

Printed: Mercoledì, 20 maggio 2015 17:55:46

```
(* ESERCIZIO: SOLUZIONE *)

exception Undefined;;

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.
Sia RELAZIONE un API per relazioni con le operazioni definte come sopra,
e Relazione un ADT che la implementa. Li si utilizzi ("si estenda RELAZIONE")
per definire un ADT per valori che hanno in aggiunta alle operazioni di RELAZIONE,
anche l'operazione size che fornisce il numero di coppie della relazione.

soluzione del 18/5/2015
module type RELAZIONES =
  sig type ('a,'b) sizedRel
    val relazioneC: unit -> ('a,'b) sizedRel
    val isIn1: ('a,'b)sizedRel -> 'a -> bool
  end;;

module RelazioneS =
  (struct
    include RELAZIONE
    type ('a,'b) sizedRel = int * ('a,'b) relazione
    let relazioneC () = (0,Relazione.relazioneC())
    let isIn1 (n,r) x = Relazione.isIn1 r x
  end:RELAZIONES);;
*)

(* completare l'esercizio, compilare e provvedere a short tests per l'esecuzione
soluzione del 19/5/2015*)

module type RELAZIONES =
  sig type ('a,'b) sizedRel
    val relazioneC: unit -> ('a,'b) sizedRel
    val isIn1: ('a,'b)sizedRel -> 'a -> bool
    val getAll2: ('a,'b) sizedRel -> 'a -> 'b list
    val add: ('a,'b) sizedRel -> 'a -> 'b -> ('a,'b) sizedRel
    val size: ('a,'b) sizedRel -> int
  end;;
```

Printed: Mercoledì, 20 maggio 2015 17:55:46

```

module RelazioneS =
  (struct
    include Relazione
    type ('a,'b) sizedRel = int * ('a,'b) relazione
    let relazioneC = fun () -> (0,Relazione.relazioneC())
    let isIn1 (n,r) x = Relazione.isIn1 r x
    let getAll2 (n,r) x = Relazione.getAll2 r x
    let add (n,r) u v =
      let listIn x l = List.fold_right(||)(List.map(fun y->y=x) l) false
      in if listIn v (Relazione.getAll2 r u)
         then (n,r)
         else (n+1, Relazione.add r u v)
    let size (n,r) = n
  end:RELAZIONES);;

```

(\* Esercizio 4.

Discutere i limiti di questo approccio all'estensione dei tipi astratti.

In particolare si dica:

- perchè anche in questo caso si può parlare di riuso del software e quale codice risulta riusato
- perchè questa definizione è diversa dalla tipica estensione nei linguaggi Object Oriented e più precisamente, quale codice che in Java è riusabile in questo approccio non è riusabile.

Suggerimento. Aiutarsi illustrando un ipotetico contesto di uso definito da un programma P che introduce RELAZIONE, definisce il modulo Relazione e la usa diffusamente in una sezione Q del programma. Il programma P è quindi aggiornato con una nuova sezione H che introduce RELAZIONES e il modulo RelazioneS.

Si usino P, Q, H o altre sotto-sezioni, opportunamente introdotte, per rispondere ai punti (a) e (b) sopra.

\*)

(\* Esercizio 5.

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

- definisce una relazione rr = ("A",1),("C",3),("A",5). Quindi,
- applica ad essa le seguenti operazioni:

```

  Relazione.isIn1 rr "C"
  RelazioneS.size rr
  RelazioneS.getAll2 rr "A"

```

soluzione del 19/5/2015\*)

```

let rr = Relazione.relazioneC();;
let rr = Relazione.add rr "A" 1;;
let rr = Relazione.add rr "C" 3;;
let rr = Relazione.add rr "A" 5;;
(* definizione di rr di tipo (string,int) relazione *)
Relazione.isIn1 rr "C";;
(* dobbiamo creare una nuova relazione di tipo (string,int) sizedRel *)
let rr = RelazioneS.relazioneC();;
let rr = RelazioneS.add rr "A" 1;;
let rr = RelazioneS.add rr "C" 3;;
let rr = RelazioneS.add rr "A" 5;;
RelazioneS.size rr;;
RelazioneS.getAll2 rr "A";;

```