

Printed: Mercoledì, 20 maggio 2015 17:54:38

```
(* ESERCIZIO: SOLUZIONE *)

exception Undefined;;
exception AlreadyDefined ;;

module type RELAZIONE =
  sig type ('a,'b) relazione
    val relazioneC: unit -> ('a,'b) relazione
    val isIn1: ('a,'b) relazione -> 'a -> bool
    val getAll2: ('a,'b) relazione -> 'a -> 'b list
    val add: ('a,'b) relazione -> 'a -> 'b -> ('a,'b) relazione
  end;;

module Relazione =
  (struct
    type ('a,'b) relazione = ('a * 'b) list
    let relazioneC = fun () -> []
    let rec isIn1 r x = match r with
      | [] -> false
      |(u,v)::rs -> (u=x) || isIn1 rs x
    let rec getAll2 r x = match r with
      | [] -> []
      |(u,v)::rs -> let s = (getAll2 rs x)
        in if (u=x) then v::s else s
    let add r x y = (x,y)::r
  end:RELAZIONE);;

(* Esercizio 3.
Siano RELAZIONE un API per relazioni con le operazioni definite come sopra,
e Relazione un ADT che la implementa. Si usino tali moduli per definire un
ADT FunRel per relazioni che sono funzioni finite su generici tipi ed aventi,
in aggiunta alle operazioni di RELAZIONE, l'operazione apply che applicata
ad un valore del dominio della funzione restituisce l'immagine della funzione.
L'operazione apply solleva eccezione Undefined se applicata su valori su cui
è indefinita.
L'operazione add solleva eccezione AlreadyDefined se si tenta di cambiare la
immagine di un valore definito.
soluzione del 19/5/2015
*)

module type FUNREL =
  sig type ('a,'b) funrel
    val relazioneC: unit -> ('a,'b) funrel
    val isIn1: ('a,'b)funrel -> 'a -> bool
    val getAll2: ('a,'b) funrel -> 'a -> 'b list
    val add: ('a,'b) funrel -> 'a -> 'b -> ('a,'b) funrel
    val apply: ('a,'b) funrel -> 'a -> 'b
  end;;

module FunRel =
  (struct
    include Relazione
    type ('a,'b) funrel = ('a,'b) relazione
    let relazioneC = fun () -> Relazione.relazioneC()
    let isIn1 r x = Relazione.isIn1 r x
    let getAll2 r x = Relazione.getAll2 r x
    let add r u v =
      match (Relazione.getAll2 r u) with
      | [] -> Relazione.add r u v
  end)

```

Printed: Mercoledì, 20 maggio 2015 17:54:38

```
        |w::l -> if (v!=w)
                then raise AlreadyDefined
                else r
let apply r u =
    match (Relazione.getAll2 r u) with
    |[] -> raise Undefined
    |v::l -> v
end:FUNREL);;

(* Esecizio 5.
Utilizzando ('a,'b) relazione (definita nel modulo Relazione, nei giorni scorsi) e
('a,'b) sizedRel (definita in esercizio 3), mostrare un programma che:

1- definisce una funzione ff = ("A",1),("C",3). Quindi,
2- applica ad essa le seguenti operazioni:
    FunRel.apply ff "C"
    FunRel.add ff "A" 5
soluzione del 19/5/2015

let ff = FunRel.relazioneC();;
let ff = ...;;
*)
```