

Printed: Martedì, 12 maggio 2015 18:12:24

```

/*
Fornire una definizione di relazione come in relazioneADT ma per relazioni modificabili.
*/

import java.io.*;
import java.util.*;

interface RelazioneAPI <T1,T2>{ //One Java API for the type Relazione mutable
    public boolean isIn1 (T1 e);
    // public boolean isIn2 (T2 e);
    public Vector<T2> getAll2(T1 e);
    // public Vector<T1> getAll1(T2 e);
    public void add(T1 x, T2 y);
}

class RelazioneADT<T1,T2> implements RelazioneAPI<T1,T2>{
    private Vector<T1> left;
    private Vector<T2> right;

    public RelazioneADT(){
        left = new Vector<T1>();
        right = new Vector<T2>();
    }
    private boolean isEmpty(){
        return left.size()==0;
    }
    public boolean isIn1(T1 e){
        return (isEmpty() || left.contains(e));
    }
    public Vector<T2> getAll2(T1 e){
        Vector<T2> r = new Vector<T2>();
        for (int i=0; i<left.size(); i++){
            if (left.get(i).equals(e)) r.add(right.get(i));
        }
        return r;
    }
    public void add(T1 e1, T2 e2){
        left.add(e1); right.add(e2);
    }
    class pair{
        T1 l; T2 r;
    }
}

/*
usare vettori di coppie
*/

class RelazioneADTP<T1,T2> implements RelazioneAPI<T1,T2>{
    private Vector<pair> R;
    ....
}

/*
1) Estendere RelazioneADT in un ADT contenente anche l'operazione size che calcola il nume
coppie (anche duplicate) inserite;
*/
/*
2) Estendere RelazioneADT in un ADT contenente anche l'operazione size che calcola il nume

```

Printed: Martedì, 12 maggio 2015 18:12:24

```
coppie differenti contenute nella relazione;
-- facile
3) Estendere RelazioneADT in un ADT contenente anche l'operazione dom1 che calcola un Vect
che fornisce la proiezione, sul primo componente, della relazione.
4) Estendere RelazioneADT (completa delle operazioni isIn2 e getAll1) in un ADT contenente
un metodo booleano che calcola true se this è una relazione simmetrica.
5) Estendere RelazioneADT in un ADT per relazioni simmetriche.
*/
```

```
class Main{
    public static void main(String[] args){
        Integer n1 = new Integer(4);
        Integer n2 = new Integer(5);
        Integer n3 = new Integer(7);
        String s1 = new String("abba");
        String s2 = new String("Rolling");
        String s3 = new String("abba");
        System.out.println(s1==s3);
        System.out.println(s1.equals(s3));
        RelazioneAPI<String,Integer> r1 = new RelazioneADT<String,Integer>();
        r1.add(s1,n3);
        RelazioneAPI<String,Integer> r2 = r1;
        System.out.println(r1==r2);
        System.out.println(r1.equals(r2));
        r1.add(s1,n2);
        r1.add(s2,n1);
        System.out.println("il numero di s1 in r1 è: " +r1.getAll2(s1).size());
        r1.add(s2,n1);
        System.out.println(r1==r2);
        System.out.println(r1.equals(r2));
        //short test for RelazioneADTP -----
        ....
    }
}
```