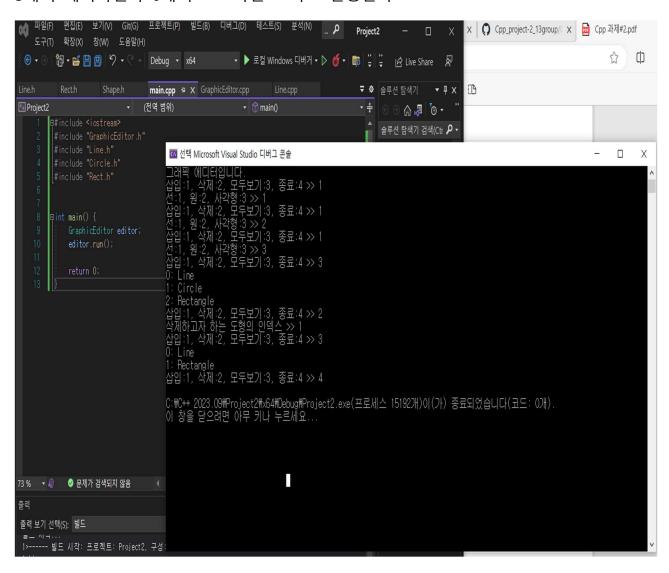
과제 #3

13조(전대원, 소재현)

5개의 헤더파일과 6개의 소스파일 그리고 실행결과



헤더파일

Circle.h

GraphicEditor.h

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

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

public:
    GraphicEditor();
    ~GraphicEditor();

    void run();
    void insert();
    void remove();
    void viewAll();
};
```

Line.h

Rect.h

Shape.h

소스파일

Circle.cpp

```
#include <iostream>
using namespace std;
#include "Shape.h"
#include "Circle.h"
void Circle::draw() {
            cout << "Circle" << endl;
}</pre>
```

GraphicEditor.cpp

```
#include <iostream>
#include "GraphicEditor.h"
#include "Line.h"
#include "Circle.h"
#include "Rect.h"
#include <vector>
using namespace std;
GraphicEditor::GraphicEditor() {
}
GraphicEditor::~GraphicEditor() {
    for (Shape* shape : v) {
       delete shape;
    v.clear();
}
void GraphicEditor::run() {
    cout << "그래픽 에디터입니다." << endl;
   while (true) {
```

```
int command;
       cout << "삽입:1, 삭제:2, 모두보기:3, 종료:4 >> ";
       cin >> command;
       switch (command) {
       case 1:
           insert();
           break;
       case 2:
           remove();
           break;
       case 3:
           viewAll();
           break;
       case 4:
           return;
       default:
           std::cout << "잘못된 입력입니다.\n";
           break;
       }
   }
}
void GraphicEditor::insert() {
    int shapeType;
   cout << "선:1, 원:2, 사각형:3 >> ";
   cin >> shapeType;
   Shape* p;
   switch (shapeType) {
   case 1:
       p = new Line;
       break;
   case 2:
       p = new Circle;
       break;
   case 3:
       p = new Rect;
       break;
   default:
       cout << "잘못된 입력입니다.\n";
       return;
   v.push_back(p);
}
void GraphicEditor::remove() {
    int index;
   cout << "삭제하고자 하는 도형의 인덱스 >> ";
   cin >> index;
    if (index < 0 \mid | index >= v.size()) {
       cout << "잘못된 인덱스입니다. \n";
       return;
   }
   delete v[index];
   v.erase(v.begin() + index);
```

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

Line.cpp

```
#include <iostream>
using namespace std;

#include "Shape.h"
#include "Line.h"

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

Main.cpp

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

int main() {
    GraphicEditor editor;
    editor.run();
    return 0;
}
```

Rect.cpp

Shape.cpp