

#HW05

1. 문제

이 문제는 가상함수를 이용한 예제인 그림 9-13에 있던 Shape와 Circle, Rect, Line을 이용하여 그래픽 편집기를 만드는 문제입니다. 그래픽 편집기에서 추가적으로 구현해야 하는 기능은 삭제, 모두보기 기능이 있습니다.

2. 문제 해결 방법

그래픽 편집기를 만들기 위해 필요한 클래스와 멤버 변수는 다음과 같습니다.

GraphicEditor 클래스

- 멤버 변수

 - pStart, pLast, size(배열 리스트의 크기)

- 멤버 함수

 - run(); 실행함수

 - add(); 선택에 따라 도형을 리스트에 추가

 - del(); 인덱스 입력에 따라 해당 도형을 삭제

 - show(); 리스트를 출력

UI 클래스

- 멤버 함수

menu(); 삽입 삭제 모두보기 종료 선택

inputShape(); 삽입할 도형의 종류 선택

inputDel(); 삭제할 도형의 인덱스를 선택

+) 추가적으로 UI 클래스의 모든 멤버 함수들은 static

이번 문제의 핵심은 GraphicEditor 클래스입니다.

add();

먼저 add(); 함수부터 설명하자면, 이전 예제에서는 강제로 첫 번째 항목을 생성하고 시작했지만, 이번 예제에서는 첫 번째 항목이 없다면 생성해주어야합니다.

```
void GraphicEditor::add(int num) {  
    switch (num) {  
        case 1:  
            if (size == 0) {  
                pStart = new Line();  
                pLast = pStart;  
            }  
            else {  
                pLast = pLast->add(new Line());  
            }  
            size++;  
            break;  
    }  
}
```

size가 0이라면, 처음 도형을 생성하고 pStart와 pLast를 설정
이미 도형이 있다면, 도형을 생성하고 pLast를 생성한
도형으로 설정합니다.

switch-case 문을 사용하여 사용자로부터 받아온 도형을
종류에 맞게 생성합니다.

del():

del 함수는 target과 prev 변수를 사용하여 삭제할 도형을 삭제하고 prev에 저장되어있는 도형의 next를 삭제한 도형의 다음 도형으로 설정하면, 해당 인덱스의 도형을 삭제할 수 있습니다.

```
bool GraphicEditor::del(int num) {
    int cnt = 0;
    Shape* target = pStart;
    Shape* prev = NULL;
    if (num == 0) {
        pStart = pStart->getNext();
        delete target;
    }
    else {
        while ((target != NULL) && (num > cnt)) {
            prev = target;
            target = target->getNext();
            cnt++;
        }
        if (target == NULL) {
            cout << "잘못된 인덱스입니다." << endl;
            return false;
        }
        else {
            prev->setNext(target);
            delete target;
        }
    }
    size++;
}
```

show():

show 함수는 9-13 예제에 있는 그대로 작성했습니다.

```
// 현재 연결된 모든 도형을 화면에 그린다.  
Shape* p = pStart;  
while(p != NULL) {  
    p->paint();  
    p = p->getNext();  
}
```

```
void GraphicEditor::show() {  
    Shape* p = pStart;  
    int i = 0;  
    if (p == NULL)  
        cout << "도형을 생성해주세요." << endl;  
    else {  
        while (p != NULL) {  
            cout << i << ": ";  
            p->paint();  
            p = p->getNext();  
            i++;  
        }  
    }  
};
```

3. 아이디어 평가

add 함수는 Shape 클래스의 add 함수를 사용하여 쉽게 구현할 수 있지만, del 함수는 연결 리스트 개념 중 삭제하는 알고리즘을 알고 있어야 해결 할 수 있습니다.

알고리즘을 대략 설명하면 다음과 같습니다.

- 1) 삭제할 노드를 찾는다
- 2) 그 전 노드와 삭제할 노드의 다음 노드를 연결한다.
- 3) 노드를 삭제한다.

크게 3과정인데, 이를 구현하기 위해서는 다음 노드 뿐만 아니라 전 노드의 정보까지 필요하므로, prev 라는 변수가 필요합니다. prev와 next를 이용해서 리스트를 연결해주고, 삭제하는 알고리즘이 이번 과제의 중요한 알고리즘입니다.