

문제 정의: 다양한 도형을 추가, 삭제, 조회할 수 있는 프로그램을 C++로 구현하는 것이다.
객체 지향적 설계를 기반으로 유연성과 재사용성을 높이는 것을 목표로 한다.

문제 해결 방법

도형의 동적관리

사용자가 추가한 도형을 동적으로 생성하고 관리한다.

연결리스트를 이용하여 메모리를 효율적으로 활용한다.

동적으로 생성된 객체를 안전하게 삭제.

유효성 검사

사용자가 입력한 값이 유효한지 확인한다.

잘못된 입력을 처리하고 사용자에게 안내 메시지를 제공한다.

객체 재향 설계

도형 추가 및 삭제 같은 기능이 클래스 내에 캡슐화되도록 설계한다.

유지보수 및 확장이 용이하도록 구조화 한다.

아이디어

Shape 추상 클래스

모든 도형의 공통적인 특성을 정의한 클래스이다.

도형을 그리는 기능은 가상 함수로 선언하여 상속받은 클래스가 구체적으로 구현하도록 한다.

다음 노드를 가리키는 포인터 Shape* next를 포함하여 연결 리스트의 역할을 수행한다.

Circle, Rect, Line 파생 클래스

Shape 클래스를 상속받아 구체적인 도형을 구현한다.

UI 정적 클래스

사용자로부터 입력을 받고, 올바른 값인지 검증한다.

도형 삽입, 삭제와 같은 기능 선택 및 세부 입력을 담당한다.

연결 리스트를 활용한 도형 관리

Shape* next 포인터를 이용해 도형들을 동적으로 연결한다.

리스트의 시작점 pStart와 마지막 도형을 가리키는 포인터 pLast를 사용해 삽입, 삭제를 구현한다.