

ESERCIZI SU GESTIONE DELLA MEMORIA

MAGGIO 2022

Esercizio 1

Si consideri il seguente frammento in un linguaggio con garbage collection mediante contatori dei riferimenti (le variabili adottano un modello a riferimento):

```
class Elem {
    int n;
    Elem next;

    Elem foo() {
        Elem p = new Elem(); // oggetto OGG1
        p.next = new Elem(); // oggetto OGG2
        Elem q = new Elem(); // oggetto OGG3
        q.next = p.next;
        return p.next;
    }
}
Elem r = Elem.foo();
```

Si dica quali sono i valori dei contatori dei riferimenti dei tre oggetti al termine dell'esecuzione del frammento. Spiegare brevemente (max 10 righe) il ragionamento dietro alla risposta.

Esercizio 2

Si consideri un linguaggio con modello di variabili a riferimento, garbage collection e polimorfismo parametrico, dove `List< T >` indica un tipo polimorfo “lista di T” che supporta le seguenti operazioni:

`new: Unit -> List< T >`, metodo statico, crea una nuova lista vuota;
`get: int -> T`, restituisce l’elemento della lista su cui è chiamata l’operazione alla posizione data in input;
`set: int, T -> Unit`, sostituisce l’elemento presente nella lista su cui è chiamata l’operazione alla posizione data in input con l’elemento `T` dato in input. Se non esiste già un elemento alla posizione data in input, la lista resta invariata;
`add: T -> List< T >` aggiunge in coda alla lista l’elemento dato in input e restituisce la lista aggiornata.

Esempio:

```
List< Int > l = List< Int >.new();
l = l.add( 10 ).add( 20 ).add( 30 );
print( l.get( 0 ) + l.get( 1 ) ); // stampa 30
l.set( 2, 50 );
print( l ); // stampa [ 10, 20, 50 ]
```

Dove `print`, applicato ad una lista, ne stampa il contenuto tra parentesi quadre, con gli elementi separati da virgole.

Domanda 1: cosa stampa il seguente frammento nel quale il passaggio dei parametri avviene per riferimento? Spiegare brevemente (max 10 righe) il ragionamento dietro alla risposta.

```
List< Int > l = List< Int >.new();           // OGG1
List< Int > h = List< Int >.new();           // OGG2
l.add( 0 ).add( 1 ).add( 2 ).add( 3 );
h.add( 100 ).add( 200 );
void f( List< Int > x ){
    List< Int > w = List< Int >.new();       // OGG3
    w.add( 10 ).add( 20 );
    x.set( 3, 40 );
    h = w;
    w.set( 0, 30 );
}
f( l );
```

```
print( l );  
print( h );
```

Domanda 2: si supponga il garbage collector utilizzi contatori dei riferimenti. Considerando il codice di Domanda 1, quali sono i valori dei contatori dei riferimenti di OGG1, OGG2 e OGG3 (indicati dai commenti) fino a prima dell'istruzione `f(l);`? Spiegare brevemente (max 10 righe) il ragionamento dietro alla risposta.

Esercizio 3

Si assuma di avere un linguaggio che adotti la tecnica del reference counting. Dato **OGG**, generico oggetto nello heap, indichiamo con **OGG.ref-c** la sua rappresentazione interna del reference count accessibile solo al garbage collector. Si consideri il seguente frammento di codice:

```
C bar = new C(); // oggetto OGG2
{
  C foo = new C(); // oggetto OGG1
  C fie = foo;
  bar = fie;
}
```

Si dia il valore di **OGG1.ref-c** e **OGG2.ref-c** dopo l'esecuzione del frammento. Quali di questi oggetti possono essere restituiti alla lista libera? Spiegare brevemente (max 10 righe) il ragionamento dietro alla risposta.

Esercizio 4

Si scriva un breve frammento di codice, nel linguaggio di programmazione preferito, che generi una dangling reference, spiegando brevemente (max 10 righe) i passaggi dell'esecuzione.

Esercizio 5

Si assuma di avere uno linguaggio che adotti la tecnica del reference counting. Dato **OGG**, generico oggetto nello heap, indichiamo con **OGG.ref-c** la sua rappresentazione interna del reference count accessibile solo al garbage collector. Si consideri il seguente frammento di codice:

```
class C {
  int n;
  C next;
}
C foo (){
  C p = new C();    // oggetto OGG1
  p.next = new C(); // oggetto OGG2
  C q = new C();    // oggetto OGG3
  q.next = p.next;
  return p.next;
}
C r = foo();
```

Si dica quali sono i valori dei contatori dei riferimenti dei tre oggetti eseguendo l'esecuzione del frammento. Spiegare brevemente (max 10 righe) i passaggi dell'esecuzione.