

배우는 내용

1. 프로그래밍언어 개요
2. 구문 구조-언어가 제공하는 의미와 형식 개요
3. 변수-속성, 바인딩, 타입검사..
4. 타입-문자열, 배열, 포인터, 구조적 타입도
5. 제어구조-수식, assignment, 반복문
6. **부프로그램, argument** 구조적 언어
7. 객체, 추상데이터타입
8. 함수형 (functional) 언어
9. 논리적 (logical)언어 새로운 개념

ch12 객체지향언어

| 객체지향 프로그래밍 개념

객체 지향 개념(Object oriented concept)

→ 추상 데이터 타입과 비슷함.(캡슐화)

- 추상 데이터 타입 자체를 **클래스 (class)** 라 하고
- 클래스에 속하는 각 개체를 **객체 (object)** 라 하고,
해당 클래스에 대한 **사례 (instance)** 라 하고,
- 객체에 대한 연산 (부프로그램) 들을 **메소드 (method)** 라 함.
 - > 메소드는, 인터페이스를 나타내는 헤더 선언과 구현으로 나뉨

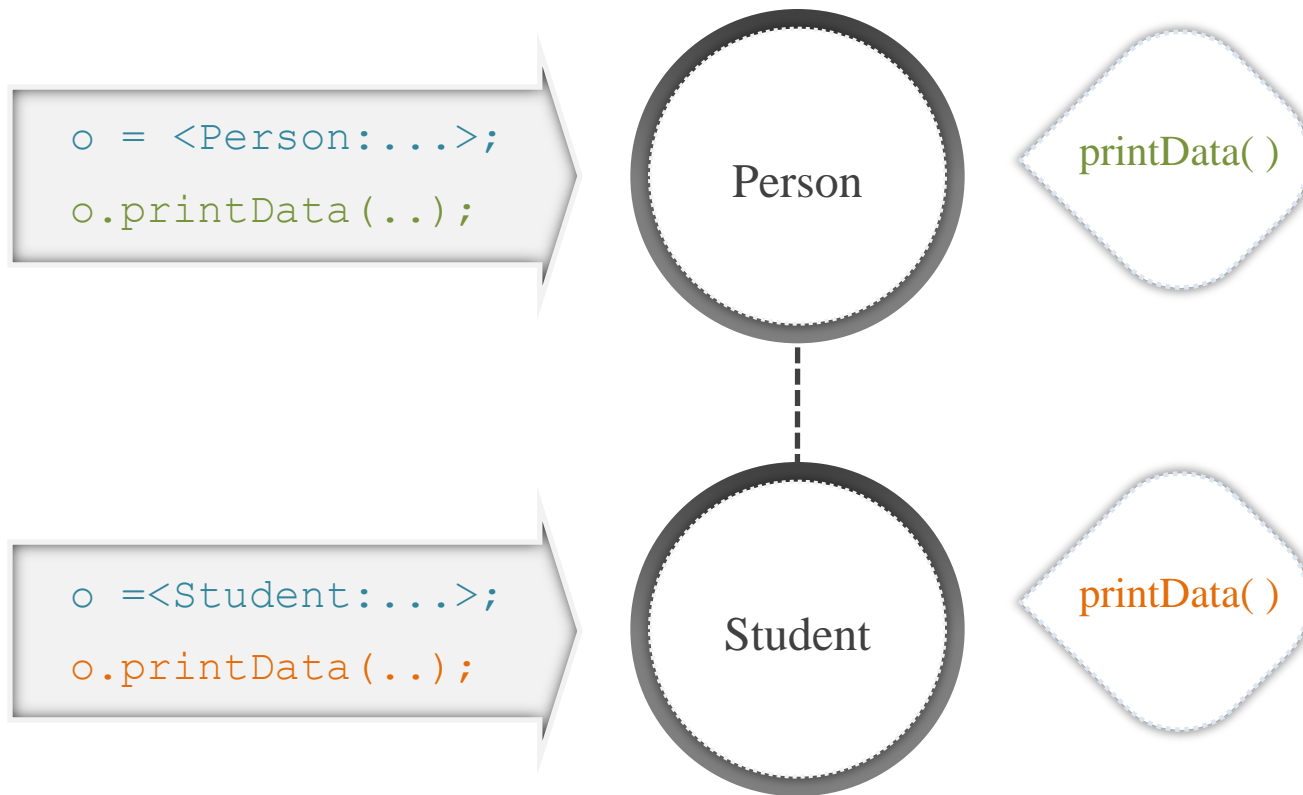
→ 추상 데이터 타입과 다름

- 상속 (inheritance)
- 다형성 (polymorphism)
- 기타

| 다형성 (Polymorphism)

다형성이란?

➔ 메소드 호출의 모양은 같아도, 다른 메소드가 선택될 수 있는 성질



ch13 함수형 프로그래밍 언어

함수형 언어

- 수학의 함수와 비교
- recursion
- Scheme 언어

(1)

```
(DEFINE (square x) (* x x))  
(square 7) → 49
```

(2)

```
(DEFINE ls '(a b c))  
(car ls) ==> a  
(cdr ls) ==> (b c)  
(car '((1) 2)) ==> (1)  
(cdr '((1) 2)) ==> (2)
```

(3)

```
(DEFINE (max3 n1 n2 n3)
  (if (> n1 n2)
    (if (> n1 n3)
      n1
      n3)
    (if (> n2 n3)
      n2
      n3)))
```

(4)

```
(third '(a b c d e)) => c
```

그러면..

```
(define (third ls)
  (car (cdr (cdr ls))))
```

(5)

```
(reverse3 '(a b c))    (c b a)
```

```
(DEFINE (reverse3 ls)
```

```
  (CONS (CADDR ls)      ; (3rd ls)
```

```
    (CONS (CADR ls)      ; (2nd ls)
```

```
      (CONS (CAR ls) '())))) ; (1st ls)
```


With Recursion

```
(6) (if (= n 0)
        0
        (+ n (sum (- n 1)))))
```

```
(7)
(define (length ls)
  (if (null? ls)
      0
      (+ 1 (length (cdr ls)))))
```

(8)

```
(define (member? item ls)
  (cond
    ((null? ls) #F)
    ((eq? item (car ls)) #T)
    (else (member? item (cdr ls)))))
```