

PL Assignment #8: Cute17 Project

과제물 부과일 : 2017-05-11 (목)

Program Upload 마감일 : 각 아이টে에 따라 다름

문제

그 동안 과제를 수행해 온 Cute 17문법에 따른 프로그램을 입력하면 결과를 출력하는 인터프리터를 설계 및 구현하라

Project 내용

아래와 같이 2개의 item으로 나뉜다. 아래 일정에 따라 수행하며, 각 아이টে 별로 보고서를 제출한다. 조를 구성한 경우 각 조원 간의 역할 분담도 명확하게 되어야 한다.

Item 1. 프로그램의 interpretation 환경 구현,

Item 2. 변수의 바인딩 처리: **define**문으로 변수로 정의하고 사용할 수 있도록 한다.

함수의 바인딩 처리: **define**문으로 함수를 정의하고 사용할 수 있도록 지원한다.

(함수가 함수의 인자로 넘어 갈 수 있어야 한다. - SSR방식)

주의)

- 변수는 **define** 으로 값을 지정한다
예) test case 1, 2 번 등 참고
- 이름없는 함수의 정의는
(**lambda** (*parameter_list*) (*optional_definition_of_local_vars*) (*function_body*))
와 같은 형식으로 구현한다. (*optional_definition_of_local_vars*) 는 없어도 된다
예) test case 8번 등 참고
- 함수가 이름을 가지게 되는 방법
define 으로 변수에 값을 지정하듯이 **define**으로 이름없는 함수를 지정할 수 있다.
예) test case 12 번 등 참고
- 함수는 이름을 통해 다른 함수의 인자로 전달 될 수 있다.
예) test case 18, 19번 등 참고
- 함수 내에서 함수가 정의될 수 있다.
lambda에서 **define** 을 활용해서 (*optional_definition_of_local_vars*) 내부에서 정의
예) test case 20 번 등 참고

점수 기입이 없는 것은 1점

나머지는 문제는 주석에 점수와 결과 같이 기입되어 있음.

Project TEST Case

1. (**define** a 1)
a ;; 1출력
2. (**define** b `(1 2 3))
b ;; `(1 2 3)
3. (**define** c (- 5 2))
c;; 3
4. (**define** d `(+ 2 3))
d; `(+ 2 3)

5. (define test b)
test;;` (1 2 3) (2점)
6. (+ a 3);;4
7. (define a 2) ;; 앞에 a가 정의된 후에 새로 정의
(* a 4);;8
8. ((lambda (x) (* x -2)) 3);;1
9. ((lambda (x) (/ x 2)) a) ;;1(2점)
10. ((lambda (x y) (* x y)) 3 5) ;;15(2점)
11. ((lambda (x y) (* x y)) a 5) ;;10(2점)
12. (define plus1 (lambda (x) (+ x 1))) ;; 전역 함수 구현
(plus1 3) ;; 4 (3점)
13. (define mul1 (lambda (x) (* x a))) ;; 전역 함수 구현, 전역 변수 포함
(mul1 a) ;; 4 (3점)
14. (define plus2
(lambda (x) (+ (plus1 x) 1))) ;;함수 내에서 전역 함수 호출 가능
(plus2 4) ;; 6 (3점)
15. (define plus3
(lambda (x) (+ (plus1 x) a))) ;;함수 내에서 전역 함수 호출 가능, 전역 변수 포함
(plus3 a) ;; 5 (3점)
16. (define mul2
(lambda (x) (* (plus1 x) -2)))
(mul2 7) ;; -16 (3점)
17. (define lastitem
(lambda (ls)
(cond ((null? (cdr ls)) (car ls))
(#T (lastitem (cdr ls)))))) ;;Recursion 구현 (6점)
18. (define square (lambda (x) (* x x)))
(define yourfunc (lambda (x func) (func x))
(yourfunc 3 square) ;;함수에서 함수를 인자로 사용가능 (4점)
19. (define square (lambda (x) (* x x)))
(define mul_two (lambda (x) (* 2 x)))
(define new_fun (lambda (fun1 fun2 x) (fun2 (fun1 x))))
(new_fun square mul_two 10) ;; (4점)
20. (define cube
(lambda (n) (define sqrt (lambda (n) (* n n)))
(* (sqrt n) n))) ;;Nested 함수 구현 (3점)
(sqrt 4) ;; 불리면 안됨(3점)