PL

Practice

Report



[04] map function

2021/03/29

최현석

201702083

|  |
| --- |
| ▶ 주어진 문제  - List와 tuple을 활용한 Map을 구현하시오. ( tuple = string \* string )  Add key value map : (key, value) 쌍을 map에 추가한다.  이미 존재하는 key라면 value를 업데이트  Find key map: map 에서 key를 찾아 value값을 반환, 없다면 에러메세지 출력  Erase key map : map 에서 key를 찾아 해당 tuple을 삭제, 없다면 에러메세지 출력  ▶ 해결 과정 – add key value map   1. 내부에 rec add\_rec 를 선언하여 map 을 mh 와 ml 로 나눈다. mh는 []로 시작   ↳   1. 그리고 ml 에 대해 패턴 매칭을 한다.      * + 원소가 존재 -> 해당 원소의 key가 입력받은 key와 같다면 taillist인 tl에 추가한다.   같지 않다면 (a::mh) tl 로 mh와 ml를 넣어 재귀 호출한다.   * + Empty -> map에 바로 추가해준다.  1. Lrec add\_rec 의 body가 in으로 끝나면 add\_rec key value [] map 을 호출한다.     ▶ 해결 과정 – find key map     * map 에 대해 패턴 매칭을 한다.   + 원소 존재 -> 원소의 key가 일치한다면 value를 반환 / 아니라면 key tl 에 대해 재귀 호출   + Empty -> “No such key exisits” 메세지 출력   ▶ 해결 과정 – erase key map     1. Add 함수와 마찬가지로 map 을 mh 과 ml 로 나누어 내부에 rec 함수를 선언한다. 2. Ml에 대해 패턴 매칭을 한다.    * 원소 존재 -> 원소의 key가 같다면 원소를 제외한 mh 와 tl을 @ 연산을 통해 합친다.   아니라면 a::mh tl 에 대해 재귀 호출한다.   * + Empty -> “No such key exists” 메세지를 출력  1. Let – in 이 끝나면 mh를 비어있는 리스트 [] 로 넣고 호출한다.   ↳ erase\_rec key [] map  ▶ 결과( test case )     * + (String\*string) list 출력하는 함수를 만들어 test case 출력       ▶ 느낀 점   * add 와 erase 함수를 map을 그대로 사용해서 재귀 호출하여 a :: (add key value tl) 과 같이 non-tail-rec 하게 만들었는데 tail-rec 하게 구현하는 것이 좋은 방법이라 하여 방법을 찾던 중 a :: <- 이 부분을 함수의 매개변수로 넣으면 되겠다는 생각을 했다. 그래서 내부에 새로운 함수를 선언하여 tail – rec 하게 구현했다. 앞으로 non-tail-rec한 코드를 tail-rec한 코드로 변환하고 싶을 때 새로운 매개변수를 갖는 함수를 만드는 게 좋은 방법이 될 것 같다. * 과제 해결 시간 : 실습 시간 내 |

‘’’’’’’’’’’’’