

객체지향 프로그래밍 5차 과제

문제

10. 간단한 그래픽 편집기를 콘솔 바탕으로 만들어보자. 그래픽 편집기의 기능은 삽입, "삭제", "모두보기", "종료"의 4가지이고, 실행 과정은 다음과 같다.

소스 수행 결과 화면 캡처

```
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 1
선:1, 원:2, 사각형:3 >> 1
도형이 추가되었습니다.
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 1
선:1, 원:2, 사각형:3 >> 2
도형이 추가되었습니다.
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 1
선:1, 원:2, 사각형:3 >> 3
도형이 추가되었습니다.
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 3
0: Line
1: Circle
2: Rectangle
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 2
삭제하고자 하는 도형의 인덱스 >> 1
도형이 삭제되었습니다.
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 3
0: Line
1: Rectangle
삽입:1, 삭제:2, 모두보기:3, 종료:4 >>
```

문제 정의

그래픽 편집기를 콘솔 기반으로 구현하는 프로그램 입니다.

- 선(Line), 원(Circle), 사각형(Rectangle) 도형 중 하나를 삽입합니다.

- 사용자가 선택한 도형을 삭제
- 현재 삽입된 모든 도형을 출력

문제 해결 방법

1. 설계

- 공통적인 도형의 특징을 정의한 `Shape` 클래스를 작성.
- `Shape` 클래스를 상속받아 `Line`, `Circle`, `Rect` 등의 세부 도형 클래스를 구현

2. 그래픽 편집기

- `GraphicEditor` 클래스를 설계하여 도형을 관리(삽입, 삭제, 보기)하는 역할 수행
- `GraphicEditor` 는 `Shape` 객체를 동적으로 관리하기 위해 `std::vector<Shape*>` 를 사용.

3. Main

알고리즘 설명

- 도형 삽입:
 - `insertShape(int shapeType)` 메서드를 호출
 - 사용자 입력에 따라 `Shape` 포인터 배열에 `Line`, `Circle`, `Rect` 객체를 동적으로 생성하고 추가
- 도형 삭제:
 - `deleteShape(int index)` 메서드를 호출
 - 사용자로부터 삭제할 도형의 인덱스를 입력받아, 해당 도형을 삭제하고 메모리를 해제
- 모두 보기:
 - `showAllShapes()` 메서드를 호출하여, 벡터에 저장된 모든 도형 객체의 `draw()` 메서드를 호출
- 메인 루프:
 - 사용자가 종료를 선택하기 전까지 반복적으로 메뉴를 출력하고 동작을 수행