

Corso di Linguaggi di Programmazione

A.A. 2008/2009

C. Braghin e A. Ceselli

19 Febbraio 2009

Cognome e Nome:

Numero matricola:

Domanda 1. Si discutano brevemente i seguenti punti:

1. Parse tree (albero di derivazione): di cosa si tratta? Cosa rappresenta? A cosa serve?
2. Passaggio dei parametri nelle funzioni: quali possibilità ci sono? Quali sono le differenze, anche in termini di efficienza?
3. Scoping statico vs scoping dinamico.
4. Errori di tipo: cosa sono? Vengono sempre segnalati? Quando e da chi? Che "tipo" di errori generano?

Domanda 2. Discutere il concetto di Abstract Data Type (ADT). Indicare ed analizzare le finalità e le caratteristiche principali di un ADT. Presentare poi il concetto di polimorfismo parametrico, con particolare riferimento alla realizzazione di Abstract Data Types (ADT) parametrici, evidenziandone gli obiettivi principali. Riportare un esempio di ADT, estenderlo alla sua versione parametrica, e discuterne l'implementazione in almeno un linguaggio di programmazione a scelta.

Domanda 3. Scrivere in Scheme una funzione `contaParentesi` che prenda come argomento una lista qualsiasi (cioè una lista che contiene elementi di qualunque tipo, anche liste) e che restituisca il numero di liste (anche annidate - vedi ultimi 2 esempi) contenute dalla lista data in input.

La funzione `contaParentesi` si dovrebbe comportare nel modo seguente:

```
> (contaParentesi '(1 2 3))
0
> (contaParentesi '())
0
> (contaParentesi '(a (1 2 3) b))
1
> (contaParentesi '(a (1 2 (e f g)) b))
2
> (contaParentesi '(a (1 2 ) (e f g) b))
2
```

Domanda 4.

A) Realizzare un programma Prolog che, ricevuta in ingresso una lista, controlli che la lista sia composta da sole copie di un particolare elemento. Ad esempio:

```
?- allequal([a, a, a, a]).
```

Yes

```
?- allequal([a, a, b, a]).
```

No

```
?- allequal([a, b, c, d]).
```

No

Che risultato si può attendere utilizzando il predicato nel seguente modo:

```
?- allequal(X).
```

Perchè ?

B) Realizzare un programma Prolog che, ricevuta in ingresso una lista, controlli che nella lista *non* compaiano duplicati. Ad esempio:

```
?- alldifferent([a, b, c, d]).
```

Yes

```
?- alldifferent([a, b, a, d]).
```

No

```
?- alldifferent([a, b, c, b]).
```

No

```
?-
```

Che risultato si può attendere utilizzando il predicato nel seguente modo:

```
?- alldifferent(X).
```

Perchè ?