

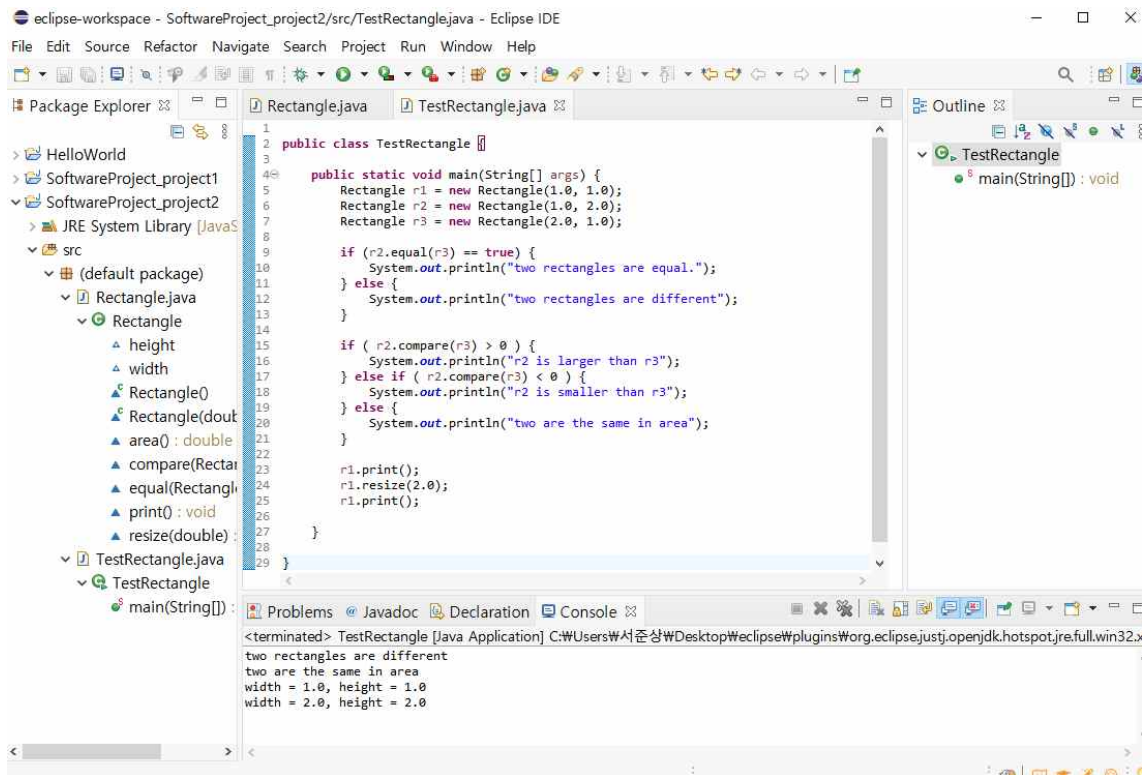
소프트웨어프로젝트

Project 2

소프트웨어학과 학과
서준상 이름
20201672 학번
2021/03/27(토) 날짜

1. 최종 테스트 과정 화면

- TestRectangle 작동 결과



rectangle.c 파일에 나와있는 대로

r1에는 (1.0, 1.0), r2에는 (1.0, 2.0), r3에는 (2.0, 1.0)을 넣고 작동시킨 결과,

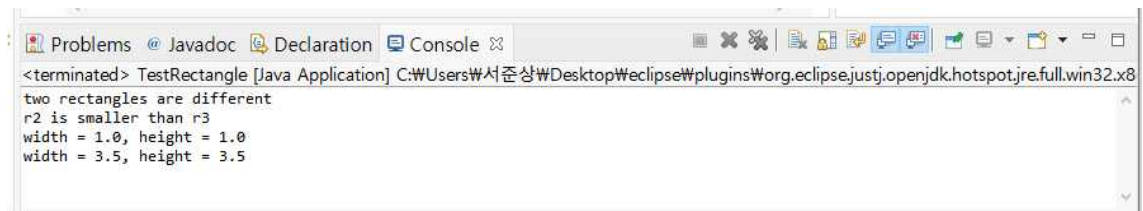
two rectangles are different (서로 다른 사각형)

two are the same in area (넓이는 같음) 이라는 결과가 나왔으며,

r1.resize(2.0)을 한 결과,

처음에 print() 했을 때 1.0, 1.0으로 나오던 r1의 밑변과 높이가 2.0, 2.0으로 바뀌었습니다.

```
Rectangle r1 = new Rectangle(1.0, 1.0);    r1.print();
Rectangle r2 = new Rectangle(2.0, 4.0);    r1.resize(3.5);
Rectangle r3 = new Rectangle(3.0, 3.0);    r1.print();
```



값을 바꿔 입력해도, 정확한 결과가 나옵니다.

- r2의 넓이는 $2 \times 4 = 8$, r3는 $3 \times 3 = 9$ 로 r2가 작으며, r1.resize(3.5)도 문제없이 작동.

2. 설계 노트

```
double width;  
double height;
```

```
Rectangle() {  
    this(1.0, 1.0);  
}
```

```
Rectangle(double x, double y) {  
    this.width = x;  
    this.height = y;  
}
```

가장 먼저, 사각형의 밑변과 높이를 저장하는 부분입니다.

double을 통해 밑변과 높이 변수를 선언해준 뒤,

Rectangle r1 = Rectangle(x,y)를 선언했을 때, r1.width에 x, r1.height에 y가 저장되도록 하였습니다.

print(), equal(Rectangle r), compare(Rectangle r), resize() 부분은

circle.java에 나온 것과 같은 방식으로,

r1.print()는 사각형 r1의 밑변과 높이를 출력,

r1.equal(r2)은 두 사각형이 같은지를 확인,

- 밑변과 높이가 서로 반대인 경우는 다른 것으로 나오도록 작성하였습니다.

- 예) 밑변 1, 높이 2인 사각형과, 밑변 2, 높이 1인 사각형은 다름(differet)으로 함.

r1.compare(r2)는 두 사각형의 넓이가 같은지를 확인,

r1.resize()는 r1의 밑변과 높이가 (x 괄호안의 정수)가 되도록 수정하지만,

area() 부분은, 원의 넓이와 사각형의 넓이를 구하는 방법이 다르기 때문에, circle.java에 나온 식에서 알맞게 수정하였습니다.

- 사각형의 넓이 = 밑변 x 높이 = r1.width x r1.height

마지막으로, class TestRectangle에선,

각 사각형 r1, r2, r3를 선언하고, rectangle.c의 main()에서 출력한 것과 같은 내용

- equal을 통해 r2와 r3가 같은 사각형인지 출력

- compare를 통해 r2와 r3의 넓이를 비교하여 출력 (결과가 0보다 크면 r2가 더 넓다)

- r1의 처음 값을 print를 통해 출력하고, resize한 값을 print를 통해 다시 출력

을 출력하도록 작성하였습니다.

3. 비교

	c 타입	java 타입	차이
1. 구조체 struct	struct rectangle{ }; 를 이용하여 작성	Rectangle() Rectangle(double x, double y) 를 이용하여 작성	java는 c 함수의 형식으로 구조체를 선언. 여러 형식의 함수 를 같은 이름으로 작성 가능 (왼쪽의 예에서도 Rectangle이 2회 사용)
2. 함수 호출	equal(r1, r2) 형식	r1.equal(r2) 형식	함수를 호출할때, 함수명이 아니라 객체명(r1)이 앞. 함수는 객체에 부속된다.
3. 출력	printf("내용%f내용", 변수);	System.out.println ("내용"+ 변수 +"내용");	c와 달리 %d, %f 등을 사용하지 않 고, '+'를 이용해 변수와 문장을 연 결하여 나타낼 수 있다.

java는 좀 더 변수(객체, object)를 중심으로 하는 언어인 것 같습니다.

c는 표의 2번처럼, r1, r2 각 객체들이 어떤 작업(equal)을 하는지가 더 중점적이라면,
java는 r1.equal(r2)와 같이, r1이 코드의 맨 앞에 위치하면서, 객체인 r1에 더 집중합니다.

그 외엔,

```
Rectangle() {
    this(1.0, 1.0);
}

Rectangle(double x, double y) {
    this.width = x;
    this.height = y;
}
```

이처럼 선언한 Rectangle(double x, double y)의 형식을 this를 통해 다른 곳에서도 사용할 수 있다는 점도 차이점인 것 같습니다.

따라서, java는 객체가 어느 부분에 사용되는지를 파악하기 쉽고, 객체 간의 관계를 더 명확히 알 수 있다면, 반대로 c는 코드가 어떤 흐름/절차로 작동하는지를 읽기 쉬운 것 같습니다.

4. 소스 프로그램

```
public class Rectangle {  
  
    double width;  
    double height;  
  
    Rectangle() {  
        this(1.0, 1.0);  
    }  
  
    Rectangle(double x, double y) {  
        this.width = x;  
        this.height = y;  
    }  
  
    void print() {  
        System.out.println("width = " + width + ", height = " + height );  
    }  
  
    boolean equal(Rectangle r) {  
        if ( (this.width == r.width) && (this.height == r.height) ) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    double area() {  
        return width*height;  
    }  
  
    double compare(Rectangle r) {  
        return this.area() - r.area();  
    }  
  
    void resize(double ratio) {  
        width *= ratio;  
        height *= ratio;  
    }  
}
```

// TestRectangle 부분

```
public class TestRectangle {  
    public static void main(String[] args) {  
        Rectangle r1 = new Rectangle(1.0, 1.0);  
        Rectangle r2 = new Rectangle(1.0, 2.0);  
        Rectangle r3 = new Rectangle(2.0, 1.0);  
  
        if (r2.equal(r3) == true) {  
            System.out.println("two rectangles are equal.");  
        } else {  
            System.out.println("two rectangles are different.");  
        }  
  
        if ( r2.compare(r3) > 0 ) {  
            System.out.println("r2 is larger than r3");  
        } else if ( r2.compare(r3) < 0 ) {  
            System.out.println("r2 is smaller than r3");  
        } else {  
            System.out.println("two are the same in area");  
        }  
  
        r1.print();  
        r1.resize(2.0);  
        r1.print();  
    }  
}
```

5. 자체평가표

평 가 표

평가 항목	학생 자체 평가 (리포트 해당 부분 표시 및 간단한 의견)	평가 (빈칸)	점수 (빈칸)
Java Style - 동작? Java 스타일로 작성했을 때 동작? - 요구 사항 만족? 즉, 진정한 Java 스타일?	레포트의 '1. 최종테스트 화면'과 '2. 설계노트' 부분 요구사항을 만족하여 JAVA스타일로 작성하였다고 생각합니다.		
두 방법 비교 - 평가 항목은? - 한눈에 보이게?	'3. 비교' 부분 평가항목 : 함수호출 방식, 코드 형식 등 표로 작성하였습니다.		
기타			
총평/계			

* 학생 자체 평가는 점수에 반영되지 않음.

* 학생 스스로 자신의 보고서를 평가하면서, 체계적으로 프로젝트를 마무리하도록 유도하는 것이 목적임.