

Metodologie di Programmazione

8/1 /2008

1) [12 punti] Si mostri l'evoluzione dello stato (ambiente di classi, heap, pila) durante la valutazione delle seguenti dichiarazioni di classe e durante l'esecuzione del metodo main:

```
public class Sopra extends Object {
    public static int x;
    public int y;
    public static void set (int i) {x=i+1;}
    public Sopra metodo (int i)
    {set(i); Sopra p1=new Sopra();p1.y=y+i;
        return p1; }
    {y=x;} }

public class Sotto extends Sopra {
    public int z;
    public Sopra metodo (int i)
    {set(i); Sopra p1=new Sopra(); y=p1.y+i;
        z=y+x+i; return p1;}
    {z = x+1;} }

public class Princ extends Object {
    public static void main (String[] args) {
        Sopra p1 = new Sotto();
        Sopra p2 = p1.metodo(3);
        Sopra p3 = p2.metodo(1);    }
}
```

2) Si consideri una classe astratta RelCO che definisce relazioni tra oggetti omogenei comparabili, con due implementazioni EmptyRelCO per la relazione vuota e

FullRelCO per la relazione non vuota:

```
public abstract class RelCO {
    // OVERVIEW: una RelCO è una relazione tra oggetti omogenei
    // comparabili. Il tipo degli elementi è fissato dalla prima
    // coppia inserita. E' non modificabile.
    // metodi astratti
public abstract RelCO insert (Comparable x, Comparable y) throws
    NullPointerException, ClassCastException, DuplicateException;
    // EFFECTS: se x oppure y è nullo solleva NullPointerException,
    // se c'è incompatibilità di tipo con le coppie già presenti
    //nella relazione solleva ClassCastExcpetion, se (x,y) appartiene
    //gia' alla relazione solleva DuplicatException, altrimenti
    //restituisce una nuova relazione che contiene la coppia (x,y)
public abstract boolean isIn (Comparable x, Comparable y);
    // EFFECTS: restituisce true se la coppia (x,y) appartiene a this,
    // altrimenti restituisce false
public abstract Iterator elements ();
    // EFFECTS: restituisce un generatore, che produce le coppie della
    //relazione (come oggetti di tipo Pair) ordinate rispetto alla
    //prima componente
}
```

Per ciascuna delle due sottoclassi EmptyRelCO e FullRelCO

- (a) [4 punti] si fornisca la specifica (verificando che soddisfi il principio di sostituzione)
- (b) [6 punti] si diano la rappresentazione, la funzione di astrazione e l'invariante
- (c) [8 punti] si implementino costruttori e metodi. Inoltre si provi che il metodo insert di FullRelCO soddisfa invariante e specifica