

-- 문제 정의

이 문제는 선, 원, 사각형을 동적으로 관리하고 이를 효율적으로 삽입, 삭제, 출력하는 그래픽 편집기를 만들어야 한다.

-- 문제 해결 방법

도형의 공통 인터페이스로 추상클래스(shape를 정의하였고, 순수 가상함수로써 draw를 virtual로 선언하였다.

선, 원, 사각형은 shape를 상속받아 draw()메서드를 구체적으로 구현하였다.

이를 통해 다양한 도형을 shape 타입 포인터로 관리하고, 다형성을 구현하였다.

GraphicEditor 클래스가 Shape 타입의 포인터 배열로 도형 객체를 관리한다.

동적 배열로 도형을 저장하고, 배열의 크기가 초과되면 자동으로 확장한다.

도형은 동적으로 생성, 삭제할 땐 메모리를 해제하여 메모리 누수를 방지한다.

메인함수에서는 문제의 실행 예시에 맞도록

“그래픽 에디터입니다.”를 먼저 출력하고

“삽입:1, 삭제:2, 모두보기:3, 종료:4 >> ”를 출력 한 뒤 삽입, 삭제, 출력, 종료 기능이 실행 되도록 구현하였다.

-- 아이디어 평가

새로운 도형 클래스를 추가하더라도, GraphicEditor를 수정하지 않고 Shape를 상속받아 새로운 draw() 메서드만 구현하면 된다.

배열 크기를 동적으로 확장하여 초기 크기 제한을 극복하였다.

소멸자를 통해 모든 동적 메모리를 해제하도록 하였다.

추상 클래스와 가상 함수를 통해 도형의 실제 타입에 관계없이 공통된 방식으로 도형을 관리할 수 있다.

-- 문제를 해결한 키 아이디어 또는 알고리즘 설명

추상 클래스인 Shape와 가상함수인 draw()를 이용하여 모든 도형 클래스가 자신만의 출력 방식을 정의하도록 하였다.

GraphicEditor는 Shape* 배열로 모든 도형을 관리하며, 각 도형의 타입에 관계없이 다형성을 통해 draw()를 호출하도록 하였다.

동적 배열을 통해 배열 용량을 초과하면 배열 크기를 늘리고 기존의 데이터를 새로운 배열로 복사하도록 하였다.

Shape 포인터 배열의 각 요소에 대해 draw() 호출 시, 실제 객체의 타입에 따라 적절한 출력하도록 하였다.