

추상

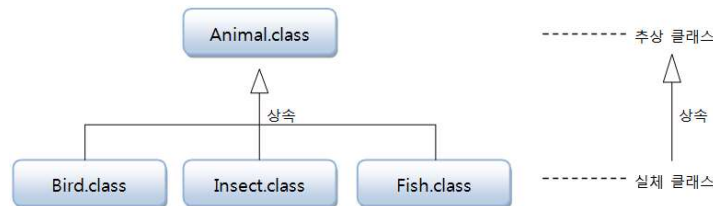


● 추상(abstract)

- 실체들 간에 공통되는 특성을 추출한 것
 - 예1: 새, 곤충, 물고기 → 동물 (추상)
 - 예2: 삼성, 현대, LG → 회사 (추상)

● 추상 클래스(abstract class)

- 실체 클래스들의 공통되는 필드와 메소드 정의한 클래스
- 추상 클래스는 실체 클래스의 부모 클래스 역할 (단독 객체 X)



*실체 클래스: 객체를 만들어 사용할 수 있는 클래스





추상

Robot Media Laboratory

- **실체 클래스의 공통된 필드와 메소드의 이름 통일할 목적**
 - 실체 클래스를 설계자가 여러 사람일 경우,
 - 실체 클래스마다 필드와 메소드가 제각기 다른 이름을 가질 수 있음
- **실체 클래스를 작성할 때 시간 절약**
 - 실체 클래스는 추가적인 필드와 메소드만 선언
- **실체 클래스 설계 규격을 만들고자 할 때**
 - 실체 클래스가 가져야 할 필드와 메소드를 추상 클래스에 미리 정의
 - 실체 클래스는 추상 클래스를 무조건 상속 받아 작성



● 추상 클래스

Robot Media Laboratory

● 추상 클래스(**abstract class**)

- 추상 메소드를 가진 클래스
- Python은 생성시 abc(**a**bstr**a**ct **b**ase **c**lass)외부 모듈을 가지고 와야 사용 가능
- 추상 메서드는 호출할 일이 없으므로 빈 메서드로 만듦

```
class 추상클래스이름(metaclass=ABCMeta):
```

```
    @abstractmethod
```

```
    def 메서드이름(self):
```

```
        pass
```





"Impossible" + "Dream"

`__add__(self, other)`



연산자	수식에	내부적인 함수 호출
덧셈	$x + y$	<code>x.__add__(y)</code>
뺄셈	$x - y$	<code>x.__sub__(y)</code>
곱셈	$x * y$	<code>x.__mul__(y)</code>
지수	$x ** y$	<code>x.__pow__(y)</code>
나눗셈(실수)	x / y	<code>x.__truediv__(y)</code>
나눗셈(정수)	$x // y$	<code>x.__floordiv__(y)</code>
나머지	$x \% y$	<code>x.__mod__(y)</code>
비트 왼쪽 이동	$x \ll y$	<code>x.__lshift__(y)</code>
비트 오른쪽 이동	$x \gg y$	<code>x.__rshift__(y)</code>
비트 AND	$x \& y$	<code>x.__and__(y)</code>
비트 OR	$x y$	<code>x.__or__(y)</code>
비트 XOR	$x \wedge y$	<code>x.__xor__(y)</code>
비트 NOT	$\sim x$	<code>x.__invert__()</code>

