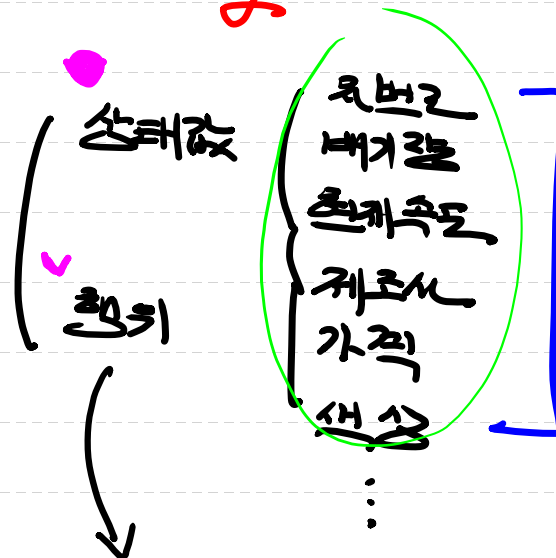


01/24

- 구조적 프로그래밍 (초차적 프로그래밍)
- 객체지향 프로그래밍

class를 이용해 객체 modeling

자동화



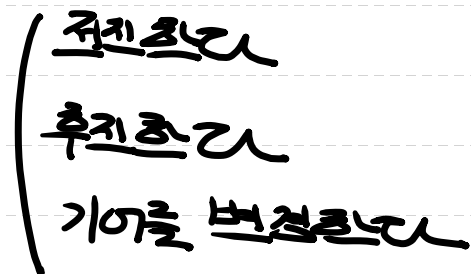
Java

field

C++

member  
Variable

member  
function



method

python

변수 → property

함수 → method

class Car(object):

# 모든 class안에는 특별한 형태의 함수를 사용할 수 있어요.

✓ # \_\_이름\_\_() => magic function

def \_\_init\_\_(self):

self.car\_number = '123' ✓

self.car\_price = 1000

self.car\_speed = 0

✓ car1 = Car()

car2 = Car()

\_\_ : dot operator

Stack

memory

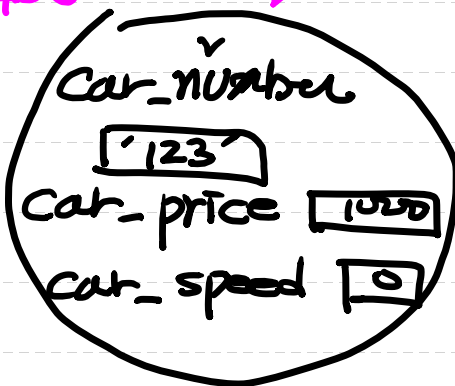
Car1

300

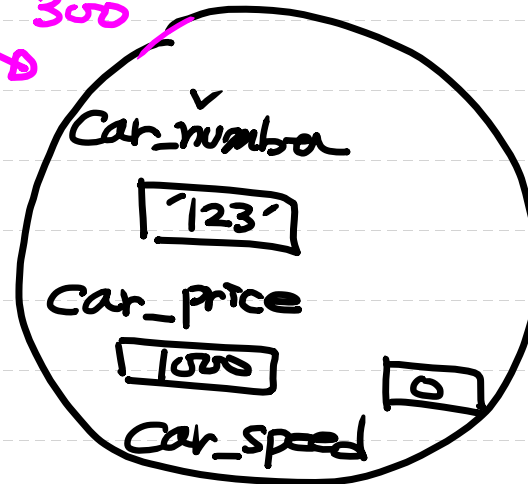
Car2

400

400



300



<class Car>

--init--() -> initializer

```

class Car(object):
    is_speeding = 120 # class variable

    # initializer
    def __init__(self, car_number, car_price, car_speed):
        self.car_number = car_number # instance variable
        self.car_price = car_price # instance variable
        self.car_speed = car_speed

    def __str__(self):
        return self.car_number

car1 = Car('123', 1000, 0)

```

car2 = Car('456', 2000, 10)

memory

--init--()

Self | 300

car\_number

'123'

...

stack

Car 1

300

Car 2

400

400

car\_number  
'456'

instance namespace

300

is\_speeding 120

car\_number '123'

...

instance namespace

instance (객체)

<class Car>

is\_speeding 120

--init--() ~

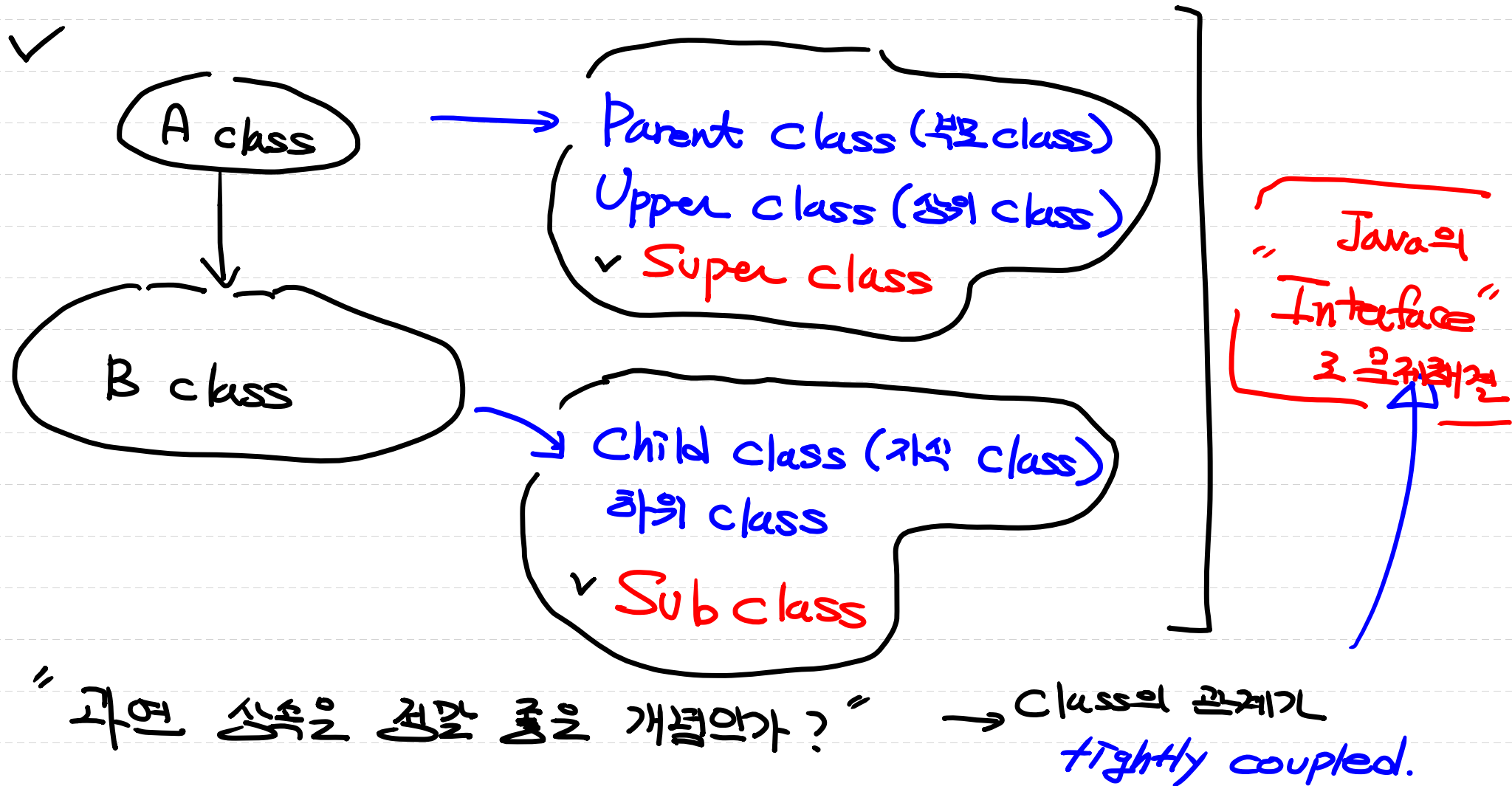
--str--() ~

class namespace

"name space"

## ① 상속 (Inheritance)

- 객체지향은 대표적 특징중의 하나.
- 큰 class의 특징을 이어받아 다른 class를 확장해서 생성하는 기법.
- python의 모든 class는 상속관계에 있습니다.



✓ 상속관계기 성립하려면 ✓

두개의 class간에 특별한 관계기 성립

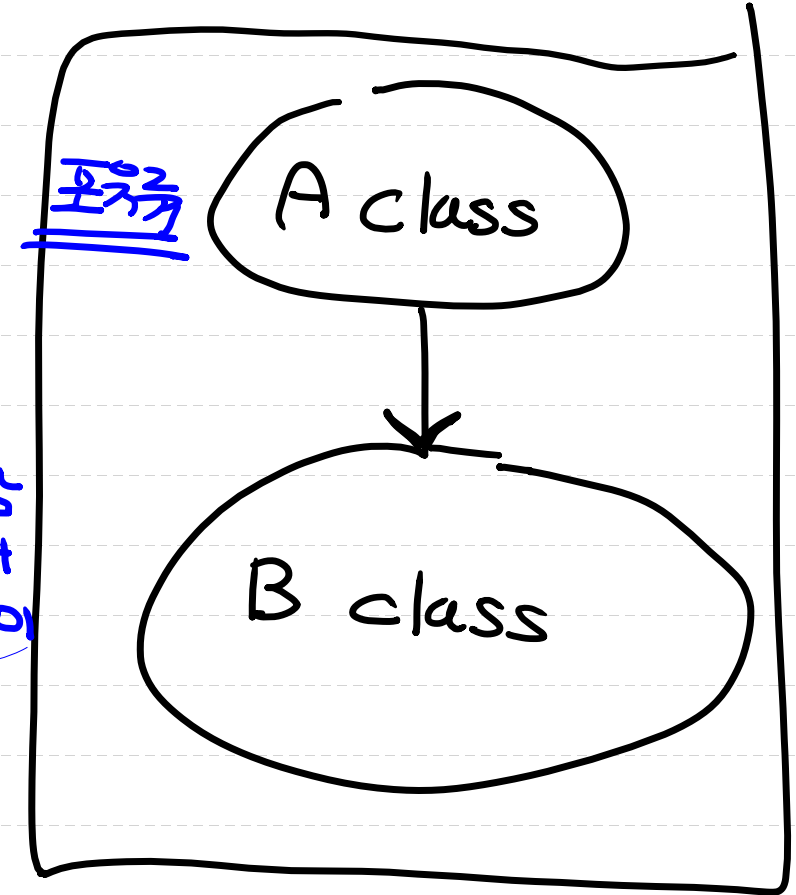


IS-A relationship  
(IS-A 관계)



Sub class is a Superclass (성립하고 그 역관계는 성립하지 X)

사람	도둑	(○)
도둑	사람	(X)



사람  
도둑  
원형이

\* [ 특정 목적을 위해 Superclass를 이용해야겠다는데  
그 기능을 더 확장한 subclass로 대처가능 ] \* ⇒ 'Django'

\* → 다형성 (polymorphism)

✓ my\_module.py

```
my_var  
    3.141592  
my_func()  
class Car():  
    ?
```

import my\_module

my\_module • my\_var

Memory

my\_module

300

300

2471 ✓

my\_var

3.1415

my\_func

class Car