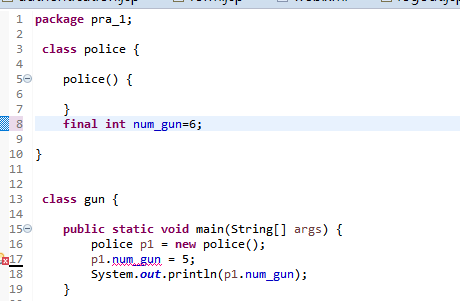
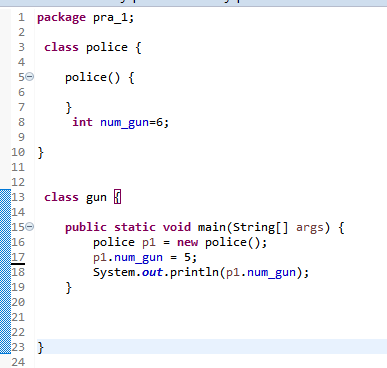
Final 변수 ( 한번 선언한 경우 변환이 불가능 그냥 int로 선언할 경우는 가능)



왼쪽은 그냥 int선언 오른쪽은 final int선언 차이를 명확한 차이

Final 메소드

오버라이딩하거나 숨길수 없음을 의미한다.

Final 클래스

-해당 클래스는 상속할 수 없음을 의미하고 말 그대로 마지막 클래스를 나타낸다.

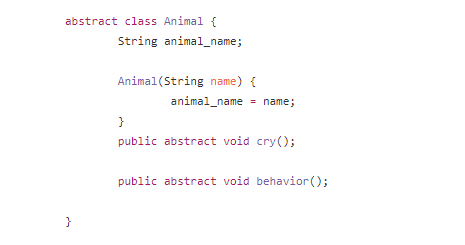
\*\*\*\* 그러면 왜 static final을 사용하는가 ?

생성자로 static final을 사용하게 되는 경우 초기의 값이 변화가 불가능하게 설정할 수 있습니다. 예를들어 국어 점수 만점의 경우 어딜 가도 100점이 기준이므로 이럴 때 static final로 변수 값을 설정하는게 효율적인 코딩입니다.

추상 클래스 (abstract class)

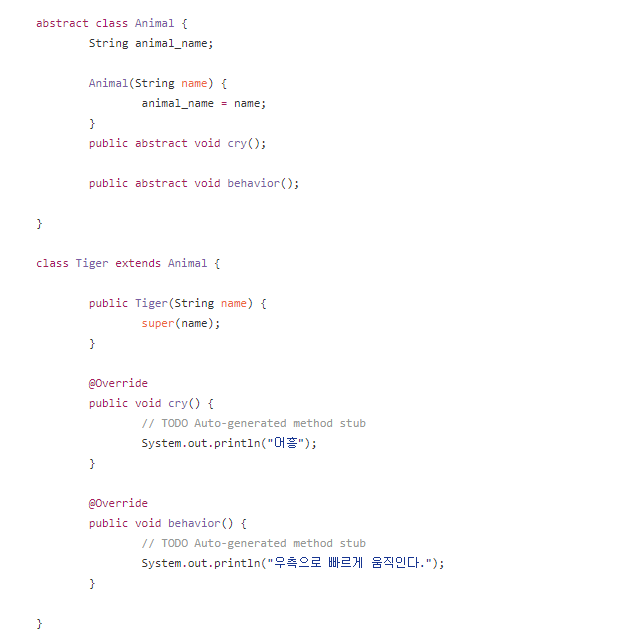
추상클래스는 일반 클래스 보다 추상적인 개념을 의미한다.

Point1 : 메서드, 멤버필드, 생성자를 정의 할 수 있지만, 추상메서드는 내용을 정의 할 수 없다.



Cry와 behavior같은 경우 정의를 할 수 없기 때문에 다음과 같이 초기화 선언되었다.

추상 클래스의 추상메서드를 정의하면 클래스 또한 추상클래스 여야 하고 추상클래스를 상속받은 클래스에서도 반드시 추상메서드를 오버라이드 해줘야한다. 단 추상클래스를 상속했다고 그 클래스가 추상클래스가 되는 것은 아니다. 메소드도 마찬가지로 상속했다고 추상 메소드가 되는 것은 아니다.



동물을 상속받은 호랑이는 추상클래스가 아니고 추상 메소드를 가지지도 않는다.

\*\*\* point2

재정의를 통해서 보다 구체적인 행동을 정의해 줄 수있다. (추상화의 의도)

Cry, behavior 모두 기존의 메소드 에서 한층 의미 부여된 결과값을 보여준다.

\*\* point 3

추상클래스는 extends로 상속을 받으며 다중 상속은 불가하다.

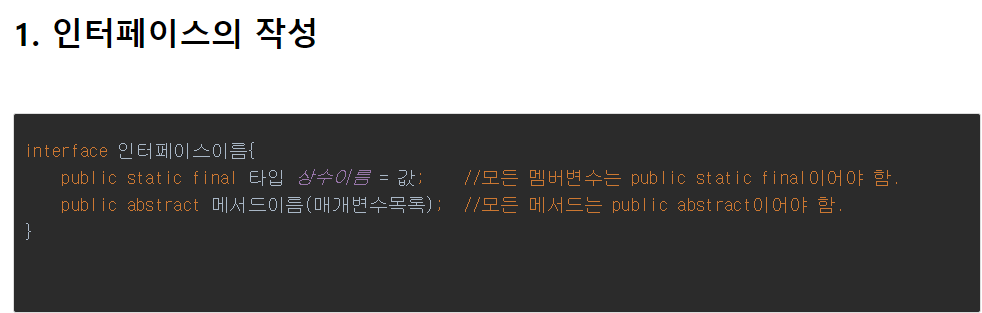


생성자에 인자로 String name을 받고 이것을 super(name)을 통하여 상위 클래스의 생성자로 보낸다. 이렇게 사용하여 추상클래스에서 정의했던 생성자를 이용하게 되고 추상클래스의 멤버 필드인 animal\_name에 등록 할 수 있습니다.

#인터페이스

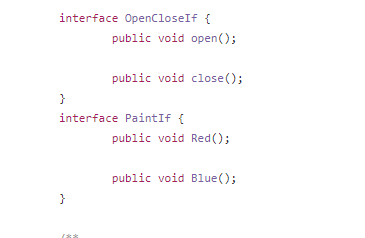
-인터페이스는 추상클래스보다 제한적인 성격을 가집니다.

-인터페이스는 멤버필드와 추상메서드만 정의 할 수 있다.

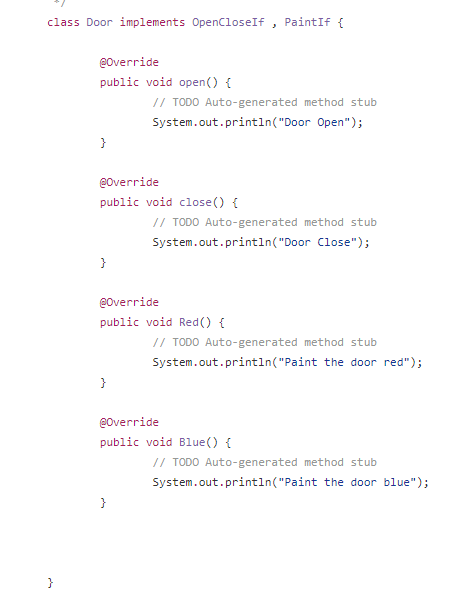


- 상속을 받으려면 extends를 사용하면 안되고 implements를 사용해야 한다.

- 인터페이스는 다중 상속이 가능하고 추상 메소드처럼 몸통을 가질 수 없다.



* 추상메소드처럼 정의를 하지 않고 사용한다.



재정의(오버라이딩) 할 때에는 몸통 부분에 표현한다. 상속처럼 사용할 시에는 implemets 를 사용하며 다중상속이 가능하다. ( 쉼표를 사용해서! )

## abstract 와 interface의 차이

Abstract는 조금 더 class의 성격을 가지고 있고 interface는 그와 다르게 메서드나 멤버 변수만을

정의하여 다른 클래스에서 이것을 사용하도록 강제하는 역할에 충실하다. 또 다른점은 인터페이스는 implements라는 키워드로 상속을 하기 때문에 다중상속이 가능하지만 abstract 는 클래스적 성향으로 다중 상속이 불가능하다.

**●멤버 필드(Member Field)**

**◎형식\_ 필드의 선언**

접근\_제한자 [지정\_예약어] 자료형 (클래스형) 필드명 [=값];

ex) private static final double pi= 3.141592;

     public transient String str= "Member Field";

**◎Field에 대한 지정 예약어**

|  |  |
| --- | --- |
| ﻿①static : | 하나의 클래스로 발생된 모든 객체가 공유하는 필드를 선언 할때 사용  객체명이 아닌 클래스명으로 접근 |
| ②final : | 상수형 필그로 한번 값이 지정되면 값을 변경하지 못하는 필드를  선언할 때 사용 |
| ③static final : | static의 속성과 final의 속성이 혼합된 형태 |
| ④transient : | 임시 메모리 필드, 네트워크로의 전송이나 파일전송 시에 그 값 자체가 없는  것과 같은 효과를 나타낼 필드를 선언할 때 사용 |

그렇다면 final 과 static final의 차이는 파이널은 값을 초기화 할수 있다( 사용자한테 입력을 받아서 또는 즉시 값의 초기화 ) final int a; (이게 가능)

Static final은 선언과 동시에 초기화를 해주어야 한다. (static final int a; 이거 불가능)