

문제 정의: 그래픽 에디터를 통해 도형을 동적으로 관리하고, 객체 지향 원칙을 활용하여 유지보수와 확장이 용이한 프로그램을 구현

프로그램은 다음의 계층 구조를 가지며, 추상 클래스와 파생 클래스를 이용한다.

Shape: 도형의 기본 클래스

Circle, Line, Rectangle: Shape를 상속받은 구체적인 도형 클래스

주요 기능으로는 도형 삽입과 삭제, 모두보기, 그리고 프로그램 종료가 있다.

문제 해결 방법: 추상 클래스 Shape를 활용

파생 클래스를 구현하여 각 도형의 특성을 나타낸다.

벡터를 활용한 객체관리: 도형 객체들을 `std::vector`에 저장하여 삽입 및 삭제를 효율적으로 수행할 수 있음. STL 벡터를 이용하여 동적 크기 조정 및 메모리 관리를 자동화

사용된 아이디어들

추상 클래스와 다형성

다형성을 활용하여 Shape* 포인터를 통해 파생클래스의 동작을 실행

상속

동적 메모리 관리

도형 객체를 동적으로 생성하고 삭제를 통해 메모리를 관리

STL 벡터를 활용한 도형 관리

벡터를 통해 도형 객체의 포인터를 저장하고 삽입, 삭제, 조회등 기본 기능을 구현

UI 클래스

프로그램의 사용자 인터페이스를 관리하며 입력과 출력을 분리함

캡슐화

캡슐화를 유지하며 데이터를 안전하게 관리