

170p

6.3.4 다형성

10 + 5

"test" + "H"

다형성(polymorphism)이라는 용어는 '여러 가지 형태를 가질 수 있는 능력'을 말한다. 예를 들면 파이썬에서 플러스(+) 기호는 산술연산의 덧셈기능과 문자열 연산에서 결합기능을 갖는다. 이처럼 플러스라는 기호 하나로 두 가지 기능을 갖는 능력을 다형성이라고 한다. 그렇다면 객체지향프로그래밍에서 다형성은 어떤 의미를 갖고 있는지를 살펴보자. '하나의 참조변수로 여러 타입(type)의 객체를 참조할 수 있는 것'으로 정의된다.

다시 말해서 부모 객체의 참조변수로 자식 객체를 다룰 수 있다는 의미가 객체지향기법에서 다형성이다. 따라서 다형성은 클래스의 상속관계에서만 나올 수 있는 용어이다.

다음 [그림] 6-8은 '비행' 클래스를 하위 3개의 클래스로 상속하는 과정에서 부모 클래스의 '날다()' 메서드가 자식 클래스에서 날다(비행기), 날다(새), 날다(종이비행기) 형태로 재정의 된다. 결국 클래스 간의 상속관계를 통해서 메서드가 재정의 되고, 이를 토대로 클래스의 다형성이 구현된다.

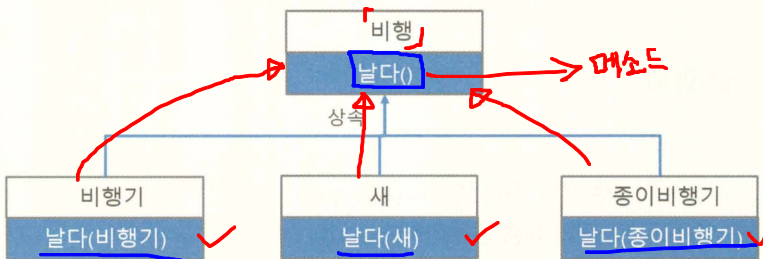


그림 6-8. 클래스의 다형성

✓: 부모클래스의 메소드가 자식클래스에서 재정의 된것

실습 다형성 예 170~171

chapter06.lecture.step07_polymorphism.py

Python Console

```
# (1) 부모 클래스
class Flight:
```

```
    # 부모 원형 함수
    def fly(self):
        print('날다, fly 원형 메서드')
```

날다, fly 원형 메서드

```
# (2) 자식 클래스 : 비행기
class Airplane(Flight):
```

```
    # 함수 재정의
    def fly(self):
        print('비행기가 날다.')
```

비행기가 날다.

```

# (2) 자식2클래스 : 새
class Bird(Flight) :

    # 함수 재정의
    def fly(self):
        print('새가 날다.')

# (2) 자식3클래스 : 종이비행기
class PaperAirplane(Flight) :

    # 함수 재정의
    def fly(self):
        print('종이 비행기가 날다.')

# (3) 객체 생성
# 부모 객체 = 자식 객체(자식1, 자식2)
flight = Flight() # 부모 클래스 객체
air = Airplane() # 자식1 클래스 객체
bird = Bird() # 자식2 클래스 객체
paper = PaperAirplane() # 자식3 클래스 객체

# (4) 다형성
flight.fly() # 날다, fly 원형 메서드

flight = air
flight.fly() # 비행기가 날다.

flight = bird
flight.fly() # 새가 날다.

flight = paper
flight.fly() # 종이 비행기가 날다.

```

해설 다형성 예

(1) 부모 클래스

Flight 클래스는 fly() 메서드를 갖는다.

(2) 자식 클래스 : 비행기

Airplane 클래스는 Flight 클래스를 상속받아서 만들어졌다. 따라서 부모 클래스의 fly() 메서드를 상속받는다. 그리고 상속 받은 메서드를 재정의 하였다.

(2) 자식 클래스 : 새

Bird 클래스는 Flight 클래스를 상속받아서 만들어졌다. 따라서 부모 클래스의 fly() 메서드를

상속받는다. 그리고 상속 받은 메서드를 재정의 하였다.

(2) 자식 클래스 : 종이비행기

PaperAirplane 클래스는 Flight 클래스를 상속받아서 만들어졌다. 따라서 부모 클래스의 fly() 메서드를 상속받는다. 그리고 상속 받은 메서드를 재정의 하였다.

(3) 객체 생성

부모 클래스와 자식 클래스 3개를 대상으로 각각 기본생성자를 이용하여 객체를 생성한다.

(4) 다형성

부모 객체의 참조변수인 flight에 자식 객체의 참조변수를 할당해 주면 부모 객체의 참조변수를 이용하여 해당 자식 객체의 멤버를 호출할 수 있다. 다음은 다형성을 위한 참조변수의 할당 예이다.

부모객체 참조변수 = 자식객체 참조변수

부모 클래스로부터 만들어진 객체의 주소가 저장된 참조변수를 -> 부모객체 참조변수
자식 클래스로부터 만들어진 객체의 주소가 저장된 참조변수를 -> 자식객체 참조변수

6.4 내장클래스

클래스는 함수와 마찬가지로 사용자가 직접 정의하는 사용자 클래스와 라이브러리 형식으로 제공되는 내장클래스로 분류된다. 파이썬에서는 풍부한 라이브러리를 통해서 수많은 함수와 클래스를 제공하고, 이를 이용하여 애플리케이션을 개발하거나 자료 분석에 활용한다.

‘제5장 모듈과 함수’ 편에서 공부한 내용 중 ‘모듈과 패키지에서 제공되는 함수를 이용하기 위해서는 먼저 import 명령어를 이용하여 해당 모듈이나 패키지를 작성중인 파일에 포함시켜야 한다.’라고 했던 것처럼 모듈이나 패키지는 함수뿐만 아니라 클래스도 제공한다.

파이썬에서 제공하는 내장클래스를 사용하기 위해서는 다음과 같은 형식으로 import 명령어를 이용하여 모듈을 포함시켜야한다. 형식1)은 해당 모듈에서 제공하는 모든 함수 또는 클래스를 가져온다. 형식2)는 해당 모듈에서 특정 클래스만 가져온다.

형식

- 1) import 모듈명
- 2) from 모듈명 import 클래스명1, 클래스명2,...

한편 builtins 모듈에서 제공하는 내장함수와 동일하게 내장클래스도 import 없이 사용할 수 있다.(참고 : <표> 5-1. builtins 모듈의 내장함수와 내장클래스)