

Programmazione

25 Gennaio 2022

Rispondere alle domande utilizzando il foglio delle risposte segnando con una X quella che pensate sia la risposta corretta. Sarà possibile cambiare la risposta ad una domanda cerchiando la risposta errata. Risposte con più di un annullamento non verranno valutate. Le risposte dovranno essere scritte a penna (nera o blue). Ogni risposta corretta ha peso 2. Se non si risponde alla domanda si ottiene 0. Tra le possibili risposte ne è presente una errata che darà peso -2 se indicata come risposta. Il tempo per completare il test è di 45 minuti. Non è possibile utilizzare nessun tipo di materiale né alcun tipo di dispositivo smart (compresi gli smart-watch).

Nome: _____

Cognome: _____

Matricola: _____

1. Da quanti bit è costituito un *byte*?
A. 4 B. 8 C. 16 D. 32
2. Indicare il valore in base 10 del del numero binario 00001001.
A. 2 B. 7 C. 9 D. 16
3. Quanti sono i bit normalmente utilizzati in *Java* per rappresentare il tipo di dato `int`?
A. 6 B. 16 C. 32 D. 64
4. Quale è il valore della variabile `z` al termine dell'esecuzione della seguente porzione di codice?

```
int x = 5;  
int y = 2;  
double z = x/y;
```

- A. 2 B. 2.5 C. 7 D. Il codice genera un errore di tipo.
5. Quale è il valore della variabile `x` al termine dell'esecuzione della seguente porzione di codice?

```
int x = 9;  
int y = 7;  
if ((y<10) || (x++<10)) {  
    --x;  
}
```

- A. 11 B. 10 C. 9 D. 8
6. Consideriamo la seguente porzione di codice:

```
int[] x = new int[] {5, 4, 3, 2, 1};
```

Quale è l'indice del valore 4?

- A. 0 B. 1 C. 2 D. 3
7. In quale parte di memoria viene memorizzato il contenuto di un array?
 - A. in un'area di memoria riservata per la memorizzazione degli array;
 - B. sicuramente nello *stack*;
 - C. sicuramente nello *heap*;
 - D. nello *stack* o nello *heap* a seconda di dove viene dichiarato nel codice.

8. Consideriamo la seguente porzione di codice:

```
int[] x = new int[] {5, 4, 3, 2, 1};  
int[] y = x;  
y[x[4]] = 0;
```

Quali valori sono contenuti nell'array x?

- A. {5, 4, 3, 2, 1}
 - B. {5, 4, 3, 2, 0}
 - C. {5, 4, 0, 2, 1}
 - D. {5, 0, 3, 2, 1}
9. Quale valore conterrà la variabile *somma* al termine dell'esecuzione della seguente porzione di codice?

```
int[] x = new int[] {5, 4, 3, 2, 1, 0};  
int somma = 0;  
for(int i=0; i<x[i]; i++) {  
    somma += x[i];  
}
```

- A. 15;
 - B. 14;
 - C. 12.
 - D. l'esecuzione non termina a seguito di un errore;
10. Quali delle seguenti informazioni vengono associate alla firma di un metodo?
- A. nome del metodo, numero e tipo dei parametri, tipo di ritorno;
 - B. nome del metodo, numero nome e tipo dei parametri, e tipo di ritorno;
 - C. nome del metodo, numero nome e tipo dei parametri;
 - D. nome del metodo, numero e tipo dei parametri.

11. Quali delle seguenti affermazioni è **vera**?
- A. una classe può implementare al più un'interfaccia o una classe;
 - B. una classe può implementare più interfacce al patto di estendere almeno una classe astratta;
 - C. una classe può implementare al più un'interfaccia ed estendere un qualsiasi numero di classi (astratte o non);
 - D. una classe può estendere al più una classe (astratta o non) ed implementare un qualsiasi numero di interfacce.
12. Quali sono le differenze tra *interfacce* e *classi astratte*?
- A. nessuna, rappresentano esattamente la stessa cosa;
 - B. le interfacce definiscono solo i tipi dei metodi che una classe dovrà implementare mentre le classi astratte descrivono anche il loro stato e forniscono delle implementazioni;
 - C. le classi astratte definiscono solo i tipi dei metodi che una classe dovrà implementare mentre le interfacce descrivono anche il loro stato e forniscono delle implementazioni;
 - D. Ambedue definiscono i tipi dei metodi che una classe dovrà implementare, solo che le classi astratte, a differenza delle interfacce, possono essere istanziate.
13. Un metodo *m* in una data classe *A* si dice *sovraccaricato* (*overloaded*) quando:
- A. la classe *A* fornisce due metodi con il nome *m* ma signature diverse;
 - B. la classe *A* fornisce un secondo metodo con nome diverso ma con la stessa signature di *m*;
 - C. *A* fornisce un secondo metodo con la stessa signature di *m*, ma tipo di ritorno diverso.
 - D. una classe non può sovraccaricare un metodo;
14. Un metodo *m* in una data classe *B* che estende la classe *A* si dice *sovrascritto* (*overriding*) quando:
- A. la classe *A* fornisce un metodo con nome *m* ma signature diversa rispetto alla versione presente nella classe *B*;
 - B. la classe *A* fornisce un metodo con la stessa signature di *m* nella classe *B*;
 - C. la classe *A* fornisce un metodo con la stessa signature di *m*, ma tipo di ritorno diverso.
 - D. una classe non può sovrascrivere un metodo;

15. Consideriamo la dichiarazione delle seguenti classi Java:

```
class ClasseA {  
  
    public void m1( ) {  
        System.out.println(" ClasseA->m1()");  
        m2();  
    }  
  
    public void m2( ) {  
        System.out.println(" ClasseA->m2()");  
    }  
  
}  
  
class ClasseB extends ClasseA {  
  
    public void m2( ) {  
        System.out.println(" ClasseB->m2()");  
    }  
  
}
```

Consideriamo, inoltre, la seguente porzione di codice:

```
ClasseA c = new ClasseB ();  
c.m1();
```

Quale è il risultato della sua esecuzione?

A. Viene stampato a video:

```
ClasseA->m1()  
ClasseA->m2()
```

B. Viene stampato a video:

```
ClasseA->m1()  
ClasseB->m2()
```

C. Viene stampato a video:

```
ClasseB->m1()  
ClasseB->m2()
```

D. Il codice non viene eseguito a seguito di un errore di tipo.