

# Corso di Linguaggi di Programmazione

## SESTO APPELLO A.A. 2006/2007

C. Braghin e A. Ceselli

12 febbraio 2008

Cognome e Nome: .....

Numero matricola: .....

**Domanda 1.** Si discuta brevemente il meccanismo di gestione esplicita delle *eccezioni* a livello di linguaggio presente nei moderni linguaggi di programmazione. In particolare, si considerino brevemente almeno i seguenti punti, con esempi in Java o in un linguaggio a piacere: (i) Definizione di eccezione, (ii) Scelte nella definizione della gestione delle eccezioni: chi le gestisce, chi le deve gestire, dove e come devono venire specificati i gestori di eccezioni, cosa succede se si solleva un'eccezione che non viene gestita da nessuna parte del codice (e.g., si tratta di un errore di compilazione o run-time)? (iii) Errori sintattici e/o lessicali possono venire considerati eccezioni? E gli errori di tipo? Giustificare la risposta. (iv) Esempio di eccezione definita dall'utente e suo utilizzo.

**Domanda 2.** Discutere il concetto di programmazione orientata agli oggetti. Evidenziare le differenze tra l'approccio procedurale e l'approccio orientato agli oggetti nella realizzazione di programmi. Descrivere le features richieste ad un linguaggio di programmazione per supportare la programmazione orientata agli oggetti. Confrontare il concetto di sottoclasse con il concetto di sottotipo. Descrivere e confrontare, anche attraverso esempi, il tipo di supporto alla programmazione orientata agli oggetti in almeno due linguaggi di programmazione.

**Domanda 3.** Scrivere in Scheme una funzione `riproduci` che prenda come argomento una lista `list` di interi e che:

- verifichi che gli interi siano tutti nell'intervallo da 1 a 10 (estremi compresi);
- ritorni come risultato una lista dove l'occorrenza di ciascun elemento della lista `list` di input è ripetuta un numero di volte pari al suo valore numerico.

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

```
> (riproduci '(1 2 3))
(1 2 2 3 3 3)
> (riproduci '(5))
(5 5 5 5 5)
> (riproduci '())
()
```

```
> (riproduci '(11))  
#f
```

**Domanda 4.** Realizzare un programma prolog *palindrome(X)* che verifichi se una stringa X è palindroma o meno. Esempi:

```
?- palindrome([a]).
```

Yes

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

Yes

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

Yes

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

No

Si discuta, inoltre, l'esito di una query del seguente tipo:

```
?- palindrome(X).
```