

프로그래밍의 원리 2012 가을

실습 10

OCaml 프로그래밍 연습, 예외 처리

서울대학교 프로그래밍 연구실

최민아

2012년 11월 21일

이번 실습의 목적은:

- OCaml 프로그래밍을 연습하며 함수형 언어와 타입 시스템에 익숙해진다.
- 예외 처리 방법을 익힌다.

1. 병합정렬(Mergesort)

Mergesort는 1945년에 폰 노이만이 고안한 $O(n \log n)$ 정렬 알고리즘입니다. Mergesort는 다음과 같이 동작합니다:

- (a) 정렬되지 않은 리스트를 길이가 반인 두 개의 리스트로 나눕니다.
- (b) 리스트의 길이가 1이 될 때까지 반으로 나누는 과정을 반복합니다. 길이가 1인 경우 리스트가 정렬되었으므로 해당 리스트를 돌려 줍니다.
- (c) 각각 정렬된 두 리스트를 정렬된 하나의 리스트로 합칩니다.

다음 `mergeSort` 함수에 쓰인 두 함수 `split`과 `merge`를 정의하세요.

```
let rec mergeSort lst =  
  match lst with
```

```

| [] -> []
| [a] -> [a]
| l ->
    let (m, n) = split l in
    let m' = mergeSort m in
    let n' = mergeSort n
    in
    merge m' n'

```

```

let _ = mergeSort [5 ; 4 ; 3 ; 7 ; 9]
- : int list = [3; 4; 5; 7; 9]

```

2. 예외 처리(Exception catch)

예외가 발생할 경우 특정 코드를 실행함으로써 예외를 처리할 수 있습니다. 다음 코드를 컴파일하고 b에 0을 입력해 보세요.

```

let s = try
string_of_int(
let a=read_int() in
let b=read_int() in
a/b)
with Division_by_zero -> "error"
let _ = print_string (s^"\n")

```

3. 계산기

exp 타입의 식을 받아 해당 식과 그 식을 계산한 값을 포함하는 문자열을 리턴하는 함수 calc을 만들어 보세요.

```
calc : exp → string
```

```

type exp = NUM of int
        | ADD of exp * exp
        | MUL of exp * exp
        | DIV of exp * exp

```

0으로 나누는 경우, `Division.by_zero` 예외를 처리해 “Div by zero”를 리턴하세요.

```
let _ = print_string (calc (ADD ((NUM 1), (MUL ((NUM 1), (NUM 4))))))  
(1 + (1 * 4)) = 5  
let _ = print_string (calc (DIV ((NUM 2), (NUM 0))))  
Div by zero
```