

# HW03

202204103 ict융합공학부 이승훈

202204023 ict융합공학부 김보민

## [Shape.h]

```
#ifndef SHAPE_H
#define SHAPE_H

class Shape {
protected:
    virtual void draw() = 0;

public:
    void paint();
};
#endif
```

## [Shape.cpp]

```
#include "Shape.h"
#include <iostream>

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

## [Circle.h]

```
#ifndef CIRCLE_H
#define CIRCLE_H
#include "Shape.h"

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

#### [Circle.cpp]

```
#include "Circle.h"
#include <iostream>

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

#### [GraphicEditor.h]

```
#ifndef GRAPHICEDITOR_H
#define GRAPHICEDITOR_H
#include <vector>
#include "Shape.h"

class GraphicEditor {
private:
    std::vector<Shape*> v;
    std::vector<Shape*>::iterator it;

public:
    GraphicEditor();
    void start();
    ~GraphicEditor();
};
#endif
```

#### [GraphicEditor.cpp]

```
#include "GraphicEditor.h"
#include "UI.h"
#include "Circle.h"
#include "Rect.h"
#include "Line.h"
#include <iostream>

GraphicEditor::GraphicEditor() {
    std::cout << "그래픽 에디터입니다.\n";
    start();
}

void GraphicEditor::start() {
    while (true) {
```

```

int n;
n = UI::selectMenu();
switch (n) {
case 1:
    n = UI::selectShape();
    switch (n) {
case 1:
        v.push_back(new Line());
        break;
case 2:
        v.push_back(new Circle());
        break;
case 3:
        v.push_back(new Rect());
        break;
    }
    break;
case 2:
    n = UI::selectDelIndex();
    if (n < v.size()) {
        it = v.begin();
        Shape* tmp = *(it + n);
        v.erase(it + n);
        delete tmp;
    }
    break;
case 3:
    UI::showAll(v, it);
    break;
case 4:
    return;
default:
    std::cout << "잘못 입력하셨습니다.\n";
    break;
}
}
}

```

```

GraphicEditor::~GraphicEditor() {
    for (it = v.begin(); it != v.end(); ++it) {
        delete *it;
    }
}

```

```
    }  
}
```

#### [Line.h]

```
#ifndef LINE_H  
#define LINE_H  
#include "Shape.h"  
  
class Line : public Shape {  
protected:  
    virtual void draw();  
};  
#endif
```

#### [Line.cpp]

```
#include "Line.h"  
#include <iostream>  
  
void Line::draw() {  
    std::cout << "Line" << std::endl;  
}
```

#### [Rect.h]

```
#ifndef RECT_H  
#define RECT_H  
#include "Shape.h"  
  
class Rect : public Shape {  
protected:  
    virtual void draw();  
};  
#endif
```

#### [Rect.cpp]

```
#include "Rect.h"  
#include <iostream>  
  
void Rect::draw() {  
    std::cout << "Rectangle" << std::endl;  
}
```

#### [UI.h]

```
#ifndef UI_H
#define UI_H
#include <vector>
#include "Shape.h"

class UI {
public:
    static int selectMenu();
    static int selectShape();
    static int selectDelIndex();
    static void showAll(std::vector<Shape*> &v, std::vector<Shape*>::iterator
&it);
};

#endif
```

#### [UI.cpp]

```
#include "UI.h"
#include <iostream>

int UI::selectMenu() {
    int n;
    std::cout << "삽입:1, 삭제:2, 모두보기:3, 종료:4 >> ";
    std::cin >> n;
    return n;
}

int UI::selectShape() {
    int n;
    std::cout << "선:1, 원:2, 사각형:3 >> ";
    std::cin >> n;
    return n;
}

int UI::selectDelIndex() {
    int n;
    std::cout << "삭제하고자 하는 도형의 인덱스 >> ";
    std::cin >> n;
    return n;
}
```

```
}
```

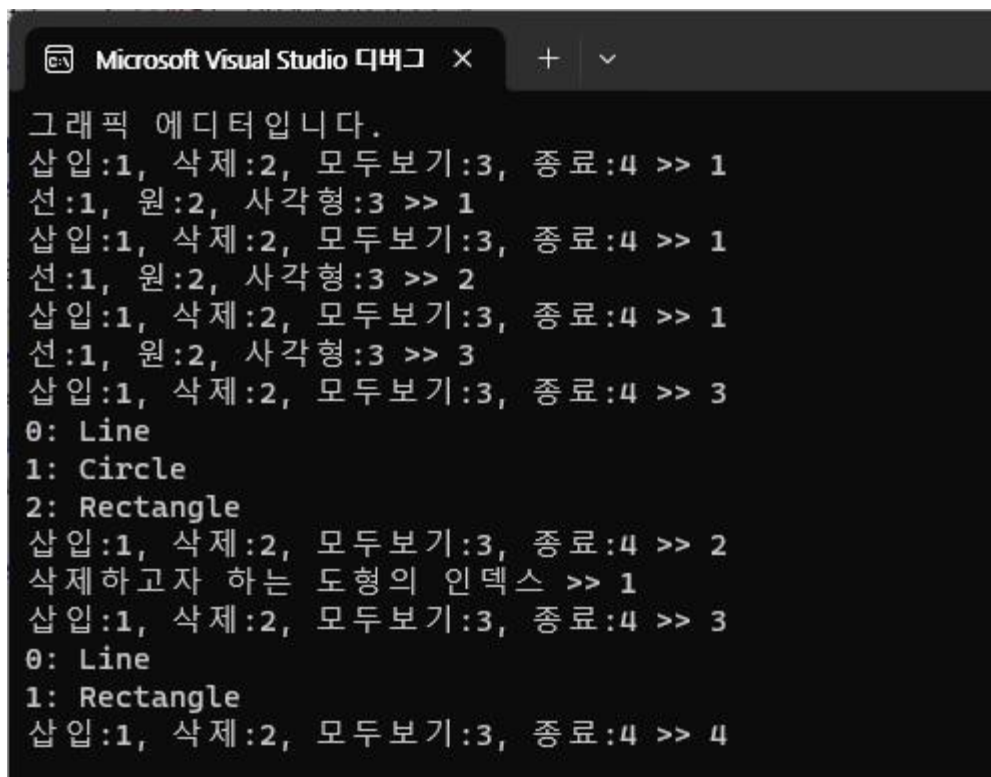
```
void UI::showAll(std::vector<Shape*> &v, std::vector<Shape*>::iterator &it) {  
    int i = 0;  
    for (it = v.begin(); it != v.end(); it++, i++) {  
        std::cout << i << ": ";  
        v.at(i)->paint();  
    }  
}
```

[main.cpp]

```
#include "GraphicEditor.h"
```

```
int main() {  
    GraphicEditor* graphicEditor = new GraphicEditor();  
  
    delete graphicEditor;  
    return 0;  
}
```

[실행결과]



```
Microsoft Visual Studio 디버그 x + v  
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
```