1. **상속**

상속을 통해 자식클래스에서 부모클래스의 정보를 가져다 쓸 수 있고, 불필요한 수고를 덜어줄 수 있다.

예)



위와 같은 Triangle 클래스는 Shape라는 클래스를 상속받기 때문에 extends를 붙여서 상속되었음을 선언할 수 있다.

1. **캡슐화**

정보은닉을 위해 전역변수를 설정할 때 Private를 사용한다.

예) 

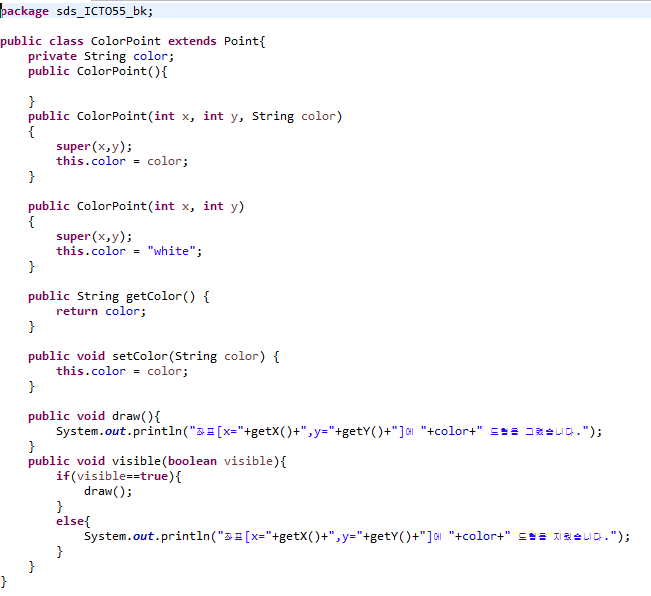
위와 같은 Rect라는 클래스 안에 캡슐화한 변수 height와 width를 보게 되면 앞에 Private를 써놓은 것을 확인할 수 있다. 그렇기 때문에 getter와 setter를 통해 값을 불러오거나 지정해야만 한다.

1. **다형성**

* **오버로딩**

메서드를 한 클래스 안에서 여러 번 중복으로 정의할 수 있는 것이다.

예)



위의 코드를 보게 되면 ColorPoint라는 메서드가 세 개 선언된 것을 볼 수 있다. 근데 여기서 다른 점은 매개변수를 가지고 있는 것과 없는 것, 그리고 매개변수의 개수로 구분이 되는데 color값을 x, y 다음에 입력하지 않고 메서드를 불러오게 되면 자동으로 white라는 색으로 정의가 되게 예시를 들어보았다. 이에 대한 결과는 맨 아래의 결과를 확인해 보면 알 수 있다.

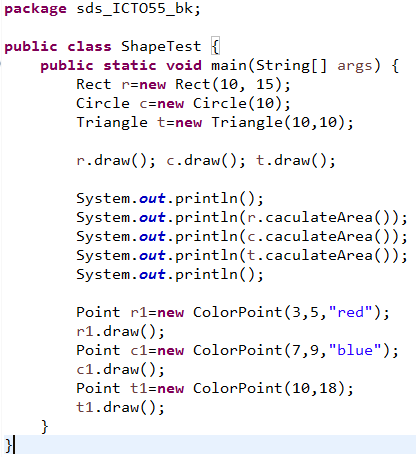
* **오버라이딩**

상속관계에서 똑같은 메서드를 사용하지만 다른 결과가 나오는 것을 말한다.

예)

Shape를 상속 받은 Rect, Triangle, Circle 클래스를 보게 되면 모두 draw()라는 메서드를 가지고 있다. 하지만 메인에서 컴파일을 해보면 각자 각 클래스에서 선언해 준 메서드를 불러오게 된다. 이에 대한 결과는 아래의 결과 사진을 보면 확인할 수 있다.

**< 메 인 >**



**< 결 과 >**





