HW02

ICT융합공학부 202204023 김보민
ICT융합공학부 202204103 이승훈

< Shape.h >

```
#ifndef SHAPE_H
#define SHAPE_H

class Shape {
        Shape* next;

protected:
        virtual void draw() = 0;

public:
        Shape();
        virtual ~Shape();
        Shape* add(Shape* p);
        void paint();
        Shape* getNext();
};
#endif
```

< Shape.cpp >

```
#include "Shape.h"
#include <iostream>
using namespace std;
Shape::Shape() {
        next = NULL;
}
Shape::~Shape() {}
Shape* Shape::add(Shape* p) {
        this->next = p;
        return p;
}
void Shape::paint() {
        draw();
}
```

```
Shape* Shape::getNext() {
	return next;
}
```

< UI.h >

```
#ifndef UI_H
#define UI_H
#include <string>

class UI {
public:
	static int menu00();
	static int menu01();
	static int menu02();
};
#endif
```

< UI..cpp >

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

using namespace std;

int UI::menu00() {
	int key;
	cout << "삽입:1, 삭제:2, 모두보기:3, 종료:4 >>";
	cin >> key;
	return key;
}

int UI::menu01() {
	int key;
	cout << "선:1, 원:2, 사각형:3 >>";
	cin >> key;
	return key;
}

int UI::menu02() {
	int key;
	cout << "삭제하고자 하는 도형의 인덱스 >>";
	cin >> key;
```

```
        return key;
}
```

< GraphicEditor.h >

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

class GraphicEditor {
        Shape* pStart;
        Shape* pLast;

public:
        GraphicEditor();
        void insertItem(int type);
        void deleteItem(int index);
        void show();
        void run();
};
#endif
```

< GraphicEditor.cpp >

```
#include "GraphicEditor.h"
#include "Line.h"
#include "Circle.h"
#include "Rect.h"
#include "UI.h"
#include <iostream>
using namespace std;

GraphicEditor::GraphicEditor() {
        pStart = pLast = NULL;
}

void GraphicEditor::insertItem(int type) {
        Shape* p = NULL;
        switch (type) {
        case 1:
```

```
                p = new Line();
                break;
        case 2:
                p = new Circle();
                break;
        case 3:
                p = new Rect();
                break;
        default:
                break;
        }
        if (pStart == NULL) {
                pStart = p;
                pLast = p;
                return;
        }
        pLast->add(p);
        pLast = pLast->getNext();
}

void GraphicEditor::deleteItem(int index) {
        Shape* pre = pStart;
        Shape* tmp = pStart;
        for (int i = 0; i < index; i++) {
                pre = tmp;
                tmp = tmp->getNext();
        }
        if (tmp == pStart) {
                pStart = tmp->getNext();
                delete tmp;
        }
        else {
                pre->add(tmp->getNext());
                delete tmp;
        }
}

void GraphicEditor::show() {
        Shape* tmp = pStart;
        int i = 0;
        while (tmp != NULL) {
```

```
                cout << i++ << ": ";
                tmp->paint();
                tmp = tmp->getNext();
        }
}

void GraphicEditor::run() {
        cout << "그래픽 에디터입니다." << endl;
        int menu, index, type;
        while (true) {
                menu = UI::menu00();
                switch (menu) {
                case 1:
                        type = UI::menu01();
                        insertItem(type);
                        break;
                case 2:
                        index = UI::menu02();
                        deleteItem(index);
                        break;
                case 3:
                        show();
                        break;
                default:
                        return;
                }
        }
}
```

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

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

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

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

< Rect.h >

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

class Rect : public Shape {
```

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

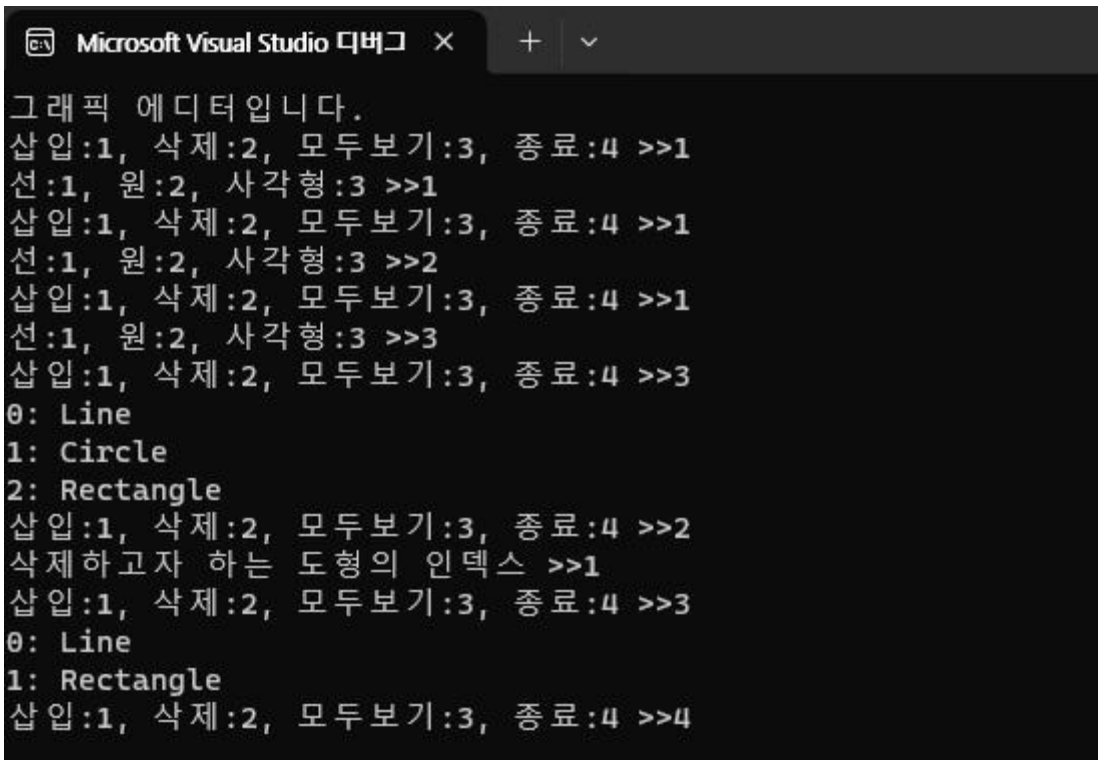
< Rect.cpp >

```
#include "Rect.h"
#include "Shape.h"

#include <iostream>
using namespace std;

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

< Main.cpp >

```
#include "GraphicEditor.h"
#include "Circle.h"
#include "Line.h"
#include "Rect.h"
#include "Shape.h"
#include "UI.h"
#include <iostream>
using namespace std;

int main() {
        GraphicEditor graphicEditor;
        graphicEditor.run();
}
```

< 실행결과 >


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