

# 자바프로그래밍및실습 레포트

과제2 (7장)



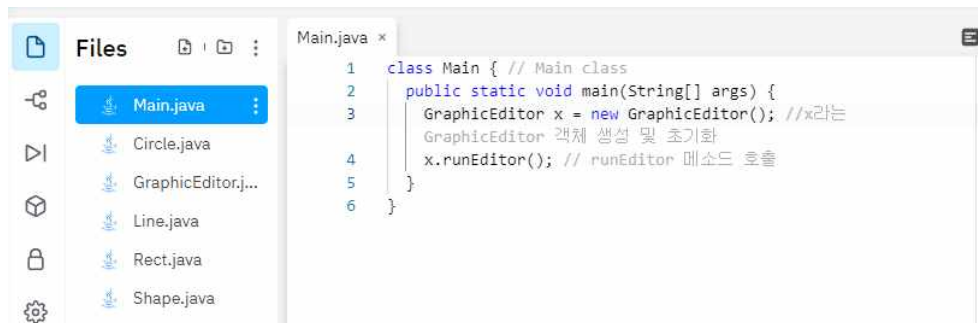
제출일	2021.11.17	전공	자동차기계공학과
과목	자바프로그래밍및실습	학번	21912317
담당교수	윤종희 교수님	이름	조현근

## ■ 서론

자바프로그래밍및실습 과목에서 과제 2(7장 실습문제 3번)에 대해서 코드의 구성과 동작 원리에 대한 간단한 설명과 결과 화면을 캡처해서 소개하도록 하겠다.

## ■ 본문

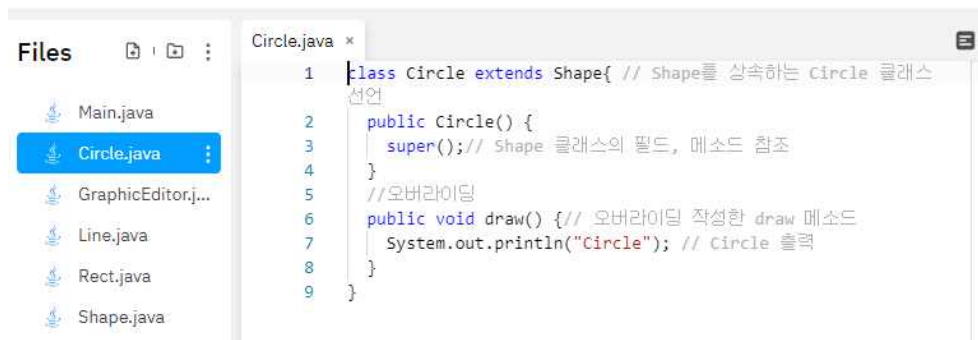
### Main.java



Main.java에서 x라는 GraphicsEditor 객체를 생성하고 초기화 하였고, GraphicsEditor 속 runEditor 메소드를 호출하도록 구성하였다.

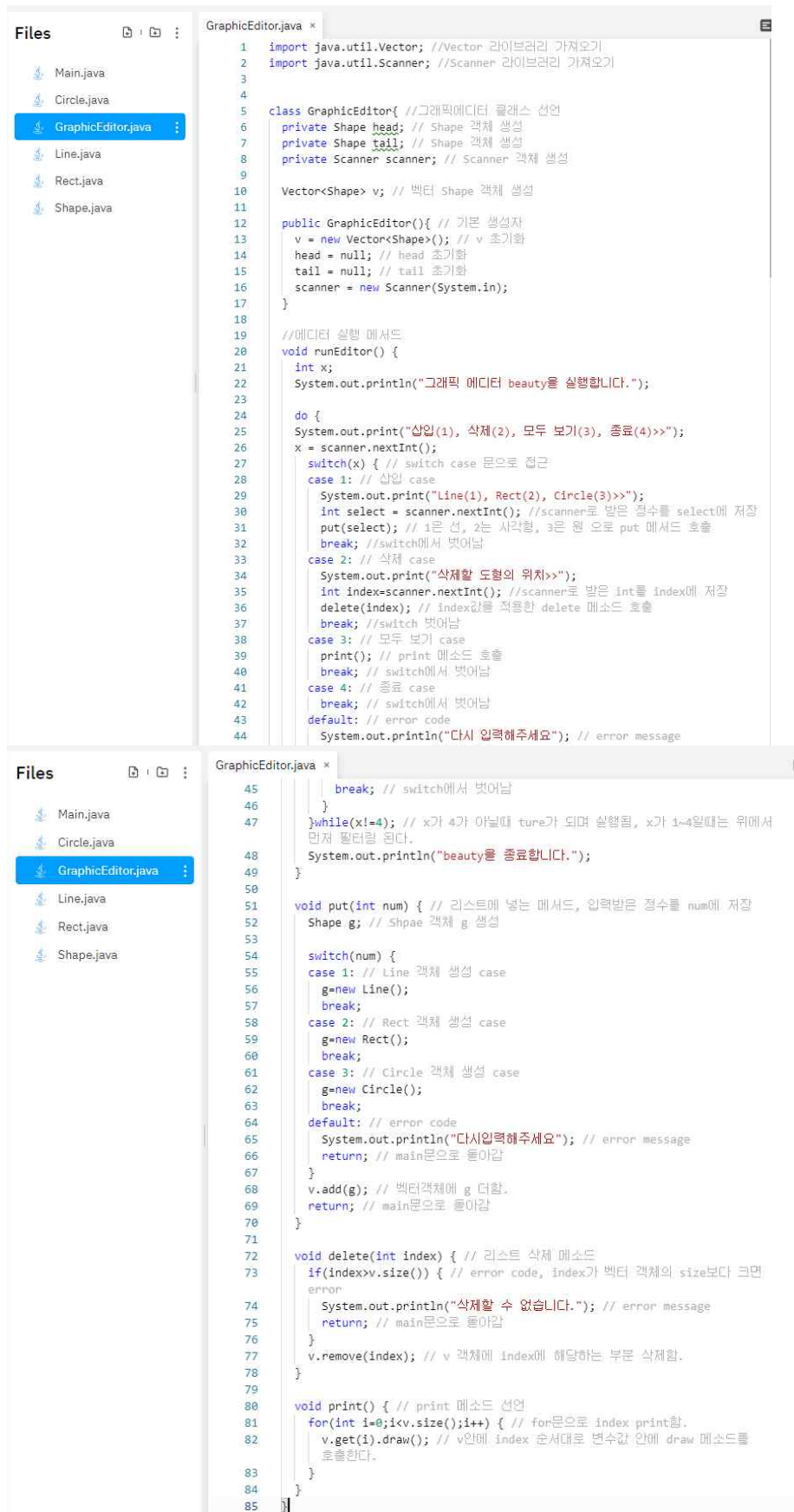
최대한 Main.java는 코드를 깔끔하게 보이기 위해 각각 class를 따로따로 .java로 만들었다.

### Circle.java



Circle.java에서 Shape를 상속하는 자식 class로 Circle을 선언하였고, 부모 클래스인 Shape class의 필드, 메소드를 참조하는 super객체를 생성하였다. draw를 오버라이딩하여 작성하였다.

## GraphicEditor.java

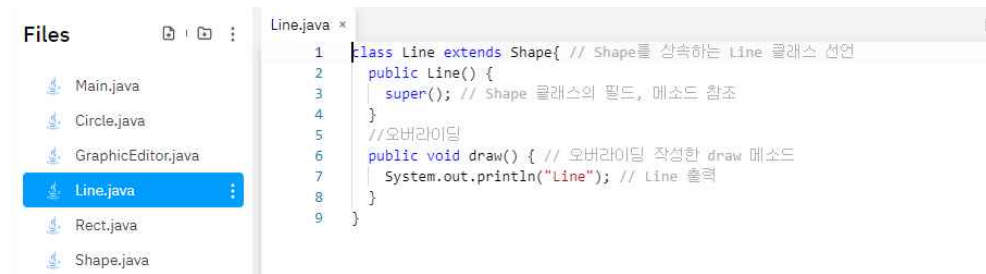


```
1 import java.util.Vector; //Vector 라이브러리 가져오기
2 import java.util.Scanner; //Scanner 라이브러리 가져오기
3
4
5 class GraphicEditor{ //그래픽에디터 클래스 선언
6     private Shape head; // Shape 객체 생성
7     private Shape tail; // Shape 객체 생성
8     private Scanner scanner; // Scanner 객체 생성
9
10    Vector<Shape> v; // 벡터 Shape 객체 생성
11
12    public GraphicEditor(){ // 기본 생성자
13        v = new Vector<Shape>(); // v 초기화
14        head = null; // head 초기화
15        tail = null; // tail 초기화
16        scanner = new Scanner(System.in);
17    }
18
19    //에디터 실행 메서드
20    void runEditor() {
21        int x;
22        System.out.println("그래픽 에디터 beauty를 실행합니다.");
23
24        do {
25            System.out.print("삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>");
26            x = scanner.nextInt();
27            switch(x) { // switch case 문으로 접근
28                case 1: // 삽입 case
29                    System.out.print("Line(1), Rect(2), Circle(3)>>");
30                    int select = scanner.nextInt(); //scanner로 받은 정수를 select에 저장
31                    put(select); // 1은 선, 2는 사각형, 3은 원 으로 put 메서드 호출
32                    break; //switch에서 벗어나
33                case 2: // 삭제 case
34                    System.out.print("삭제할 도형의 위치>>");
35                    int index=scanner.nextInt(); //scanner로 받은 int를 index에 저장
36                    delete(index); // index값을 적용한 delete 메소드 호출
37                    break; //switch 벗어나
38                case 3: // 모두 보기 case
39                    print(); // print 메소드 호출
40                    break; // switch에서 벗어나
41                case 4: // 종료 case
42                    break; // switch에서 벗어나
43                default: // error code
44                    System.out.println("다시 입력해주세요"); // error message
45
46                break; // switch에서 벗어나
47            }
48        }while(x!=4); // x가 4가 아닐때 true가 되며 실행됨, x가 1~4일때는 위에서
49        //먼저 필터링 된다.
50        System.out.println("beauty를 종료합니다.");
51    }
52
53    void put(int num) { // 리스트에 넣는 메서드, 입력받은 정수를 num에 저장
54        Shape g; // Shape 객체 g 생성
55
56        switch(num) {
57            case 1: // Line 객체 생성 case
58                g=new Line();
59                break;
60            case 2: // Rect 객체 생성 case
61                g=new Rect();
62                break;
63            case 3: // Circle 객체 생성 case
64                g=new Circle();
65                break;
66            default: // error code
67                System.out.println("다시입력해주세요"); // error message
68                return; // main문으로 돌아감
69        }
70        v.add(g); // 벡터객체에 g 더함.
71        return; // main문으로 돌아감
72    }
73
74    void delete(int index) { // 리스트 삭제 메소드
75        if(index>v.size()) { // error code, index가 벡터 객체의 size보다 크면
76            error
77            System.out.println("삭제할 수 없습니다."); // error message
78            return; // main문으로 돌아감
79        }
80        v.remove(index); // v 객체에 index에 해당하는 부분 삭제함.
81    }
82
83    void print() { // print 메소드 선언
84        for(int i=0;i<v.size();i++) { // for문으로 index print함.
85            v.get(i).draw(); // v안에 index 순서대로 변수값 안에 draw 메소드를
86            //호출한다.
87        }
88    }
89}
```

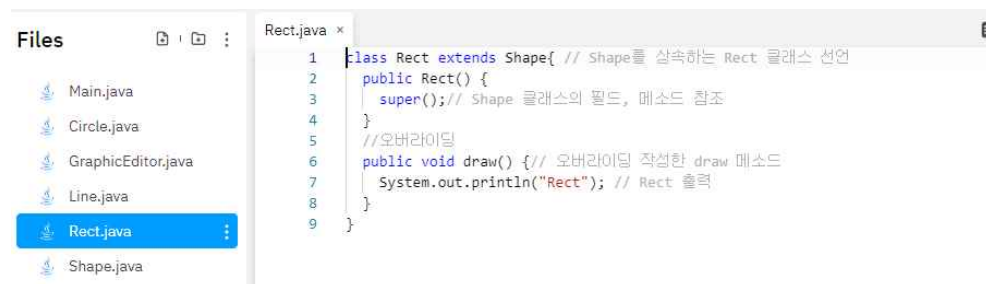
GraphicEditor.java에서 GraphicEditor class선언 및 Shape, Scanner 객체 생성하였고, runEditor 메소드는 do while문으로 구성하였고 이 메소드 안에서 put, delete,

print메소드를 호출하는 방식으로 코드를 구성하였다.

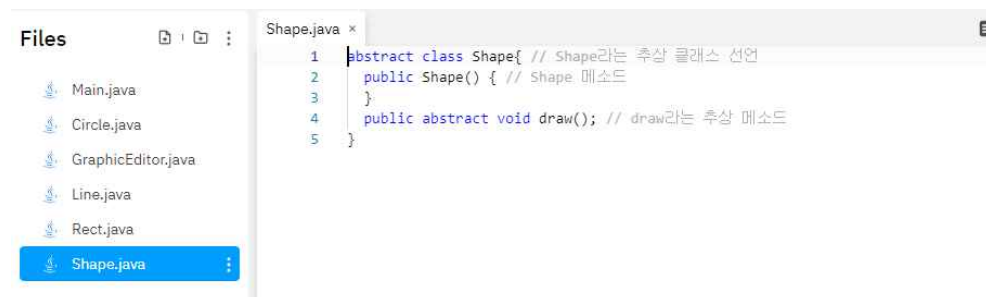
## Line.java



## Rect.java



## Shape.java



Line.java, Rect.java에서는 Circle.java와 마찬가지로 코드를 구성하였다.

Shape.java는 추상 class로 Shape를 선언하였으며 draw라는 추상 메소드를 만들었다,

아래 사진은 결과물이다.

```
❖ javac -classpath ./run_dir/junit-4.12.jar:target/dependency/* -d . Circle.java GraphicEd: Q X
ava Line.java Main.java Rect.java Shape.java
❖ java -classpath ./run_dir/junit-4.12.jar:target/dependency/* Main
그래픽 에디터 beauty를 실행합니다.
삽입 (1), 삭제 (2), 모두 보기 (3), 종료 (4)>>1
Line(1), Rect(2), Circle(3)>>2
삽입 (1), 삭제 (2), 모두 보기 (3), 종료 (4)>>1
Line(1), Rect(2), Circle(3)>>3
삽입 (1), 삭제 (2), 모두 보기 (3), 종료 (4)>>3
Rect
Circle
삽입 (1), 삭제 (2), 모두 보기 (3), 종료 (4)>>2
삭제할 도형의 위치>>3
삭제할 수 없습니다.
삽입 (1), 삭제 (2), 모두 보기 (3), 종료 (4)>>4
beauty를 종료합니다.
❖
```

## ■ 결론

과제를 미루지 않고 더 빨리 공부하기 시작했으면 코드를 잘 작성할 수 있었을 것 같은데 아쉽다. 그리고 내 java를 비롯한 객체지향적 사고방식 및 프로그래밍 실력이 많이 부족했다. 하지만 전 보다 class선언하는 부분에 있어서는 자신감이 생겼고, abstract 및 오버라이딩, Vector에 관련된 개념이 확실히 잡혔다.