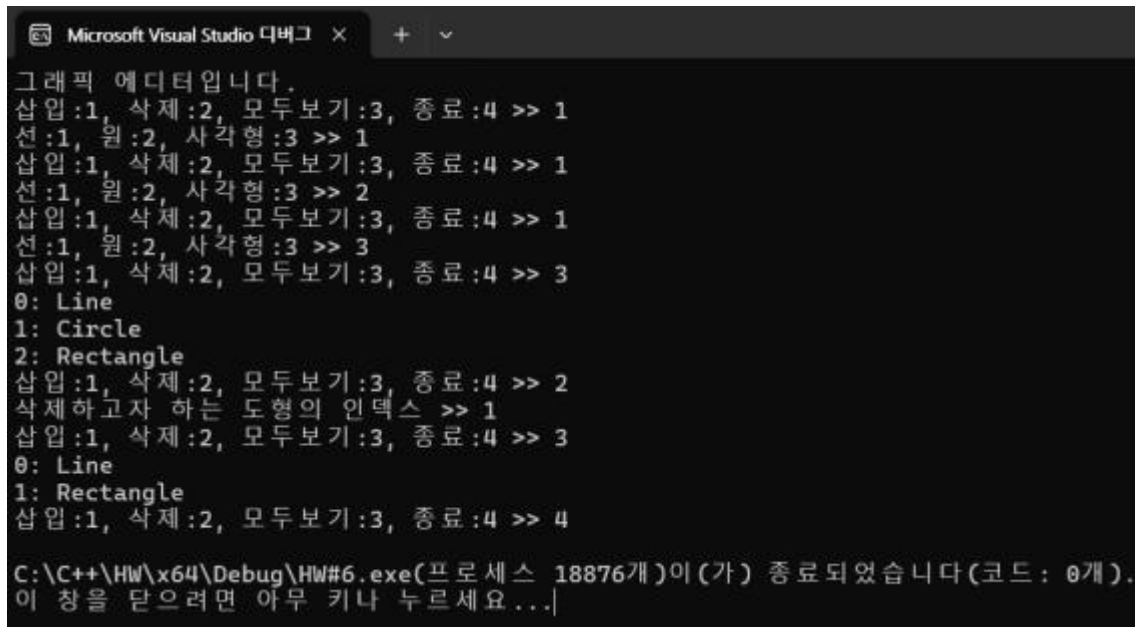


## ◆ 소스 수행 결과 화면



```
Microsoft Visual Studio 디버그
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#6.exe(프로세스 18876개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요 ...|
```

## ◆ 소스 구현 설명

### 1. 문제 정의

이 프로젝트의 목표는 그래픽 편집기를 구현하는 것이다. 사용자가 다양한 도형(원, 직선, 사각형 등)을 생성하고 이를 관리할 수 있도록 하며, 각각의 도형에 대한 작업(삽입, 삭제, 표시 등)을 수행할 수 있는 사용자 인터페이스를 제공한다. 이 시스템은 도형 객체를 동적으로 관리하고, 다양한 도형의 시각적 표현을 지원해야 한다.

### 2. 문제 해결 방법

- 'Shape'라는 기본 클래스를 생성하여 모든 도형의 공통 속성과 메소드를 정의하고, 이를 상속받아 각 도형별 클래스를 구현한다.
- 각 도형 클래스는 'Shape' 클래스로부터 상속받아, 고유한 'draw()' 메서드를 구현함으로써 다형성을 활용한다. 이를 통해 객체 생성 시 추가적인 조건 없이 각 도형의 기능을 사용할 수 있다.
- 벡터를 이용해 도형들의 리스트를 관리함으로써 동적 메모리 사용을 극대화하고, 도형의 삽입 및 삭제를 용이하게 처리한다.
- UI 클래스에서 사용자의 입력을 처리하여 사용자가 원하는 기능을 선택하고 수행할 수 있도록 한다.

### 3. 아이디어 평가

- **재사용성:** 도형에 대한 공통 클래스인 'Shape'를 정의함으로써 코드의 재사용성이 높아졌다. 새로운 도형을 추가할 경우 기존 코드를 수정할 필요 없이 새로운 클래스를 추가하면 된다.
- **효율성:** 벡터 사용 덕분에 동적으로 도형을 추가하고 삭제할 수 있어, 메모리 관리와 성능 측면에서 우수하다.

- 유지보수 용이성: 각 도형의 기능이 독립적으로 설계되어 있어, 특정 도형의 기능을 수정하더라도 전체 시스템에 미치는 영향을 최소화할 수 있다.

#### 4. 문제를 해결한 키 아이디어 또는 알고리즘 설명

- **Shape 클래스:** 공통 속성과 메서드를 포함하는 기본 클래스이다. 순수 가상 메서드 'draw()'를 선언하여 자식 클래스에서 구현하도록 하였다.

- **도형 클래스:** 'Circle', 'Line', 'Rectangle' 등 다양한 도형 클래스를 정의하고, 각각의 'draw()' 메서드에서 도형의 속성을 출력하도록 구현하였다.

- **GraphicEditor 클래스:** 도형 추가, 삭제 및 출력 메서드를 포함하며, 도형 관리의 핵심 비즈니스 로직을 구현한다. 사용자 입력을 바탕으로 도형을 관리하는 명령어 구조를 제공한다.

- **UI 클래스:** 사용자와의 상호작용을 처리하는 클래스이다. 메뉴를 통해 사용자의 입력에 따라 도형을 추가하고 삭제하며, 전체 도형을 표시할 수 있는 기능을 제공한다.

이러한 구조를 통해 프로그램은 쉽게 확장할 수 있으며, 다양한 도형의 기능을 효율적으로 관리할 수 있다.