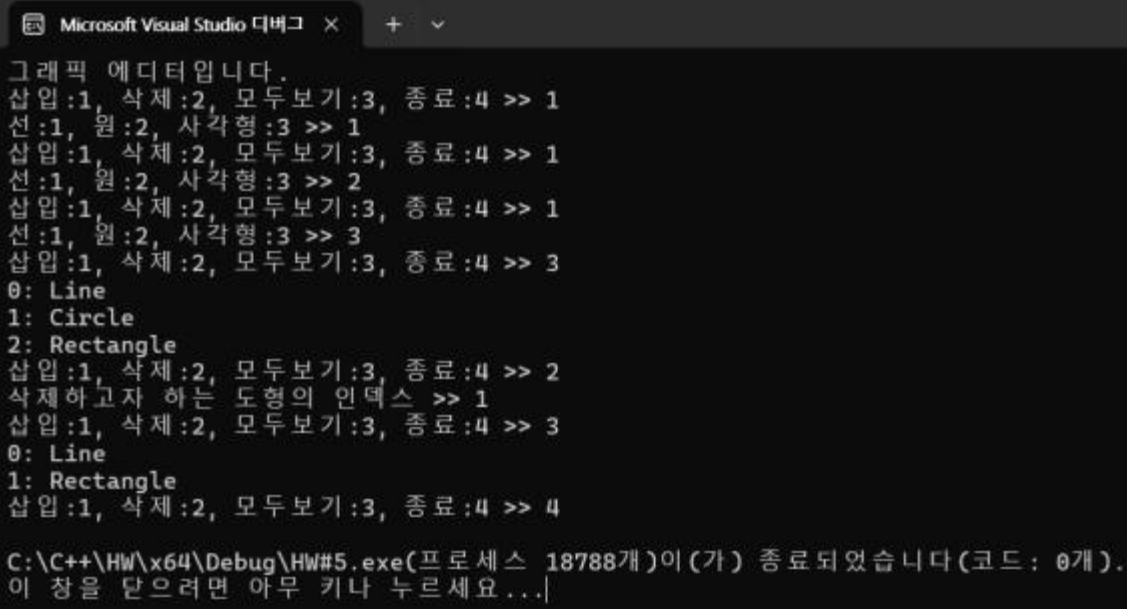


◆ 소스 수행 결과 화면



```
Microsoft Visual Studio 디버그 콘솔
그래픽 에디터입니다.
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 1
선:1, 원:2, 사각형:3 >> 1
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 1
선:1, 원:2, 사각형:3 >> 2
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 1
선:1, 원:2, 사각형:3 >> 3
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 3
0: Line
1: Circle
2: Rectangle
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 2
삭제하고자 하는 도형의 인덱스 >> 1
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 3
0: Line
1: Rectangle
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 4
C:\C++\HW\x64\Debug\HW#5.exe(프로세스 18788개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

◆ 소스 구현 설명

1. 문제 정의

간단한 그래픽 편집기를 구현하는 것이다. 이 편집기는 사용자가 기본 도형인 선(Line), 원(Circle), 직사각형(Rect) 등을 그릴 수 있도록 하며, 도형을 추가, 삭제, 수정, 표시하는 기능을 제공한다. 사용자는 명령어를 통해 도형을 조작할 수 있다.

2. 문제 해결 방법

- 기본 클래스 설계

Shape 클래스는 모든 도형의 기본 클래스를 정의한다. 이 클래스는 도형의 공통 속성과 메서드를 정의하고, draw()라는 순수 가상 함수를 포함한다. 이를 통해 각 도형 클래스(Circle, Line, Rect)가 자신의 방식으로 draw() 메서드를 구현할 수 있다.

- 구체적인 도형 클래스 구현

Circle, Line, Rect 클래스는 각각의 도형을 나타내는 클래스를 구현한다. 각 클래스는 Shape 클래스를 상속받아 자신의 고유한 속성과 메서드를 구현한다. 이러한 구조는 코드의 재사용성을 높인다.

- 도형 관리 클래스 구현

GraphicEditor 클래스는 도형을 관리하는 역할을 수행한다. 연결리스트를 사용하여 도형을 동적으로 추가하고 삭제할 수 있도록 해야 한다. 이 클래스는 도형 리스트의 시작(pStart)과 끝(pLast)을 가리키는 포인터를 사용하여 도형을 관리한다.

이 클래스의 주요 메서드로는 add(), remove(), edit(), paint() 등이 있다.

3. 아이디어 평가

- 효율성

연결 리스트를 사용하여 도형을 추가하거나 삭제하는 과정에서 시간 복잡도로 성능을 유지할 수 있다. 도형을 추가할 때마다 배열을 복사할 필요가 없어 메모리 사용이 효율적이다.

각 도형을 동적으로 생성하여 메모리를 할당함으로써 필요한 만큼만 메모리를 사용하게 된다. 이는 메모리 관리 측면에서도 유리하다.

- 다형성

Shape 클래스를 통해 다양한 도형을 일관되게 처리할 수 있어, 코드의 유지보수성이 높아진다.

4. 문제를 해결한 키 아이디어 또는 알고리즘 설명

- 다형성 활용

Shape 클래스를 통해 모든 도형의 공통 인터페이스를 정의한다. 이 클래스는 draw()라는 순수 가상 함수를 포함하고, 각 도형 클래스(Circle, Line, Rect)는 이 메서드를 오버라이드하여 자신의 방식으로 도형을 그리는 로직을 구현한다. 이를 통해 GraphicsEditor 클래스에서 다양한 도형을 일관되게 처리할 수 있었다.

- 연결리스트 기반 도형 관리

GraphicsEditor 클래스는 연결리스트를 사용하여 도형을 동적으로 관리한다. 도형을 추가할 때마다 새로 생성된 도형 객체를 리스트에 연결하고, 삭제할 때는 특정 인덱스를 찾아 해당 도형을 제거한다. 이 구조는 도형의 추가 및 삭제가 효율적이며, 메모리의 동적 관리가 가능하다.