

## Tuần 5. (1) Đa hình, instanceof, abstract

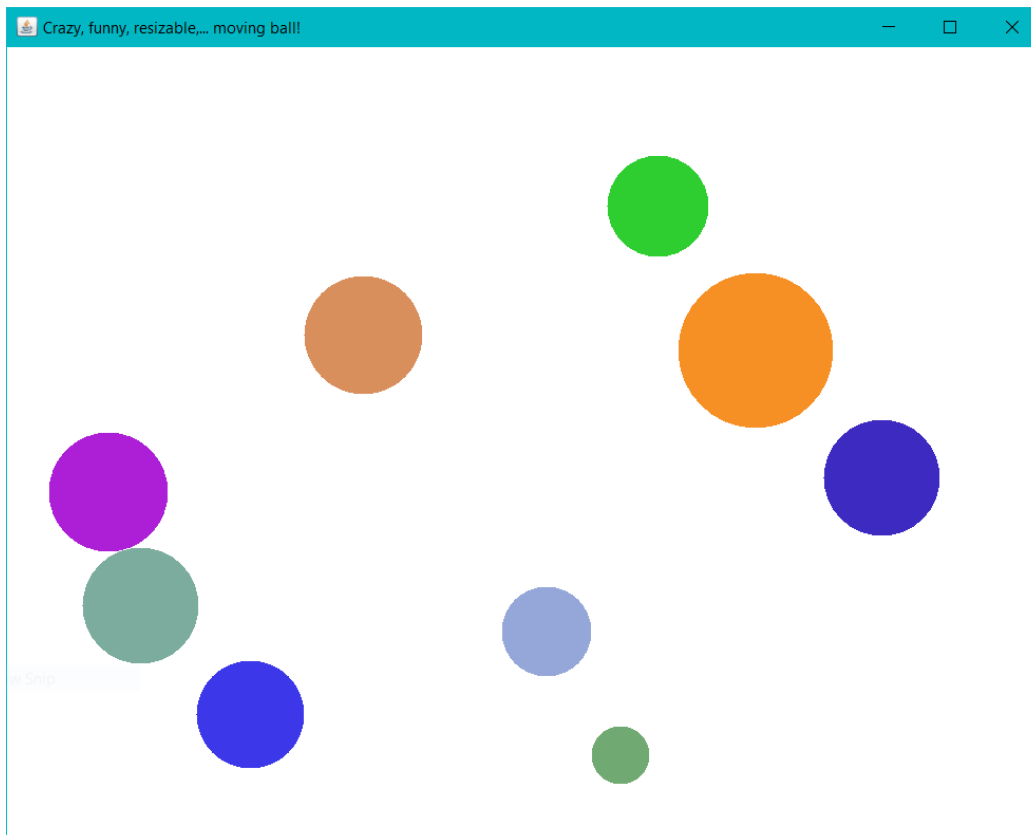
Giả sử bạn cần viết một ứng dụng đồ họa với những thông tin thiết kế ban đầu như sau

- Diagram: là lớp đại diện cho sơ đồ đang được vẽ
- Layer: một đối tượng thuộc lớp Diagram có một hoặc nhiều đối tượng thuộc lớp Layer
- Shape: là lớp đại diện cho các hình vẽ khác nhau (Rectangle, Square, Triangle, Circle).  
Mỗi đối tượng Layer chứa nhiều đối tượng lớp Shape
- Các hình vẽ có thuộc tính để xác định vị trí và kích thước
- Các hình vẽ có thể được tô màu và có thể di chuyển

Hãy:

- Định nghĩa các lớp trên (**Diagram**, **Layer**, **Shape**, **Rectangle**, **Square**, **Triangle**, **Circle**, và các lớp khác nếu cần thiết)
- Bổ sung phương thức cho lớp Layer để xóa tất cả các đối tượng thuộc lớp Triangle trong lớp
- Bổ sung phương thức cho lớp Diagram để xóa tất cả các đối tượng thuộc lớp Circle trong Diagram
- Viết phương thức main để kiểm thử các phương thức trên

Chương trình mẫu (chỉ có Circle, <https://youtu.be/RfG2hQN3zfM>):



## Tuần 6. (2) Đa hình

Tiếp tục từ bài buổi trước, hãy:

- Thêm thuộc tính visible cho Layer. Khi thuộc tính này là false, các hình thuộc đối tượng Layer đấy sẽ không được vẽ trên Diagram
- Viết phương thức xóa các hình trùng nhau trong một Layer (ví dụ: với Circle, 2 hình trùng nhau sẽ có tọa độ tâm và độ lớn bán kính như nhau)
- Viết phương thức để chuyển từng loại hình vẽ vào từng đối tượng Layer. Sau khi chạy phương thức này, tất cả các Square của Diagram sẽ được chuyển vào 1 Layer, các Circle được chuyển vào 1 Layer khác,... Nếu số loại hình nhiều hơn số Layer hiện có thì tạo thêm.
- Ngoài những hình đã có (Rectangle, Square, Triangle, Circle), bổ sung thêm hình lục giác đều **Hexagon**

Yêu cầu:

- Tất cả mọi chương trình phải có đủ comment cho từng class, từng hàm
- Các thuộc tính cần có đủ setter, getter tương ứng