

GOCCE DI JAVA

Capitolo 5

Domande vero-falso

Selezionando una risposta, il bottone corrispondente diventa verde se la risposta è giusta, altrimenti diventa rosso.



GOCCE
DI JAVA





**GOCCE
DI JAVA**



1. Esiste una corrispondenza ben precisa fra nomi delle classi e nomi di file: in un file sorgente ci può essere un'unica classe 'public', che deve avere lo stesso nome del file.

Vero

Falso



**GOCCE
DI JAVA**



2. Un'eccezione non catturata si propaga verso l'esterno, di blocco in blocco: se raggiunge il `main`, provoca la terminazione immediata del programma.

Falso

Vero



GOCCE
DI JAVA



3. Se in un'istruzione `try-catch` è presente la clausola `finally`, il suo blocco di istruzioni viene comunque eseguito (indipendentemente dal fatto che un'eccezione sia stata generata/catturata o meno).

Falso

Vero



GOCCE
DI JAVA



4. La programmazione orientata agli oggetti incapsula dati e metodi in classi: i dati ed i metodi di una classe sono fortemente collegati tra di loro.

Falso

Vero



**GOCCE
DI JAVA**



5. Come l'istanza di un tipo di dato primitivo (come `int`) è detta variabile, così l'istanza di un tipo classe è detta oggetto.

Falso

Vero



GOCCE
DI JAVA



6. Tutti gli oggetti Java sono passati per valore.

Falso

Vero



GOCCE
DI JAVA



7. La definizione di una classe con modificatore `public` deve essere memorizzata in un file che ha esattamente lo stesso nome della classe e come estensione `.class`.

Falso

Vero



GOCCE
DI JAVA



8. Ogni classe Java è una sottoclasse di `Object` e, quindi, eredita i metodi definiti nella classe `Object`.

Falso

Vero



GOCCE
DI JAVA



9. Le parole chiave `public` e `private` sono nomi di variabili d'istanza.

Falso

Vero



GOCCE
DI JAVA



10. Quando un oggetto viene creato, generalmente le sue variabili d'istanza sono incapsulate nell'oggetto e possono essere utilizzate solo attraverso metodi della classe dell'oggetto (normalmente attraverso i metodi privati della classe).

Falso

Vero



GOCCE
DI JAVA



11. Normalmente si hanno diversi costruttori per un classe: ciò può essere ottenuto attraverso la sovrascrittura dei costruttori.

Falso

Vero



GOCCE
DI JAVA



12. Per scrivere il miglior programma possibile, il programmatore deve conoscere perfettamente i dettagli interni di ogni classe che il programma utilizza.

Falso

Vero



GOCCE
DI JAVA



13. Spesso, le classi possono essere derivate da altre classi che forniscono operazioni che le nuove classi possono utilizzare, oppure includono come campi oggetti di altre classi: tale riutilizzo del software può migliorare enormemente la produttività di un programmatore.

Falso

Vero



GOCCE
DI JAVA



14. I modificatori `public` e `private` sono utilizzati per controllare l'accesso alle variabili ed ai metodi d'istanza di una classe.

Falso

Vero



GOCCE
DI JAVA



15. I membri privati di una classe così come la definizione dei metodi pubblici (ovvero l'implementazione della classe) non sono accessibili a chi utilizza la classe.

Falso

Vero



GOCCE
DI JAVA



16. Fornire un metodo `set` ed uno `get` per dei dati privati è equivalente a rendere quei dati pubblici.

Falso

Vero



GOCCE
DI JAVA



17. Quando un oggetto viene creato, le sua variabili d'istanza devono essere esplicitamente inizializzate da un metodo costruttore.

Falso

Vero



GOCCE
DI JAVA



18. Le variabili d'istanza possono essere inizializzate implicitamente al loro valore di default (0 per tipi numerici, **false** per variabili Booleane e **null** per tipi classe), possono essere inizializzate in un costruttore oppure possono essere impostate successivamente alla creazione dell'oggetto.

Falso

Vero



GOCCE
DI JAVA



19. Se il programmatore non fornisce un costruttore senza argomenti, Java ne crea automaticamente uno per la classe.

Falso

Vero



GOCCE
DI JAVA



20. Se dei costruttori sono forniti per una classe, ma nessuno di essi è un costruttore senza argomenti, e viene effettuato un tentativo di invocare un costruttore senza argomenti per inizializzare un oggetto della classe, viene generato un errore di compilazione.

Falso

Vero



GOCCE
DI JAVA



21. I metodi di una classe possono essere sovraccaricati, ovvero diversi metodi di una classe possono avere esattamente lo stesso nome.

Falso

Vero



GOCCE
DI JAVA



22. Variabili d'istanza pubbliche possono essere manipolate solo dai metodi della classe.

Falso

Vero



GOCCE
DI JAVA



23. I metodi `set` sono anche detti metodi mutatori (in quanto tipicamente cambiano un valore).

Falso

Vero



GOCCE
DI JAVA



24. Accedere a dati privati attraverso metodi `set` e `get` non solo protegge le variabili d'istanza dal ricevere valori non corretti, ma anche isola gli utilizzatori della classe dalla rappresentazione delle variabili: quindi, se la rappresentazione cambia (tipicamente, per motivi di miglioramento delle prestazioni), solo l'implementazione dei metodi di accesso deve essere modificata e gli utilizzatori non devono effettuare alcun cambiamento fintanto che l'interfaccia fornita dai metodi rimane la stessa.

Falso

Vero



GOCCE
DI JAVA



25. Il riutilizzo del software garantisce una qualità del codice molto superiore a quella che normalmente potremmo ottenere da soli.

Falso

Vero



GOCCE
DI JAVA



26. Alcune variabili d'istanza devono poter essere modificate ed altre no: il programmatore può utilizzare la parola chiave **constant** per specificare che una variabile non è modificabile e che ogni tentativo di modificarla produce un errore.

Falso

Vero



GOCCE
DI JAVA



27. Una variabile `final` non può essere modificata da un'assegnazione dopo che è stata inizializzata: una tale variabile deve essere inizializzata nella sua dichiarazione.

Falso

Vero



GOCCE
DI JAVA



28. In un metodo che abbia un parametro formale con lo stesso nome di uno dei campi della classe che lo contiene, bisogna usare la parola chiave `this` per accedere al campo (altrimenti, faremmo incorrettamente riferimento al parametro formale del metodo).

Falso

Vero



GOCCE
DI JAVA



29. Ogni oggetto di una classe ha la propria copia di tutte le variabili d'istanza della classe.

Falso

Vero



GOCCE
DI JAVA



30. Una variabile di classe statica rappresenta informazione specifica di un oggetto ed ogni oggetto di una classe ha la propria copia dei dati statici.

Falso

Vero



GOCCE
DI JAVA



31. I membri statici di una classe sono disponibili solo dopo che il primo oggetto della classe viene istanziato.

Falso

Vero



GOCCE
DI JAVA



- 32.** Ogni variabile statica di una classe ed ogni metodo statico di una classe esiste e può essere utilizzato anche se nessun oggetto di quella classe è stato istanziato.

Falso

Vero



GOCCE
DI JAVA



- 33.** L'ereditarietà è una forma di riutilizzo del software nella quale nuove classi sono create a partire da classi esistenti ereditando le loro proprietà ed i loro comportamenti ed arricchendo questi ultimi con funzionalità che la nuova classe richiede.

Falso

Vero



GOCCE
DI JAVA



34. In Java, una classe è derivata da una o più superclassi.

Falso

Vero



GOCCE
DI JAVA



35. Il vantaggio dell'ereditarietà consiste nella possibilità di definire nella sottoclasse aggiunte o modifiche delle funzionalità ereditate dalla superclasse.

Falso

Vero



GOCCE
DI JAVA



36. Ogni oggetto di una superclasse può essere pensato come un oggetto delle sottoclassi di quella superclasse.

Falso

Vero



GOCCE
DI JAVA



37. L'insieme degli oggetti rappresentati da una sottoclasse è normalmente più grande dell'insieme degli oggetti rappresentato da una qualunque superclasse di quella sottoclasse.

Falso

Vero



GOCCE
DI JAVA



38. I costruttori di una superclasse sono ereditati dalle sottoclassi di quella superclasse.

Falso

Vero



GOCCE
DI JAVA



39. Un oggetto di una sottoclasse può essere trattato come un oggetto della sua superclasse.

Falso

Vero



GOCCE
DI JAVA



40. Una conversione esplicita può sempre convertire un riferimento ad una superclasse in un riferimento ad una sottoclasse.

Falso

Vero



**GOCCE
DI JAVA**



41. Se non vi è nessuna invocazione esplicita al costruttore della superclasse, Java automaticamente tenta di invocare il costruttore di default della superclasse.

Falso

Vero



GOCCE
DI JAVA



42. Per invocare esplicitamente il costruttore di default della superclasse bisogna utilizzare l'istruzione `super()`;

Falso

Vero



GOCCE
DI JAVA



43. Sebbene un oggetto di una sottoclasse sia un oggetto della superclasse, il tipo della sottoclasse è diverso dal tipo della superclasse.

Falso

Vero



GOCCE
DI JAVA



44. Assegnare un oggetto di una sottoclasse ad un riferimento della superclasse e, quindi, tentare di accedere a membri della sola sottoclasse con il riferimento alla superclasse, genera un errore di compilazione.

Falso

Vero



GOCCE
DI JAVA



45. Quando usiamo l'ereditarietà per creare una nuova classe da una classe esistente, la nuova classe eredita solo i comportamenti della classe esistente: successivamente, aggiungiamo campi per adattare la classe alle nostre necessità.

Falso

Vero



GOCCE
DI JAVA



46. La compilazione produce tanti file `.class` quante sono le classi all'interno dell'unità di compilazione.

Falso

Vero



GOCCE
DI JAVA



47. Ogni classe deve esplicitamente dichiarare il package di cui fa parte tramite l'istruzione corrispondente.

Falso

Vero



**GOCCE
DI JAVA**



48. Esiste una corrispondenza biunivoca fra nome di un package e posizione nel file system delle classi del package.

Falso

Vero



**GOCCE
DI JAVA**



49. L'esecuzione di una classe che appartiene ad certo package può essere richiesta posizionandosi direttamente nella directory che contiene il file `.class` corrispondente.

Falso

Vero



GOCCE
DI JAVA



50. La libreria `java.lang` va importata esplicitamente.

Falso

Vero



GOCCE
DI JAVA



51. Per la distruzione di oggetti, Java mette a disposizione operatori ad-hoc.

Vero

Falso



GOCCE
DI JAVA



52. Una classe può essere dichiarata sottoclasse (`extends`) di due sottoclassi diverse (che non sono una sottoclasse dell'altra) e in questo caso eredita attributi e metodi di entrambe.

Vero

Falso



GOCCE
DI JAVA



53. Un metodo di una classe non può essere contemporaneamente sovraccaricato e sovrascritto.

Vero

Falso



GOCCE
DI JAVA



54. Se `Frazione` è una classe con due attributi di tipo intero `num` e `den`, allora il seguente metodo è un costruttore sintatticamente corretto per la classe:

```
public void Frazione(int n, int d) {  
    num = n;  
    den = d;  
}
```

Vero

Falso



GOCCE
DI JAVA



55. La classe `Object` è una superclasse di tutte le classi, le quali devono esplicitamente estendere la classe `Object` mediante la clausola `extends`

Vero

Falso



GOCCE
DI JAVA



56. Il seguente codice genera un errore di compilazione

```
public class A { public A(int n) {} }  
public class B extends A { }
```

Vero

Falso



GOCCE
DI JAVA



57. Il seguente programma Java produce errori.

```
class A {  
    int a; private void f(int b) { a = b; }  
}  
public class M {  
    public static void main(String[] args) {  
        A a = new A(); a.f(3);  
    }  
}
```

Vero

Falso



GOCCE
DI JAVA

