

객체지향 프로그래밍 소스 구현 설명

202104361

ICT공학부

천승환

1. 문제 정의 : 그래픽 편집기의 기능인 "삽입", "삭제", "모두보기", "종료"를 출력 부분에 나타내고 가상 함수와 추상 클래스를 이해하는 과정이다.

2. 문제 해결 방법 :

- Shape 클래스는 추상 클래스이다. Line, Circle, Rect 클래스는 Shape 클래스를 상속 받고, 각 draw()에서 도형의 이름을 출력하게된다. UI 클래스에서 main_menu()를 통해 그래픽 편집기의 기능을 사용할 수 있고, choose_shape()를 통해 추가할 도형을 선택할 수 있으며, delete_shape()를 통해 삭제할 도형을 입력받을 수 있다. 결론적으로 GraphicEditor 클래스에서 멤버 변수인 pStart와 pLast 포인터를 NULL로 초기화한 상태에서 시작한다. 입력한 숫자에 따라 알맞은 도형을 리스트에 추가하고 p가 이를 가리키도록 한다. 리스트가 비어 있지 않을 때는 pLast의 next 포인터를 새 도형 객체로 설정하고, pLast를 새로 추가한 도형으로 갱신한다. 삭제할 도형도 삭제 대상 도형(y)과 그 이전 도형(x)을 지정하여 삭제시킨다. 삭제할 대상이 pStart인 경우 pStart를 다음 도형으로 갱신하고 삭제 대상 도형(y)를 delete한다. 삭제할 대상이 중간 또는 마지막인 경우 이전 도형(x)의 next포인터를 삭제 대상의 다음 도형으로 설정하고 삭제 대상 도형(y)를 delete한다. show()는 입력했던 도형들을 출력하여 보여주게 되고, play()는 UI를 호출하여 실행되고 4(종료)를 누르게 되면 프로그램이 종료된다.

3. 아이디어 평가 :

- Shape 클래스를 순수 가상 함수로 정의함으로써 Line, Circle, Rect가 자기 자신만의 draw()를 제공할 수 있게 된다.
- Shape의 next 포인터를 사용하여 도형들이 동적으로 연결되기 때문에 다른 도형을 추가하고자 할 때 Shape를 상속받아 새로운 draw()를 제공해주면 된다.
- UI 클래스로 입력할 공간을 따로 만듦으로써 코드를 볼 때 한 눈에 알아볼 수 있다.

4. 문제를 해결한 키 아이디어 또는 알고리즘 설명 : 도형 삽입 알고리즘은 새 도형을 리스트 끝에 계속해서 추가하기 위함이다. 도형을 입력하고 입력값에 따라 생성하게되는데 리스트가 비어있으면 pStart와 pLast를 새 객체로 설정하고, 비어있지 않다면, pLast의 next포인터를 새 객체로 설정하고 pLast를 새 객체로 갱신한다. 도형 삭제 알고리즘은 지정된 위치의 도형을 삭제하기 위함이다. 삭제 대상이 pStart

인 경우 pStart를 그 다음 도형으로 갱신한다. 그 외에는 이전 도형의 next 포인터를 삭제할 대상의 다음 도형으로 갱신한다. 이는 결국 동적으로 도형을 관리 할 수 있다는 뜻이고, 메모리를 효율적으로 사용할 수 있다는 것이다.