과제6 소스코드 설명 - 서신우

문제 정의

- Shape 클래스를 프로그램이 잘 실행되도록 적절히 수정한다.
- vector 컨테이너를 통해 도형들을 관리해야 한다.
- 사용자의 입력을 받아 도형들을 추가, 삭제, 열람할 수 있어야 한다.

문제 해결

- Shape 클래스에서 생성자와 가상 소멸자를 적절히 생성해 추후에 프로그램이 동작하는데 문제가 없도록 한다.
- Line, Circle, Rect 모든 도형들을 전부 다룰 수 있도록 Shape* 형식을 가진 벡터 컨테이너를 생성한다.
- 사용자에게 입력을 받아 삽입 동작에서는 벡터 컨테이너에 추가하고, 삭제 동작에서는 벡터 컨테이너에서 특정 인덱스에 있는 요소를 삭제하고, 모두 보기 동작에서는 반복문을 통해 컨테이너에 있는 요소들을 꺼내 보여주도록 한다.

Code

Shape.h

Shape 클래스에서는 생성자를 추가로 선언하고, 또 가상 소멸자를 추가하여 나중에 메모리 누수를 막아주는 기능을 한다.

```
class Shape {
protected:
    virtual void draw() = 0;
public:
    Shape(){ };
    virtual ~Shape(){ };
    void paint();
};
```

main.cpp

프로그램을 계속 돌리기 위한 while 문에 사용할 변수, 선택받은 동작을 넣어둘 변수, 사용자가 삽입 또는 삭제에 사용할 모양을 넣어둘 변수, Shape* 형식의 벡터 컨테이너를 위한 변수를 선언한다.

```
int main() {
    bool isRunning = true;
    int option;
    int shape;
    vector<Shape*> v;

    while (isRunning) {
        // ~~~~
    }
}
```

실행할 동작을 사용자로부터 받고 switch 문을 통해 구분한다..

```
cout << "삽입:1, 삭제:2, 모두보기:3, 종료:4 >> ";
cin >> option;
cout << endl;

switch (option) {
    case 1:
        {
                // 도형을 추가하는 코드
        }
        break;
    case 2:
        {
                // 도형을 삭제하는 코드
        }
        break;
    case 3:
        {
             // 도형을 전부 보여주는 코드
        }
        break;
```

```
    case 4:
        {
                // 프로그램을 종료시키는 코드
                isRunning = false;
        }
        break;
}
```

삽입하는 동작은 사용자로부터 삽입할 도형을 입력받고 해당하는 도형을 벡터 컨테이너에 추가한다.

```
cout << "선:1, 원:2, 사각형:3 >> ";
cin >> shape;
cout << endl;

if (shape == 1)
{
    v.push_back(new Line());
}
else if (shape == 2)
{
    v.push_back(new Circle());
}
else if (shape == 3)
{
    v.push_back(new Rect());
}
else
{
    cout << "해당 도형은 지원하지 않습니다." << endl;
}
```

삭제하는 동작은 사용자로부터 삭제할 도형을 입력받고 해당하는 도형을 벡터 컨테이너에서 삭제한다.

```
int index;
cout << "삭제하고자 하는 도형의 인덱스 : ";
cin >> index;

v.erase(v.begin() + index);
```

모두 보여주는 동작은 for 문을 통해 벡터 컨테이너에서 요소 하나하나를 꺼낸다.

```
for (int i = 0; i < v.size(); i++)
{
    cout << i << " : ";
    v.at(i)->paint();
}
```