Shape.h

11주차 과제 (전역 범위)

```
1  #ifndef Shape_H
2  #define Shape_H
3
4  class Shape {
5      Shape* next;
6  protected:
7      virtual void draw() = 0;
8  public:
9      Shape() { next = NULL; }
10     virtual ~Shape() { }
11     void paint() { draw(); }
12     Shape* add(Shape* p) { this->next = p; return p; }
13     Shape* getNext() { return next; }
14     void setNext(Shape* p) { this->next = p->next; }
15 };
16
17 #endif
```

1, 2, 17번 라인의 #ifndef, #define, #endif 명령어를 통해 헤더파일이 중복으로 정의되는것을 방지합니다.
4~15번 라인에 Shape 클래스를 생성하고 Shape 클래스 내에 가상함수 draw를 protected로 선언하고 shape의 생성자와 소멸자를 public으로 선언한다. 그리고 paint를 통해 Shape 객체가 아닌, 파생 클래스의 draw()를 호출하여 Shape를 이용하여 새로운 도형을 설정하고 포인터를 반환하는 함수를 구현합니다.

Circle.h

11주차 과제 — Circle

```
1  #ifndef Circle_H
2  #define Circle_H
3
4  class Circle : public Shape {
5  protected:
6      virtual void draw() {
7          cout << "Circle" << endl;
8      }
9  };
10
11 #endif
```

1, 2, 11번 라인의 #ifndef, #define, #endif 명령어를 통해 헤더파일이 중복으로 정의되는것을 방지합니다.
4~9번 라인에 Shape 클래스를 상속받은 Circle 클래스를 생성해 draw를 수행하고 Circle을 출력한다.

Rect.h

11주차 과제

```
1  #ifndef Rect_H
2  #define Rect_H
3
4  class Rect : public Shape {
5  protected:
6      virtual void draw() {
7          cout << "Rectangle" << endl;
8      }
9  };
10
11 #endif
```

1, 2, 11번 라인의 #ifndef, #define, #endif 명령어를 통해 헤더파일이 중복으로 정의되는것을 방지합니다.
4~9번 라인에 Shape 클래스를 상속받은 Rect 클래스를 생성해 draw를 수행하고 Rect을 출력한다.

Line.h

11주차 과제 — Line

```
1  #ifndef Line_H
2  #define Line_H
3
4  class Line : public Shape {
5  protected:
6      virtual void draw() {
7          cout << "Line" << endl;
8      }
9  };
10
11 #endif
```

1, 2, 11번 라인의 #ifndef, #define, #endif 명령어를 통해 헤더파일이 중복으로 정의되는것을 방지합니다.
4~9번 라인에 Shape 클래스를 상속받은 Line 클래스를 생성해 draw를 수행하고 Line을 출력한다.

UI.h

11주차 과제 — UI

```
1  #ifndef UI_H
2  #define UI_H
3
4  class UI {
5  public:
6      static int main_memu() {
7          int n;
8          cout << "삽입:1, 삭제:2, 모두보기:3, 종료:4 >> ";
9          cin >> n;
10         return n;
11     }
12     static int shape_menu() {
13         int n;
14         cout << "선:1, 원:2, 사각형:3 >> ";
15         cin >> n;
16         return n;
17     }
18     static int delete_menu() {
19         int n;
20         cout << "삭제하고자 하는 도형의 인덱스 >> ";
21         cin >> n;
22         return n;
23     }
24 };
25
26 #endif
```

1, 2, 26번 라인의 #ifndef, #define, #endif 명령어를 통해 헤더파일이 중복으로 정의되는것을 방지합니다.
4~24번 라인에 UI클래스를 생성하고 각각의 main_memu, shape_memu, delete_menu 등을 생성하고 각 메뉴에서 n값을 입력받아 n값을 리턴합니다.

GraphicEditor.h

11주차 과제 — GraphicEditor

```
1  #ifndef GraphicEditor_H
2  #define GraphicEditor_H
3
4  class GraphicEditor {
5      Shape* pStart;
6      Shape* pLast;
7      int count;
8  public:
9      GraphicEditor() { pStart = NULL; count = 0; }
10     void create(int num) {
11         switch (num) {
12         case 1:
13             if (count == 0) {
14                 pStart = new Line();
15                 pLast = pStart;
16             }
17             else
18                 pLast = pLast->add(new Line());
19             count++;
20             break;
21
22         case 2:
23             if (count == 0) {
24                 pStart = new Circle();
25                 pLast = pStart;
26             }
27             else
28                 pLast = pLast->add(new Circle());
29             count++;
30             break;
31
```

11주차 과제 — GraphicEditor

```
31
32         case 3:
33             if (count == 0) {
34                 pStart = new Rect();
35                 pLast = pStart;
36             }
37             else
38                 pLast = pLast->add(new Rect());
39             count++;
40             break;
41
42         }
43     }
44     void indelete(int num) {
45         Shape* p = pStart;
46         Shape* del = pStart;
47
48         if (num < count) {
49             for (int i = 0; i < num; i++) {
50                 p = del;
51                 del = del->getNext();
52             }
53             if (num == 0)
54                 pStart = p->getNext();
55             else
56                 p->setNext(del);
57             count--;
58             if (count == 1) pLast = pStart;
59             delete del;
60
61         }
62         else
63             cout << "인덱스를 잘못 입력하셨습니다." << endl;
64     }
```

GraphicEditor.h

```
65      void call() {
66          bool exit = true;
67          cout << "그래픽 에디터입니다." << endl;
68          while (exit) {
69              switch (UI::main_memu()) {
70              case 1:
71                  create(UI::shape_menu());
72                  break;
73              case 2:
74                  indelete(UI::delete_menu());
75                  break;
76              case 3: {
77                  Shape* p = pStart;
78                  for (int i = 0; i < count; i++) {
79                      cout << i << ": "; p->paint();
80                      p = p->getNext();
81                  }
82                  break;
83              }
84              case 4:
85                  exit = false;
86                  break;
87
88              }
89          }
90      }
91  };
92
93  #endif
```

GraphicEditor.h

1, 2, 93번 라인의 #ifndef, #define, #endif 명령어를 통해 헤더파일이 중복으로 정의되는것을 방지합니다.
4~91번 라인에 GraphicEditor클래스를 생성하고 Shape* pStart; 와 Shape* pLast; 로 각각 첫번째 도형과 마지막 도형을 가르키는 포인터를 선언하고, count 변수를 선언하여 현재 도형의 개수를 나타냅니다.
GraphicEditor() { pStart = NULL; count = 0; }로 각 시작 도형과 개수를 초기화 하고 switch 문을 통하여 각각의 도형을 리스트를 생성하고 도형의 개수를 증가시킵니다.
Indelete 함수를 선언하여 if문을 통해 생성된 count번호에 따라 해당 번호에 있는 도형을 삭제하고 전체 count를 하나 감소시키는 함수를 구현합니다.
그후 call함수를 선언하여 UI클래스를 상속받아 각각의 구현한 함수들을 swirch문을 통하여 경우에 맞게 동작 시킵니다.

main.cpp

11주차 과제 (전역 범위)

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

#include "Shape.h"
#include "Circle.h"
#include "Rect.h"
#include "Line.h"
#include "UI.h"
#include "GraphicEditor.h"

int main() {
    GraphicEditor* editor = new GraphicEditor;
    editor->call();
    delete editor;
}
```

5~10번 라인에 각각의 헤더 파일들을 include 하고, 12~16번 라인의 메인 함수에서 GraphicEditor 객체를 생성하고 프로그램을 실행한 이후 메모리를 해제합니다.