

```
/*
  Uso di struttura condivisa per supportare valori a componenti non modificabili, e a
  struttura dinamica. Tipico modo di realizzare i valori nei Linguaggi Funzionali.
  CONSIGLIATO: Semplice ed efficiente, evita di copiare strutture.
*/

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

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

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

    public RelazioneADT(){
    }
    public RelazioneADT(T1 x, T2 y, RelazioneADT<T1,T2> r){
        left = x; right = y; rem = r;
    }
    public boolean isIn1 (T1 e){
        return (left != null && (left.equals(e)||rem.isIn1(e)));
    }
    public boolean isIn2 (T2 e){
        return (right != null && (right.equals(e)||rem.isIn2(e)));
    }
    public Vector<T2> getAll2(T1 e){
        if (left == null) return new Vector<T2>();
        Vector<T2> r = rem.getAll2(e);
        if (left.equals(e)) r.add(right);
        return r;
    }
    public Vector<T1> getAll1(T2 e){
        if (left == null) return new Vector<T1>();
        Vector<T1> r = rem.getAll1(e);
        if (right.equals(e)) r.add(left);
        return r;
    }
    public RelazioneAPI<T1,T2> add(T1 x, T2 y){
        return new RelazioneADT<T1,T2>(x,y,this);
    }
}

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("Stones");
        System.out.println(s1 == s3);
    }
}
```

```
System.out.println(s1.equals(s3));
RelazioneAPI<String,Integer> r1 = new RelazioneADT<String,Integer>().add(s1,n3);
RelazioneAPI<String,Integer> r2 = r1.add(s1,n2);
r1 = r2.add(s2,n1);
System.out.println("il numero di s1 in r1 è: " +r1.getAll2(s1).size());
    }
}
```