

Printed: Monday, May 18, 2015 5:10:16 PM

```
(* 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 ...
    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.
*)
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 *)

(* Esecizio 4.
Discutere i limiti di questo approccio all'estensione dei tipi astratti.
In particolare si dica:
a. perchè anche in questo caso si può parlare di riuso del software e quale
   codice risulta riusato
b. perchè questa definizione è diversa dalla tipica estensione nei linguaggi
   Objcet 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.
*)

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

Printed: Monday, May 18, 2015 5:10:16 PM

- 1- definise una relazione $rr = ("A",1),("C",3),("A",5)$. Quindi,
 - 2- applica ad essa le seguenti operazioni:
 - Relazione.isIn1 rr "C"
 - RelazioneS.size rr
 - RelazioneSS.getAll2 rr "A"
- *)