15 dicembre 2003 Durata: ore 2.30

Linguaggi di Programmazione

Nome e Cognome	
Matricola	
Anno di corso	
Telefono	

1. Specificare la grammatica EBNF di un linguaggio in cui ogni frase corrisponde ad una o più dichiarazioni di classi. Ogni dichiarazione di classe è costituita da una intestazione (comprendente il nome della classe), dalla dichiarazione delle variabili di istanza (almeno una), dalla dichiarazione dei metodi (almeno uno), e da una coda (comprendente il nome della classe). Ecco un esempio di frase che definisce due classi `Alfa` e `Beta`:

```
class Alfa is
  variables
    a, b, c: integer;
    d, e: real;
    f: record (x: integer, y: record (v: string, w: real));
  methods
    gamma(a: string, d: real), delta(f: string): real;
    epsilon(c: integer): integer;
    omicron(d: real), tau(r: integer): record(a: string, b: real);
end Alfa;

class Beta is
  variables
    m, n: integer;
    r1, r2: record (x: integer, y: string);
  methods
    zeta(a: string, d: real), sigma(f: string): string;
    omega(c: integer): integer;
end Beta;
```

Le variabili (introdotte dalla keyword **`variables`**) sono definite da sequenze di identificatori terminate dal tipo corrispondente. Possibili tipi sono **`integer`**, **`real`**, **`string`** e **`record`**. I primi tre tipi sono atomici. Il tipo **`record`** definisce una struttura (racchiusa tra parentesi tonde) i cui campi (almeno uno) sono a loro volta caratterizzati da identificatori e relativi tipi. I tipi sono ortogonali tra loro. La dichiarazione dei metodi (introdotti dalla keyword **`methods`**) è simile a quella delle variabili, con l'aggiunta della dichiarazione dei parametri formali in ingresso (almeno uno) .

2. Specificare la semantica operazionale del seguente assegnamento in base alle seguenti assunzioni:
   - L'espressione e' valutata in corto circuito, da destra a sinistra.
   - Il linguaggio di specifica non comprende gli operatori di incremento e decremento (++, --).

```
v = ((a-- / 2) > c and a+b != c+2) or (a == ++b)
```

3. Data la seguente tabella di operatori:

Operatori	Associatività
^	destra
**	destra
*, /	sinistra
+, - (unari)	destra
+, - (binari)	sinistra

riscrivere la seguente espressione in forma parentetica:

```
-a+b*c/3*e**4**x^y^z--s^w
```

4. Definire nel linguaggio funzionale *Scheme* la funzione **`natoms`**, avente in ingresso una lista **`L`**, che computa il numero di elementi atomici (non liste) direttamente o indirettamente inclusi in **`L`**. Ecco alcuni esempi:

`L`	**`natoms`**
`()`	0
`(()()(A))`	1
`(A (B C))`	3
`(A (B C) (D ()))`	4

5. Definire e discutere i concetti di *scope statico* e *scope dinamico*, mettendone in evidenza vantaggi e svantaggi.

6. Definire le regole che garantiscono che una sottoclasse sia anche un sottotipo, illustrandone quindi le ragioni mediante un semplice esempio.