

Shape.h 헤더파일

```
#pragma once
class Shape {
protected:
    virtual void draw() = 0;
public:
    void paint();
};
```

Shape.cpp 파일

```
#include <iostream>
#include "Shape.h"
using namespace std;

void Shape::paint() {
    draw();
}
```

Circle.h 헤더파일

```
#include "Shape.h"

class Circle : public Shape {
protected:
    virtual void draw();
};
```

Circle.cpp 파일

```
#include <iostream>
#include "Shape.h"
#include "Circle.h"
using namespace std;

void Circle::draw() {
    cout << "Circle" << endl;
}
```

Rect.h 헤더파일

```
#include "Shape.h"

class Rect : public Shape {
```

```
protected:
    virtual void draw();
};
```

Rect.cpp 파일

```
#include <iostream>
#include "Shape.h"
#include "Rect.h"
using namespace std;

void Rect::draw() {
    cout << "Rectangle" << endl;
}
```

Line.h 헤더파일

```
#include "Shape.h"

class Line : public Shape {
protected:
    virtual void draw();
};
```

Line.cpp 파일

```
#include <iostream>
#include "Shape.h"
#include "Line.h"
using namespace std;

void Line::draw() {
    cout << "Line" << endl;
}
```

Main.cpp 파일

```
#include <iostream>
#include "Shape.h"
#include "Circle.h"
#include "Rect.h"
#include "Line.h"
#include <vector>

using namespace std;
```

```

int main() {

    Rect r;
    Line l;
    int num = 0;
    int menu = 0;
    vector<Shape*> v;

    cout << "그래픽 에디터입니다." << endl;

    while (true)
    {
        cout << "삽입: 1, 삭제: 2, 모두보기: 3, 종료: 4 >> ";
        cin >> num;

        if (num == 1)
        {
            cout << "선: 1, 원: 2, 사각형: 3 >> ";
            cin >> menu;
            if (menu == 1)
            {
                Line* l = new Line();
                v.push_back(l);
            }
            else if (menu == 2)
            {
                Circle* c = new Circle();
                v.push_back(c);
            }
            else if (menu == 3)
            {
                Rect* r = new Rect();
                v.push_back(r);
            }
        }
        else if (num == 2)
        {
            int index;
            cout << "삭제하고자 하는 도형의 인덱스 >> ";
            cin >> index;
            v.erase(v.begin() + index);
        }
        else if (num == 3)
        {
            for (int i = 0; i < v.size(); i++)
            {
                cout << i << ": ";
                v.at(i)->paint();
            }
        }
        else if (num == 4) break;
    }
}

```

main.cpp 파일을 제외한 나머지 파일은 문제의 보기와 같이 작성했다

사용자가 선택할 메뉴의 번호를 입력받을 변수 num과 세부적인 메뉴의 번호를 다루는 menu 변수를 선언해서 에디터 형식으로 작동하도록 코드를 작성했다

또한 shape 타입의 객체를 저장할 벡터 v를 선언해서 객체를 저장할 수 있도록 작성했다.

삽입 부분에서는 menu변수의 값에 따라 선, 원, 사각형의 객체를 생성하고 push_back을 이용해서 생성한 객체를 벡터 v에 넣는 코드이다.

삭제 부분에서는 num변수에 따라 삭제부분을 실행하게 되는데, 인덱스를 추가로 입력받고 입력받은 인덱스의 도형을 삭제하는 역할을 한다.

벡터 v에 대고 erase를 호출하는데, 여기서는 인자로 v의 첫번째 값인 0번째에 인덱스값을 더해서, 해당 인덱스의 객체를 삭제하도록 작성했다.

모두보기 부분에서는 for문을 통해 벡터 v의 값을 처음부터 끝까지 프린트하는 부분이다.

벡터 v에 대고 at(i)를 호출해서 해당 인덱스의 접근하고 추상클래스로 선언된 paint()를 호출해서 출력하는 코드이다.

종료 부분에서는 break를 사용해서 루프를 빠져나가도록 작성했다.