

객체지향프로그래밍 12주차 과제 소스구현 설명

ICT융합학부 소프트웨어학과
202284002 고범주

1.문제 정의:

이 프로그램은 그래픽 에디터 프로그램을 구현하는 것이 목표입니다. 사용자가 도형을 삽입, 삭제하거나, 저장된 도형을 모두 확인할 수 있도록 하며, 잘못된 입력에 대한 예외 처리와 동적 메모리를 안전하게 관리하는 것이 중요합니다.

2.문제 해결 방법:

(1) 추상 클래스 사용

Shape를 추상 클래스로 정의하여 draw(모든 도형의 공통 인터페이스)를 제공합니다. 원,사각형,선 인 도형 클래스는 Shape를 상속받아 draw메서드를 구현하게됩니다.

(2) STL Vector 활용

삽입된 도형들을 관리하기 위해 `std::vector<Shape*>`를 사용합니다. 동적 크기 조절 및 반복자를 이용한 요소 접근/삭제를 지원합니다.

(3)사용자 입력 기반 UI 설계

UI 클래스를 통해 사용자 입력을 받고, 입력값에 따라 프로그램의 흐름을 제어합니다. 메뉴 및 도형 선택, 삭제 인덱스 등을 입력받는 기능을 제공합니다.

(4)동적 메모리 관리

각 도형 객체는 new로 생성되어 벡터에 저장됩니다. 삭제 시 delete를 호출하여 메모리를 안전하게 해제합니다. 종료 시 모든 남은 도형 메모리도 해제합니다.

3.아이디어 평가

(1)추상 클래스 사용

도형별로 공통 인터페이스를 강제하며 확장이 가능하나, 클래스 설계에 대한 이해가 요구됩니다.

(2) STL Vector 활용

동적 크기 조절과 각 요소들의 삽입,삭제가 간편해지나, 벡터의 크기가 커질수록 삭제작업이 비효율적일 수 있습니다.

(3)사용자 입력 기반 UI 설계

UI클래스를 이용하여 사용자 친화적이며 프로그램의 구조를 단순화 시켜줍니다. 사용자의 잘못된 입력에 따른 예외처리를 신경 써 줘야합니다.

4.문제를 해결한 키 아이디어 또는 알고리즘 설명

(1)추상 클래스 기반 다형성

Shape 클래스를 추상 클래스로 정의하여, paint() 메서드가 draw()를 호출하게 설계합니다. 각 도형은 draw() 메서드에서 자신의 고유한 출력(선, 원, 사각형)을 정의합니다.

(2)벡터와 반복자 사용

벡터를 통해 도형 객체를 동적으로 관리하며 삽입과 삭제를 효율적으로 구현합니다. iterator를 사용하여 벡터의 특정 위치에서 객체를 삭제하고 메모리를 해제합니다.

(3)UI 클래스

사용자와의 상호작용을 캡슐화하여 프로그램 흐름을 간결화 시킵니다. 메뉴 선택, 도형 삽입, 삭제, 보기 등의 입력을 단순화하여 코드 재사용성을 증가 시킵니다.