

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÀI BÁO CÁO

Thực hành phương pháp lập trình
Hướng Đối Tượng

#Assignment 4
Đồ họa với SVG, GDI+ và CImg

Sinh viên thực hiện:

Đỗ Thanh Lam

Lớp:

15CTTN

Mã số sinh viên:

1512275

1. Cách tổ chức và sắp xếp các class:

Với cách sắp xếp này ta hoàn toàn có thể thêm vào một hình mới, hay một thuộc tính khác cho tất cả các hình một cách dễ dàng. Ví dụ như trong đồ án 3 em sẽ thêm phần ghi file SVG và đọc tập tin SVG và thuộc tính reconstruct đối tượng từ một chuỗi (char*) được đọc.

✧ Bao gồm 2 class cơ bản

- **Coor:** Tọa độ

Gồm tọa độ x, y (float) và thuộc tính getX, getY.

- **ColorRGB:** Màu

Gồm 3 màu Red, Green, Blue và thuộc tính lấy mảng động 3 phần tử ứng với giá trị ba màu.

✧ Ta thiết kế lớp **Img** là lớp nền duy nhất, các chức năng vẽ đều bắt buộc thông qua nó. User không thể tạo ra hơn 1 đối tượng lớp Img. Trong lớp Img thành phần chính là đối tượng Graphics của thư viện GDI+.

✧ Lớp **Shape** là lớp thuần ảo đại diện cho các *hình vẽ*, chứa phương thức ảo draw(), readSVG() và writeSVG() với tầm vực private và làm bạn với lớp Img để lớp Img có thể thông qua phương thức draw này.

✧ Lớp **Properties** chứa các thuộc tính chung của mỗi hình: có hai loại constructor, 1 dành cho hình không có tô nền (line), một dành cho có tô nền (circle,...). Thành phần quan trọng là phương thức getCanvas() – trả về đối tượng Graphics nên phải làm bạn với lớp Img. Thông qua phương thức getCanvas để các lớp hình sau đó kế thừa lại thực hiện chức năng draw() và writeSVG().

Gồm có m_fill (màu cần tô: colorRGB), m_stroke (nét

✧ Lớp **Transform** là lớp thuần ảo đại diện thuộc tính Rotate và Translate chứa phương thức ảo actionTrans(Graphics *canvas). Ngoài thuộc tính Rotate và Translate, user có thể viết thêm các thuộc tính khác và định nghĩa lại phương thức actionTrans.

- ⤴ Ngoài ra lớp **Circle** và **Ellipse** cũng tương tự nhau nên cho lớp Circle kế thừa Ellipse. Lưu ý các phương thức vẽ Ellipse trong GDI+ cần đường kính nên ta phải nhân 2 tham số bán kính.
- ⤴ Các lớp **Line**, **Rectangle**, **Polyline** là ba lớp hình học cơ bản nhất.
 Line gồm 2 điểm đầu và cuối.
 Rectangle gồm 1 điểm và độ dài hai cạnh hình chữ nhật (hình chữ nhật có cạnh song song với hai trục tọa độ).
 Polyline gồm tập hợp các điểm.
- ⤴ Ta cho lớp **PolyGon** kế thừa gần như hoàn toàn lớp **PolyLine** vì 2 lớp này có chung thuộc tính như nhau, tuy nhiên phương thức `draw()` và `writeSVG()` vẫn cần phải định nghĩa lại.
- ⤴ Qua cách thiết kế trên, user hoàn toàn có thể viết thêm các lớp hình khác, có thể dung lại Properties hoặc tự định nghĩa lại các thuộc tính mới.

2. Cải tiến ở đồ án 3:

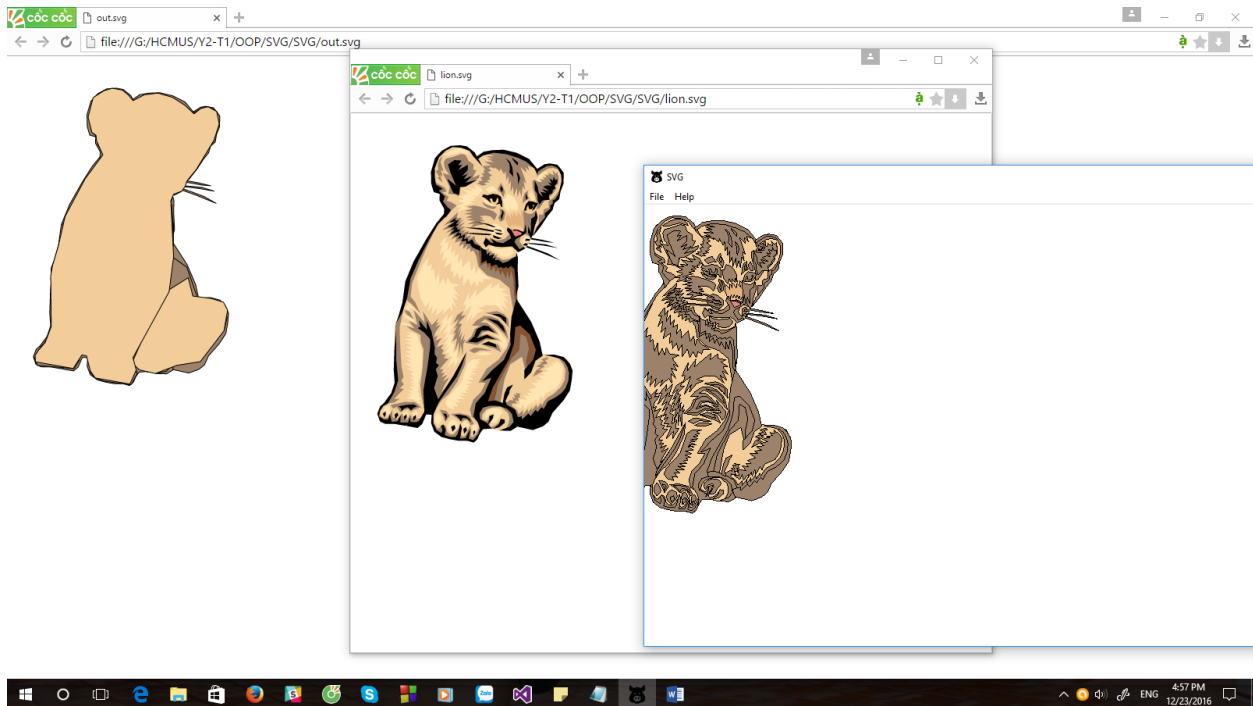
- Tạo thêm lớp **ImportSVG** với constructor là địa chỉ tập tin SVG. Cũng làm bạn với lớp `Img` để có thể vẽ thông qua lớp này.
- Lớp `Shape` thêm phương thức ảo `writeSVG(FILE*f, ostream& os)` – viết ra tập tin, và `read(FILE*f, rapidxml::xml_node<> *node, istream& is)` để đọc tập tin qua RapidXML. RapidXML có thể đọc theo mô hình cây nhưng ở đây ta chỉ tập trung vào tập tin SVG không có các thẻ lồng nhau tức là chỉ có cấp 1. Các lớp hình sau đó phải định nghĩa lại 2 phương thức này.
- Sự khác nhau giữa tập tin svg đầu vào và đầu ra:
 - ✓ Hình chuẩn hơn, nhưng hình trong file đọc vào nét mượt hơn hình xuất ra vì file xml xuất đủ cả các thuộc tính mặc định không xuất hiện trong tập tin đầu vào: tọa độ `Coor(0, 0)`, độ trong suốt nền và đường viền là 1, ...
 - ✓ Các phần tử hình học được sắp xếp lại chuẩn hơn sơ với đầu vào, cách dòng, xuống dòng...

- Nhược điểm của mô hình thiết kế này:
 - ✓ Trong phần lấy các thuộc tính trong class Properties cần phải dùng if hoặc switch case nhiều lần với mỗi shape. Như vậy đã không đáp ứng được mục tiêu linh hoạt của phương pháp lập trình hướng đối tượng.
 - ✓ Cách thức xử lý nhập thuộc tính Transform hiện chỉ có thể nhập từ tập tin SVG chỉ có 3 loại Rotate, Scale và Translate, do thứ tự xuất hiện của các thuộc tính này ảnh hưởng kết quả hình vẽ.

3. Cải tiến đề án 4

- Tạo Class Abstract Attribute gồm Value và ColorValue.
- Tạo Class Fill, Stroke, Fill_Opacity, Stroke_Opacity, Stroke_width, TranSet có abstract class là Attribute.
- Chỉnh sửa tại Properties. Tạo một Set <Attribute>.

4. Thách thức mở rộng



Em đã cố gắng hết sức nhưng hình như đã sai cách tổ chức.

Do từ khi tổ chức nếu fill hay stroke không có giá trị, em gán bằng RGBColor(0,0,0) còn nếu fill-opacity , stroke-opacity, stroke-width không có giá trị em sẽ gán bằng 0. Nên khi in ra màu nó fill đầy. Nhưng do cách tổ chức từ đầu bài nên không đỏi kịp nữa rồi.

Reference:

<http://rapidxml.sourceforge.net/manual.html> (Tài liệu Rapid XML)

File streamfile.pdf (Tài liệu đọc file stream)