Esercizi sui tipi di dato
(Fondamenti di Informatica 1 – Walter Didimo)
Soluzioni

Esercizio 1 Dire cosa visualizza il seguente frammento di codice, motivando la risposta.

```
int i = 7;
int j = 2;
double d = i / j;
System.out.println (d);
d = 7 / 2;
System.out.println (d);
d = 7 / 2.0;
System.out.println (d);
d = (double)i/j;
System.out.println (d);
i = (int)d;
System.out.println (i);
```

Soluzione

La visualizzazione appare come segue:
3.0
3.0
3.5
3.5
3
Infatti, i primi quattro valori visualizzati sono di tipo double, ma i primi due valori sono convertiti a double solo dopo aver effettuato un'operazione di divisione tra interi.
L'ultimo valore visualizzato è invece un intero ottenuto troncando la parte decimale di un valore double.

Esercizio 2 Dire cosa visualizza il seguente frammento di codice, motivando la risposta.

```
boolean a,b,c,d;
a = true;
b = false;
c = true;
a = !a;
b = (a || b) && c;
c = !b;
c = (c || b) || (b && a);
d = (a || b && c) || (c && !b);
System.out.println (a);
System.out.println (b);
System.out.println (c);
System.out.println (!d);
```

Soluzione

La visualizzazione appare come segue:
false
false
true
false
Infatti, al termine dell'istruzione a = !a, la variabile a vale false, ed il suo valore non verrà più cambiato. L'istruzione b = (a || b) && c, assegna a b il valore false, poiché è false l'espressione (a || b); il valore di b non verrà più cambiato.
Le successive due istruzioni assegnano a c sempre il valore true, poiché !b è true ed anche (c || b) è true. Infine, è facile verificare che l'espressione (a || b && c) || (c && !b) vale false.

Esercizio 3 Il seguente frammento di codice contiene degli errori. Segnalare tali errori e correggerli.

```
int a = 27;
short b = 2*a;
double c = 1000 / b;     // c deve contenere la divisione esatta
                         //  tra 1000 e b
```

Soluzione

L'istruzione

short b = 2*a;

vuole assegnare ad uno short un valore di tipo int. Ciò causa un errore di compilazione che segnala una possibile perdita di precisione. Se si è sicuri che il valore dell'espressione 2*a non sarà superiore ad al massimo valore rappresentabile dal tipo short, si può correggere l'errore imponendo un cast esplicito, cioè:

short b = (short)(2*a);

L'istruzione

double c = 1000 / b;

invece non causa errori di compilazione, ma, in base al commento, dovrebbe restituire un risultato esatto della divisione tra 1000 e b, mentre al momento restituisce un risultato in cui la parte decimale è a zero. Questo perché la divisione 1000 / b è valutata come divisione tra interi ed il suo risultato sarà intero.