클래스 상세 이해(상속, 다중상속)

- description : 클래스 상세 이해(상속, 다중상속)
- author : 도봉산핵주먹
- email : hylee@repia.com
- lastupdate : 2020-06-25

클래스 상세 이해(상속, 다중상속)

예제 코드

```
# Section07-2
# 파이썬 클래스 상세 이해
# 상속, 다중상속

# 예제1
# 상속 기본
# 슈퍼클래스 및 서브클래스 -> 모든 속성, 메소드 사용 가능


class Car:
    """Parent Class"""
    def __init__(self, tp, color):
        self.type = tp
        self.color = color

    def show(self):
        # print('Car Class "Show" Method!')
        return 'Car Class "Show" Method!'


class BmwCar(Car):
    """Sub Class"""

    def __init__(self, car_name, tp, color):
        super().__init__(tp, color) # super() 부모 클래스
        self.car_name = car_name

    def show_model(self) -> None: # -> None 힌트
        return 'Your Car Name : %s' % self.car_name


class BenzCar(Car):
    """Sub Class"""

    def __init__(self, car_name, tp, color):
```

```
        super().__init__(tp, color)
        self.car_name = car_name

    def show(self):
        super().show()
        return 'Car Info : %s %s %s' % (self.car_name, self.color,self.type)

    def show_model(self) -> None:
        return 'Your Car Name : %s' % self.car_name


# 일반 사용
print("#==== 일반사용 ====")
model1 = BmwCar('520d', 'sedan', 'red')

print(model1.color)  # Super
print(model1.type)  # Super
print(model1.car_name)  # Sub
print(model1.show())  # Super
print(model1.show_model())  # Sub
print()
# Method Overriding
print("#==== Method Overriding ====")
model2 = BenzCar("220d", 'suv', 'black')
print(model2.show())
# 부모와 자식 클래스에 똑같은 메소드가 있으면 자식 메소드가 호출됨
print()

# Parent Method Call
print("#==== Parent Method Call ====")
model3 = BenzCar("350s", 'sedan', 'silver')
print(model3.show())
print()
# Inheritance Info
print("#==== Inheritance Info ====")
print('Inheritance Info : ', BmwCar.mro())
print('Inheritance Info : ', BenzCar.mro())
print()

# 예제2
# 다중 상속
print("#==== 다중 상속 ====")
class X():
    pass


class Y():
    pass


class Z():
```

```
    pass


class A(X, Y):
    pass


class B(Y, Z):
    pass


class M(B, A, Z):
    pass

# .mro() -> 상속형태를 List형으로 반환한다.
print(M.mro())
print(A.mro())
```

실행 콘솔

```
#===== 일반사용 =====
red
sedan
520d
Car Class "Show" Method!
Your Car Name : 520d

#==== Method Overriding ====
Car Info : 220d black suv

#==== Parent Method Call ====
Car Info : 350s silver sedan

#==== Inheritance Info ====
Inheritance Info :  [<class '__main__.BmwCar'>, <class '__main__.Car'>,
<class 'object'>]
Inheritance Info :  [<class '__main__.BenzCar'>, <class '__main__.Car'>,
<class 'object'>]

#===== 다중 상속 =====
[<class '__main__.M'>, <class '__main__.B'>, <class '__main__.A'>, <class
'__main__.X'>, <class '__main__.Y'>, <class '__main__.Z'>, <class 'object'>]
[<class '__main__.A'>, <class '__main__.X'>, <class '__main__.Y'>, <class
'object'>]
```

Tip

도봉산핵주먹, python, 상속, 다중상속

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:ai:python:%ED%81%B4%EB%9E%98%EC%8A%A4_%EC%83%81%EC%84%B8_%EC%9D%B4%ED%95%B4_%EC%83%81%EC%86%8D_%EB%8B%A4%EC%A4%91%EC%83%81%EC%86%8D&rev=1593064316

Last update: **2022/03/10 19:52**