kể như một đối tượng riêng rẽ có đường biên riêng và có màu tô riêng.

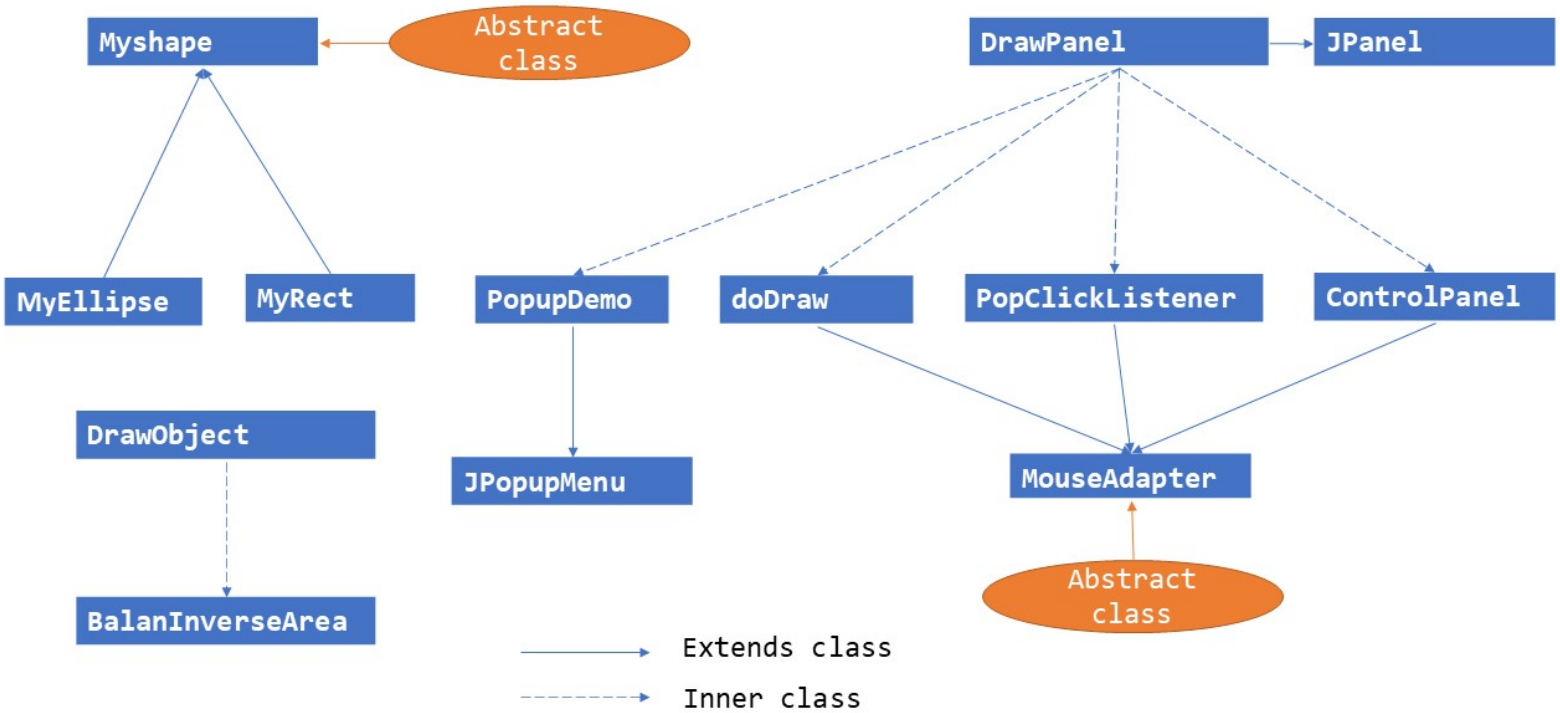
## Source code

`https://github.com/pqkhanh561/paint_complex_object.git`

# Các khó khăn khi gặp

- Nhóm mới tiếp cận và làm quen đến JAVA và JAVA.SWING.
- Tìm cách di chuyển đối tượng hình học sau khi thực hiện phép toán.
- Khó khăn xây dựng lớp đối tượng hình học cần vẽ.

# SƠ ĐỒ LỚP


Myshape
Abstract class
DrawPanel
JPanel
MyEllipse
MyRect
PopupDemo
doDraw
PopClickListener
ControlPanel
DrawObject
JPopupMenu
MouseAdapter
Abstract class
BalanInverseArea
Extends class
Inner class

# PHƯƠNG PHÁP TIẾP CẬN

## Mô hình đối tượng

- Ta sẽ tạo một vector để chứa tất cả các đối tượng được vẽ và các phép toán giao, hợp, phần bù.
- Mỗi hình đều chứa các vị trí điểm đầu và điểm cuối của chuột khi vẽ (bounding box).
- Khi ta giao, hợp hay phần bù chương trình sẽ sử dụng thuật toán balan ngược để tính toán (có thể làm một lúc được nhiều hình)

## Chức năng chính

- Chức năng vẽ : Sử dụng các lệnh vẽ hình chữ nhật, hình tròn trong thư viện `import java.awt.Graphics2D`
- Chức năng giao hình : sử dụng các lệnh giao hình trong thư viện `import java.awt.geom.Area`, và thuật toán balan ngược
- Chức năng thay đổi vị trí : tính khoảng cách giữa 2 điểm trước và sau khi di chuyển của chuột.

# PHƯƠNG PHÁP GIAO HÌNH

## Khó khăn và khắc phục

- Sử dụng phần thể hiện trên GUI dùng lớp trong java là Area. Lợi ích của thư vện area này là thỏa mãn được các phép toán mà đề bài đưa ra, tuy nhiên lớp này không dễ tùy biến được vị trí nên không thể sử dụng như một đối tượng có thể di chuyển được. Do đó phương pháp tiếp cận là kết hợp lớp Area để thể hiện được phép toán và xây dựng lớp DrawObject để quản lý các hình được vẽ.

```
public class DrawObject{
   private ArrayList<MyShape> arrShape = new ArrayList<MyShape>();
   private ArrayList<String>func = new ArrayList<String>();
   private DrawObject.BalanInverseArea area = null;
   private Color color = null;
```

Ở đây biến `arrShape` là một dãy lưu các đối tượng hình học cơ sở (hình ellipse, hình chữ nhật,...) mà đối tượng thực hiện các phép toán. Biến `func` để lưu trữ các phép toán dưới dạng biểu thức đại số. Biến `area` là vùng hình thể hiện trên GUI và được xử lý bằng thuật toán balan. Biến `color` là màu của đối tượng hình vẽ.

# PHƯƠNG PHÁP GIAO HÌNH

## Khó khăn và khắc phục

- Phép toán giữa các hình rất khó khăn vì phải lưu lại thứ tự thực hiện phép toán với các hình. Do đó nhóm thực hiện lưu giữ các phép toán dưới dạng một biểu thức đại số và dùng thuật toán balan ngược để chuyển biểu thức này thành dạng ký pháp tiền tố thuận lợi cho việc thực hiện các phép toán.

| Infix | Prefix | Postfix |
|---|---|---|
| $x + y$ | $+xy$ | $xy+$ |
| $x + y - z$ | $- + xyz$ | $xy + z-$ |
| $x + y * z$ | $+x * yz$ | $xyz * +$ |
| $x + (y - z)$ | $+x - yz$ | $xyz - +$ |

# RÚT GỌN CODE

```
subtractMenuItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        doMathForDrawObject(SUBTRACT);
    }
});

addMenuItem = new JMenuItem("Add");
clickItem.add(addMenuItem);
addMenuItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        doMathForDrawObject(ADD);
    }
});

intersectMenuItem = new JMenuItem("Intersect");
clickItem.add(intersectMenuItem);
intersectMenuItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        doMathForDrawObject(INTERSECT);
    }
});
```

# RÚT GỌN CODE

Thay vì thực hiện các dòng lệnh code giống nhau giữa các phép toán. Ta xây dựng một hàm để thực hiện các phép toán như trên để giảm thiểu số lượng code và tránh trùng lặp code.

```
public void doMathForDrawObject(String func){
    DrawObject tmp = selected.get(0);
    for (DrawObject ob: selected) {
        if (selected.indexOf(ob) != 0) {
            tmp = tmp.do_math(ob, func);
        }
    }
    ListObject.removeAll(selected);
    selected.clear();
    ListObject.add(tmp);
    dp.repaint();
}
```

# DEMO CHƯƠNG TRÌNH

## Ưu điểm

- Lợi dụng thuật toán balan ngược để xử lý các phép toán trên nhiều hình.
- Cho phép di chuyển, thay đổi màu sắc bất kỳ đổi tượng nào.
- Thực hiện khá tốt yêu cầu bài toán.
- Dễ mở rộng chương trình.

## Khuyết điểm

- Thiết kế giao diện chưa đẹp.
- Không thể resize được đối tượng.
- Chưa thể vẽ được một số hình phức tạp, như hình đa giác bất kỳ.

# DEMO CHƯƠNG TRÌNH

## Phát triển

- Cho phép người dùng có thể vẽ được nhiều hình hơn : Polygon, hình thoi,...
- Cho phép người dùng có thể resize hình được.
- Cho phép sắp xếp các vị tương đối giữa các object.
- Khi thực hiện phép toán mà tạo ra 2 vùng hình riêng biệt thì sẽ tạo ra 2 đối tượng độc lập với nhau.

# DEMO CHƯƠNG TRÌNH

## Hướng dẫn sử dụng

- Khi ta nhấn chuột phải thì sẽ hiện ra một hộp thoại thực thi chương trình
- Trong lệnh **Operation**, chứa các phép toán giữa hai hình học : phép giao(Intersect), phép hợp(Add), phép lấy phần bù(Subtract). Khi làm phép toán trên các hình xong, ta vẫn có thể thay đổi được màu và vị trí của hình sau khi được giao xong.
- Trong lệnh **Draw**, ta sẽ chọn các đối tượng để vẽ : Ellipse (hình ellipse), Rectangle (hình chữ nhật).
- Trong lệnh **Color**, ta có thể thay đổi màu cho đối tượng. Ở đây chương trình, lấy đại diện ba màu cơ bản (RED, GREEN, BLUE). Ban đầu khi ta vẽ, màu sẽ được khởi tạo ngẫu nhiên. Lưu ý, để thay đổi được màu của đối tượng, ta cần chọn vào đối tượng trước
- Để di chuyển đối tượng, ta nhấn giữ trái chuột và kéo.
- Để chọn một hình bất kỳ, ta double click vào hình đó. Khi ta bỏ chọn, ta double click ngược lại vào hình.

# CẢM ƠN THẦY VÀ CÁC BẠN ĐÃ LẮNG NGHE