

hw#06

202104255 윤여준

문제 정의

이번 문제는 저번 문제에서 구현했던 `GraphicEditor` 클래스를 `Vector` 컨테이너를 사용해서 다시 구현하는 예제입니다.

문제 해결 방법

이 문제는 `GraphicEditor` 예제와 `Vector` 컨테이너를 이해했다면 쉽게 해결할 수 있는 문제였습니다. 이전 `GraphicEditor` 예제에서는 삽입, 삭제, 모두보기 기능을 선형 리스트를 이용해서 구현했다면, 이번 예제에서는 `Vector` 컨테이너의 `push_back`, `erase`, `print` 함수를 이용해서 해결했습니다.

`add` -> `push_back`

`del` -> `erase`

`show` -> `print`

아이디어 평가

이번 문제를 해결 하기 위해서는 Vector 컨테이너의 함수들을 사용할 줄 알아야 했습니다.

```
void GraphicEditor::add(int n) {  
    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;  
        default:  
            cout << "메뉴를 잘못 선택하셨습니다." << endl;  
    }  
}
```

push_back

```
bool GraphicEditor::del(int n) {  
    vector<Shape*>::iterator it;  
    it = v.begin();  
  
    if (n > v.size() || n < 0)  
        return false;  
    else {  
        for (int i = 0; i < n; i++) {  
            it++;  
        }  
        v.erase(it);  
        return true;  
    }  
}
```

erase

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

print

push_back() --> Vector 컨테이너 마지막 배열에 삽입
erase() --> Vector 컨테이너 마지막 배열 삭제
print() --> Vector 컨테이너 모든 배열 출력