

- 문제 정의 : 문제를 명확히 이해하고 정의함.

해당 문제는 `vector<Shape*> v`를 이용해서 간단한 그래픽 편집기를 제작하는 것이다.

우선 구성으로는 Shape와 파생클래스 Circle, Line, Rect 클래스가 있으며,
해당 도형 객체를 v에 삽입하고 관리하는 것이다.

- 문제 해결 방법 : 문제를 해결하기 위한 아이디어들

우선 그전에 배웠던 개념인 가상함수 virtual 키워드를 사용하였기 때문에, 파생 클래스에 구현하는 아이디어가 필요하였고,

이 문제의 핵심은 도형의 삽입과 삭제인데,

`vector<Shape*> v`를 선언하고, `v.push_back(new 도형())`을 통해서 도형 객체를 삽입하였다. 삭제는 `new`를 사용하였기 때문에 `delete`라는 연산자와 `vector`에서 삭제하기 위한 `erase()`를 사용하였다.

- 아이디어 평가 : 문제 해결을 위해 제시한 아이디어들 수행 평가와 결과

아이디어는 `vector<Shape*> v`로 선언하였기 때문에,
도형 객체를 삽입하기 위해서 `push_back()` 멤버 함수를 사용하였다.

삽입할 때, '`new 도형()`'의 형태로 할당해주었고,
삭제할 때, `delete`를 사용해서 `new`를 통해 할당받았던 공간을 반환해주었다.
또한, `erase`를 통해서 `vector`의 해당 값을 삭제해주었다.

평가를 해보자면, 처음 수행하였을 때, `delete`만 하면 값이 삭제될 줄 알았는데,
오류가 나서 찾아보니 `erase`라는 것을 사용해야겠다고 생각해냈다.

- 문제를 해결한 키 아이디어 또는 알고리즘 설명

문제를 해결한 키 아이디어는 문제에서 주어진 virtual 상속 개념을 제외하면, vector를 통한 push_back과 erase, 그리고 new연산자와 delete연산자를 같이 활용했던 점이 키 아이디어이다.

Shape 클래스를 포인터 객체로 받고,
new 연산자를 통해 Line, Circle, Rect 객체를 할당받았다.
delete와 erase를 통해서 삭제를 하였다.

erase(v.begin() + index)으로 하였는데,
v.begin()이 벡터의 첫 번째 요소를 가리키는 iterator를 반환하고,
거기에 index를 더해서 입력받은 index 위치의 요소를 가리킬 수 있었다.