PROGRAMMAZIONE II (A,B) - a.a. 2019-20
Primo Appello – 17 Gennaio 2020

Domande di base

1. Si consideri il seguente programma (scritto con una sintassi Java-like)

```
class A  {public void m1() {...};}
class B extends A {public void m2() {...};}
class C extends B {public void m1() {...};}

\\ main
A a = new B();
A b = new C();
b=a;
b.m1();
b.m2();
```

   Si dica se il `main` supera il controllo statico dei tipi. Motivare la risposta.

2. Si dica se nel caso di *blocchi in-line* le regole di *scope* statico e dinamico si comportano nello stesso modo. Motivare la risposta.

3. Si descriva il ruolo del *contatore dei referimenti* nella tecnica di garbage collection detta *reference counting*. Questa tecnica consente di determinare come garbage strutture dati con cicli interni? Motivare la risposta.

Esercizio 1

Si consideri un tipo di dato astratto *GenericContainer* introdotto per rappresentare una collezione di oggetti generici. Gli elementi della collezione sono disposti in ordine qualunque e sono di molteplicità qualunque (sono ammesse ripetizioni). Elementi di valore null non devono fare parte della collezione. L'interfaccia generica `GenericContainer <T>` include, tra gli altri, i seguenti metodi

- `public void insert(T elem)` *il cui effetto è quello di inserire l' elemento generico* `elem` *nella collezione*
- `public int delete(T elem)` *il cui effetto è quello di rimuovere una occorrenza dell'elemento* `elem` *dalla collezione e di restituire il numero degli elementi uguali a* `elem` *ancora presenti nella collezione*
- `public int delOccurrences(T elem, int n)` *il cui effetto è quello di rimuovere un numero* `m` *di occorrenze dell'elemento* `elem` *dalla collezione, dove* $m \leq n$*, e di restituire il numero degli elementi uguali a* `elem` *effettivamente rimossi (ovvero* `m`*)*
- `public int size ()` *restituisce il numero degli elementi presenti nella collezione*
- `public int getOccurrences(T elem)` *il cui effetto è quello di restituire il numero delle occorrenze dell' elemento* `elem` *nella collezione*
- `public T minOccurrences()` *il cui effetto è quello di restituire l' elemento della collezione avente il minor numero di occorrenze*

1. Si definisca la specifica dell'interfaccia `GenericContainer<T>`, indicando l'elemento tipico e per ogni metodo le clausole REQUIRES, MODIFY ed EFFECTS, il valore restituito e le eventuali eccezioni lanciate in dipendenza dei parametri attuali.

2. Si assuma di implementare la classe `MyGenericContainer<T>` che implementa `GenericContainer<T>` con le seguenti variabili d'istanza:

```
private ArrayList<Pair<T>> elts;
private static class Pair<T> {
    //Overview: Tipo immutabile rappresentante una coppia (T,int)
          T t;
          int i;
          Pair(T y, int j) {t = y; i = j;}
          boolean contains(T x) { return (x.equals(t));}
          int occurrences( ) { return i;}
       }
```

   (a) Si definiscano la funzione di astrazione e l'invariante di rappresentazione.

   (b) Si implementino il costruttore ed i metodi `insert` e `delOccurrences` e si dimostri che l'implementazione proposta preserva l'invariante di rappresentazione.

3. Si consideri la classe `OrderedGenericContainer<T>` in cui gli elementi della collezione sono ordinati in modo decrescente rispetto al numero di occorrenze. La classe `OrderedGenericContainer<T>` può essere sottotipo di `MyGenericContainer<T>` secondo il principio di sostituzione di Liskov? Motivare le proprie risposte.

Esercizio 2

Si consideri il seguente programma OCAML

```
let equalfun f g =
  let rec equalfrom x = f  x = g  x && equalfrom (x+1)
     in  equalfrom 1;;
let mult_sum (x, y) =
    let z = x + y in
       fun w -> w * z;;
let z = 4;;
let mf2 x = x*z;;
let f = mult_sum (3, 4);;
equalfun f mf2;;
```

Si simuli la valutazione del programma mostrando la struttura della pila dei record di attivazione.

Esercizio 3

Estendere il linguaggio didattico funzionale con una nuova forma di astrazione funzionale `both-fun` che consente di applicare una coppia di funzioni non ricorsive ad un valore producendo come risultato una coppia di valori. Per esempio in sintassi concreta l'astrazione `both-fun(fun x= x+1, fun x=x*2)` applicata al valore `5` produce come risultato la coppia `<6,10>`.

- Definire le regole di valutazione dell'astrazione `both-fun` e estendere consistentemente la struttura dell'interprete del linguaggio didattico funzionale.