

```

✓ #include<iostream>
  | #include "Shape.h"

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

```

shape클래스의 함수들이 사라진것 말고는 전 과제와 차이가 없다. vector<Shape*>를 이용해서 그래픽 편집기를 만들어야 하기에 전 과제에서 GraphicEditor 부분을 새로 만들었다.

```

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

✓ void Shape::paint() {
  |     draw();
  | }

```

202104231 신윤섭

```

#include<iostream>
#include<vector>
#include"UI.h"
using namespace std;

class GraphicEditor {
    vector<Shape*> v;
    vector<Shape*>::iterator it;
public:
    GraphicEditor();
    void insert();
    void deleteShape();
    void showAll();
};

#include"Shape.h"
#include"Circle.h"
#include"Rect.h"
#include"Line.h"
#include"GraphicEditor.h"
using namespace std;

void GraphicEditor::insert() {
    int n = UI::shapeMenu();
    if (n == 1)
        v.push_back(new Line());
    else if (n == 2)
        v.push_back(new Circle());
    else if (n == 3)
        v.push_back(new Rect());
    else cout << "입력 에러" << endl;
}

void GraphicEditor::deleteShape() {
    int n = UI::deleteMenu();
    Shape* del;
    it = v.begin();
    for (int i = 0; i < n; ++i)
        ++it;
    del = *it;
    it = v.erase(it);
    delete del;
}

void GraphicEditor::showAll() {
    for (int i = 0; i < v.size(); ++i) {
        cout << i << " : ";
        v[i]->paint();
    }
}

GraphicEditor::GraphicEditor() {
    cout << "그래픽 에디터입니다" << endl;
    while(true) {
        int m = UI::mainMenu();
        if (m == 1)
            insert();
        else if (m == 2)
            deleteShape();
        else if (m == 3)
            showAll();
        else if (m == 4) break;
        else cout << "입력 에러" << endl;
    }
}

```

vector 클래스를 활용하기 위해 vector를 include해주고 GraphicEditor클래스를 만들어 주었다. 그다음 Shape클래스 타입의 vector v를 선언해주고 v의 포인터 it도 선언해 준다.

GraphicEditor클래스는 생성자와 insert, delete, showAll 함수를 가지고 있다.

insert는 생성할 도형을 선택하고 생성하는 함수이다.

UI::shapeMenu에서 리턴받은 값에따라 생성할 도형을 정한다. 만약 1이 었다면 v.push_back(new Line())을 한다. 이는 동적 생성한 Line객체를 vector의 마지막에 원소로 추가하는 것이다. 나머지도 동일하다.

deleteShape는 UI::deleteMenu에서 리턴받은 숫자번째의 도형을 지우는 것이다. it = v.begin()을 해서 it을 vector의 첫 원소를 가리키게하고. 지울 도형의 번째 수 만큼 it을 +해서 포인터 it이 해당 도형을 가리키게한다. 그다음 Shape타입의 객체 포인터 del을 it이 가리키고 있는 도형을 가리키게한다. 그다음 it = v.erase(it)으로 해당 원소를 삭제하고 delete del로 동적생성한 도형을 반환한다. showAll은 for문을 사용해 v.size만큼 반복해서 vector의 각 원소에 paint함수를 실행시켜 출력한다.

그 다음은 GraphicEditor의 생성자로 GraphicEditor 객체 생성시 "그래픽 에디팅비니다"를 출력후 무한루프에 들어간다 UI::mainMenu로 리턴 받은 값에따라 도형을 생성할지 삭제할지 보여줄지 에디터를 종료할지 판단한다.

```
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 >> 1
선:1, 원:2, 사각형:3 >> 2
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 3
0: Line
1: Circle
2: Rectangle
3: Circle
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 2
삭제하고자 하는 도형의 인덱스 >> 2
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 3
0: Line
1: Circle
2: Circle
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 2
삭제하고자 하는 도형의 인덱스 >> 0
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 3
0: Circle
1: Circle
삽입:1, 삭제:2, 모두보기:3, 종료:4 >> 4

C:\Users\dbstj\source\repos\Project3\x64\Debug\Project3.exe(프
```