

객체지향프로그래밍 과제5

202284006 김동휘

문제 정의

그래픽 편집기를 콘솔 바탕으로 구현한다. 사용자가 다양한 도형의 형태(Line, Circle, Rectangle)를 삽입하고 삭제하고 현재 삽입된 도형들을 모두 확인할 수 있도록 해야 한다. 주요 요구 사항은 다음과 같다.

1. 도형 삽입 기능
 - A. 사용자가 원하는 도형을 선택하여 삽입한다.
 - B. 삽입된 도형은 연결 리스트의 마지막에 추가한다.
2. 도형 삭제 기능
 - A. 사용자가 입력한 인덱스의 도형을 삭제한다.
 - B. 삭제 후 연결 리스트를 재연결하여 연결 리스트 구조 유지한다.
3. 도형 전체 보기
 - A. 삽입된 도형을 인덱스 번호와 함께 순차적으로 출력한다.
4. 프로그램 종료
 - A. 사용자가 종료 명령을 내리면 프로그램 종료한다.

문제 해결 방법:

1. 추상 클래스 사용
 - A. Shape 클래스를 추상 클래스로 설계하여 상속받는 도형의 기본 형태를 만든다.
 - B. Shape 클래스의 draw() 메서드를 순수 가상 함수로 선언하여 각 도형이 이를 상속받아 재정의 하도록 한다.
2. 연결 리스트
 - A. Shape 클래스의 멤버로 자기 참조 포인터 next를 두어 연결 리스트 구조를 구현한다.
 - B. 새로 삽입되는 도형은 add() 메서드를 통해 리스트의 마지막에 추가된다.
3. UI 클래스
 - A. UI 클래스를 통해 입력을 처리한다.

B. 메뉴와 도형 선택, 삭제 인덱스 번호 선택 등의 기능을 제공한다.

4. GraphicEditor 클래스

A. 연결 리스트를 관리하는 클래스이다.

B. insertItem(int type): 선택된 도형을 리스트에 삽입한다.

C. deleteItem(int index): 특정 인덱스의 도형을 삭제한다.

D. show(): 리스트에 포함된 도형들을 순차적으로 출력한다.

E. run(): 전체 프로그램 실행을 관리하는 반복문이다.

아이디어 평가:

1. 추상 클래스와 상속을 통해 새로운 도형 클래스를 쉽게 추가가 가능하다.
2. 연결 리스트 구조를 통해 도형 삽입 및 삭제 시 효율적으로 관리한다.
3. 입력과 도형 관리 프로그램이 분리되어 코드 가독성과 유지 보수에 효과적이다.

문제 해결의 핵심 아이디어:

1. Shape 클래스의 draw 메서드를 순수 가상 함수로 선언하여 도형별로 구체적인 구현을 제공한다.
2. paint 메서드 호출 시 실제 객체의 draw 메서드가 실행된다.
3. Shape* add(Shape* p) 메서드를 통해 새로운 도형을 리스트에 추가한다.
4. 삭제 시 특정 인덱스를 찾아 리스트를 재연결하여 메모리 누수를 방지한다.
5. UI 클래스는 입력 GraphicEditor 클래스는 입력에 따른 작업을 수행하여 가독성과 독립적인 클래스를 구현한다.