

Cognome	
Nome	
Matricola	
Anno di corso (II, III, etc.)	

## Linguaggi di Programmazione

a.a. 2013-2014

Esame del 3-02-2014

**Istruzioni:** Per ognuna delle seguenti 30 domande sono elencate 5 possibili risposte, di cui soltanto una è corretta. Marcare con una X la casella della risposta che si ritiene corretta. Non è consentito marcare più di una risposta.

Ogni risposta corretta vale 1 punto; ogni risposta errata vale -0.25 punti; ogni risposta non fornita vale 0 punti.

Non è consentito l'uso di appunti, né di altre forme di informazione. Tempo a disposizione: 90 minuti.

---

1. Qual è la sintassi corretta per scrivere un commento per la Javadoc ?

- ☒ `/** ... */`
- ☐ `\\ ... \\`
- ☐ `// ... //`
- ☐ `/** ... **/`
- ☐ `\* ... * \`

2. Quale delle seguenti liste contiene solo parole chiave di Java ? (Tutte tranne una contengono una parola chiave di C che non esiste in Java)

- ☒ static, throw, implements, do, import
- ☐ instanceof, volatile, finally, default, protected
- ☐ catch, final, class, switch, include
- ☐ extends, interface, constant, Double, break
- ☐ public, int, throws, define, abstract

3. Quale delle seguenti affermazioni è vera ?

- ☒ Un file `.java` può contenere più classi, ma solo una può essere `public`
- ☐ Un file `.java` non può contenere più di una classe
- ☐ Un file `.java` può contenere più classi `public`
- ☐ Un file `.java` può contenere più classi, ma deve contenerne almeno una con lo stesso nome del file
- ☐ Un file `.java` può contenere più classi, ma almeno una deve essere `public`

4. Con quale modificatore deve essere dichiarata una variabile d'istanza, affinché il suo valore non possa essere modificato una volta inizializzato ?

- ☒ `final`
- ☐ `static`
- ☐ `final static`
- ☐ `private`
- ☐ `private static`

5. Quale delle seguenti affermazioni è corretta ?

- ☒ Una variabile locale di tipo numerico non viene inizializzata per default
- ☒ Una variabile locale non può essere contrassegnata come `static`
- ☐ Una variabile locale non può contenere un oggetto
- ☐ Un metodo può accedere alle variabili locali di un altro metodo
- ☐ Il metodo `main` non può contenere variabili locali

6. Quale delle seguenti non è una convenzione sulle maiuscole adottata in Java ?

- ☒ I nomi dei metodi iniziano per lettera maiuscola
- ☐ I nomi delle classi iniziano per lettera maiuscola
- ☐ I nomi delle variabili e degli attributi iniziano per lettera minuscola
- ☐ I nomi delle costanti sono scritti in lettere maiuscole
- ☐ I nomi composti sono scritti con la notazione a cammello

7. Qual è la sintassi corretta per il metodo `main` ?

- ☒ `public static void main (String[] args) { }`
- ☐ `public static void Main (String[] args) { }`
- ☐ `public static void main (String, args) { }`
- ☐ `public static void main (String[], args) { }`
- ☐ `public static void Main (String, args) { }`

### 8. Una variabile di istanza di tipo int dichiarata con l'attributo static :

- ☒ può essere modificata, ma il suo valore cambia per tutti gli oggetti della classe
- ☐ può essere acceduta ma non può essere modificata
- ☐ può assumere valori diversi per oggetti diversi
- ☐ non può essere acceduta
- ☐ non viene inizializzata per default

### 9. Perché compilando il seguente programma si ottiene un errore ?

```
public class A {  
    static int a;  
    public static void main (String[] args) {  
        A b;  
        System.out.println(b.a);  
    }  
}
```

- ☒ Perché nessun oggetto è stato costruito per il riferimento b
- ☐ Perché l'attributo a non è stato inizializzato
- ☐ Perché l'attributo a è statico e quindi non può essere acceduto
- ☐ Perché non è stato dichiarato nessun metodo costruttore
- ☐ Perché l'attributo a non è stato dichiarato come public

### 10. Qual è l'output del seguente programma ?

```
class A {  
    public static void main (String[] args) {  
        for (int i = 0; i < 4; i += 2) {  
            System.out.print(i + " ");  
        }  
        System.out.println(i);  
    }  
}
```

- ☒ Si ottiene un errore di compilazione
- ☐ 0 2 4
- ☐ 0 2
- ☐ 0 2 2
- ☐ 0 1 2 3

11. Per conoscere la lunghezza di una stringa si può :

- ☒ invocare su di essa il metodo `length()`
- ☐ accedere al suo attributo `length`
- ☐ accedere al suo attributo `size`
- ☐ invocare su di essa il metodo `size()`
- ☐ invocare su di essa il metodo `getLength()`

12. Sia `s` un riferimento alla stringa "Laurea in Informatica". Quale delle seguenti istruzioni ritorna false?

- ☒ `s.trim().toLowerCase().equals("Laurea in Informatica");`
- ☐ `s.equalsIgnoreCase("Laurea in Informatica");`
- ☐ `s.equals("Laurea in Informatica");`
- ☐ `s.trim().equalsIgnoreCase("Laurea in Informatica");`
- ☐ `s.toLowerCase().trim().equalsIgnoreCase("Laurea in Informatica");`

13. Sia `s` un riferimento alla stringa "prova 123". Cosa restituisce l'istruzione `s.substring(0,1);` ?

- ☒ La stringa "p"
- ☐ La stringa "pr"
- ☐ Il carattere "p"
- ☐ La stringa vuota ""
- ☐ Un errore di compilazione

14. Quali delle seguenti affermazioni sul metodo `charAt` è corretta ?

- ☒ ha un parametro di tipo `int` e restituisce un `char`
- ☐ ha due parametri, uno di tipo `String` e uno di tipo `int`, e restituisce un `char`
- ☐ ha un parametro di tipo `char` e restituisce un `int`
- ☐ ha un parametro di tipo `String` e restituisce un `char`
- ☐ ha un parametro di tipo `String` e restituisce un `int`

15. Per che cosa si potrebbe usare l'espressione `String [] a = new String [b] ?`

- ☒ Per creare un array di **b** riferimenti a oggetti di tipo stringa
- ☐ Per creare un array di **b** stringhe vuote
- ☐ Per creare un array di **a** stringhe vuote
- ☐ Per creare un array di **a** riferimenti a oggetti di tipo stringa
- ☐ Per creare il riferimento **a** per un oggetto di tipo stringa

16. Dato l'array `int [] n = {0,1,2,3,4,5,6}`, cosa stampa l'istruzione `System.out.println(n[1] + n[3]);` ?

- ☒ La stringa 4
- ☐ La stringa 2
- ☐ La stringa 0+2
- ☐ La stringa 1+3
- ☐ La stringa `n[1]n[3]`

17. Sia **a** un `ArrayList` avente per parametro la classe **A**, e sia **b** un oggetto della classe **A**. Quale delle seguenti istruzioni per aggiungere **b** ad **a**, in coda, è corretta ?

- ☒ `a.add(b);`
- ☐ `add(a,b);`
- ☐ `Arrays.add(a,b);`
- ☐ `add(b,a);`
- ☐ `a = a.add(b);`

18. Quale delle seguenti affermazioni è corretta ?

- ☒ Se **a** è una stringa, la sua dimensione è data da `a.length()`
- ☐ Se **a** è una stringa, la sua dimensione è data da `a.length`
- ☐ Se **a** è un array, la sua dimensione è data da `a.length()`
- ☐ Se **a** è un `ArrayList`, la sua dimensione è data da `a.length()`
- ☐ Se **a** è un `ArrayList`, la sua dimensione è data da `a.length`

19. Quale delle seguenti istruzioni non genera un errore di compilazione ?

- ☒ `ArrayList<Boolean> bits = new ArrayList<Boolean>();`
- ☐ `ArrayList<Boolean> bits = new ArrayList();`
- ☐ `ArrayList<boolean> bits = new ArrayList<boolean>();`
- ☐ `ArrayList<boolean> bits = new ArrayList;`
- ☐ `ArrayList<Boolean> bits = new ArrayList<Boolean>;`

20. Sia B una sottoclasse di A. Quale delle seguenti affermazioni è corretta ?

- ☒ Gli attributi di B sono quelli dichiarati in B piú quelli dichiarati in A
- ☐ B non può avere attributi propri
- ☐ Gli attributi di B sono solo quelli dichiarati in A
- ☐ Gli attributi di B sono solo quelli dichiarati in B
- ☐ Nessuna delle altre risposte è corretta

21. Sia m un metodo public della classe A, avente due parametri. Sia B una sottoclasse di A, e supponiamo che in B sia presente una versione del metodo m con un solo parametro. Quale delle seguenti affermazioni è corretta ?

- ☒ Su un oggetto di B si possono invocare sia la versione del metodo m con due parametri, sia quella con un solo parametro
- ☐ Su un oggetto di B si può invocare solo la versione del metodo m con due parametri
- ☐ Su un oggetto di B si può invocare solo la versione del metodo m con un parametro
- ☐ Su un oggetto di B si può invocare la versione del metodo m con due parametri solo se l'oggetto è contenuto in una variabile di A
- ☐ Nessuna delle altre risposte è corretta

22. Quale delle seguenti affermazioni è corretta ?

- ☒ Il modificatore `private` rende possibile l'accesso solo all'interno della classe
- ☐ Il modificatore `private` rende possibile l'accesso solo all'interno della classe e delle sue sottoclassi
- ☐ Il modificatore `private` rende possibile l'accesso solo all'interno della classe e del package che la contiene
- ☐ Il modificatore `private` rende possibile l'accesso da qualunque classe
- ☐ Il modificatore `private` rende impossibile l'accesso da parte di chiunque

23. Quale delle seguenti affermazioni è sbagliata ?

- ☒ Una variabile può essere contemporaneamente `private` e `protected`
- ☐ Una variabile può essere contemporaneamente `private` e `static`
- ☐ Una variabile può essere contemporaneamente `protected` e `static`
- ☐ Una variabile può essere contemporaneamente `private` e `final`
- ☐ Una variabile può essere contemporaneamente `protected` e `final`

24. Qual è il comportamento del seguente programma, contenuto nel file B.java ?

```
class B {  
    public static void main (String[] args) {  
        System.out.println("superclasse"); }  
}  
class A extends B {  
    public static void main (String[] args) {  
        System.out.println("sottoclasse"); }  
}
```

- ☒ Viene stampata la stringa “superclasse”, perché B è la classe che ha lo stesso nome del file
- ☐ Viene stampata la stringa “sottoclasse”, perché il `main` di A sovrascrive quello di B
- ☐ Si ottiene un errore di compilazione, perché vi è un conflitto tra i due `main`
- ☐ Viene stampata la stringa “superclasse” e poi viene stampata la stringa “sottoclasse”, perché i due `main` vengono eseguiti in successione
- ☐ Si ottiene un errore di compilazione, perché nessuna classe è dichiarata `public`

25. Quale delle seguenti affermazioni è falsa ?

- ☒ Un costruttore può essere dichiarato `final`; in tal caso, non è sovrascrivibile
- ☐ Una variabile di istanza può essere dichiarata `final`; in tal caso, non è modificabile
- ☐ Una variabile locale può essere dichiarata `final`; in tal caso, non è modificabile
- ☐ Una classe può essere dichiarata `final`; in tal caso, non è derivabile
- ☐ Un metodo può essere dichiarato `final`; in tal caso, non è sovrascrivibile

26. Quale delle seguenti affermazioni è falsa ?

- ☒ Una sottoclasse di una classe `abstract` deve definire tutti i metodi `abstract`
- ☐ Una classe può estendere una sola classe `abstract`
- ☐ Una classe `abstract` non può essere istanziata
- ☐ Oltre a metodi `abstract`, una classe `abstract` può contenere metodi non `abstract`
- ☐ Una classe `abstract` può avere un costruttore

27. Quale delle seguenti affermazioni è falsa ?

- ☒ Una classe può implementare al più un'interfaccia
- ☐ In un'interfaccia non si possono definire variabili ma solo costanti
- ☐ Un'interfaccia non può essere istanziata
- ☐ Una classe che implementa un'interfaccia deve definirne tutti i metodi
- ☐ Un'interfaccia non ha un costruttore di default

**28. Sia a un metodo suscettibile di lanciare un'eccezione di tipo controllato e una di tipo non controllato, e supponiamo che si vogliano gestire all'interno di a entrambe le eccezioni. Allora occorre :**

- ☒ Mettere le istruzioni che potrebbero lanciare le due eccezioni in un blocco `try`, seguito da due blocchi `catch`, uno per ognuna delle due eccezioni
- ☐ Mettere le istruzioni che potrebbero lanciare le due eccezioni in un blocco `try`, seguito da un unico blocco `catch` per entrambe le eccezioni
- ☐ Mettere le istruzioni che potrebbero lanciare le due eccezioni in un blocco `try`, e segnalare nella dichiarazione del metodo solo l'eccezione di tipo controllato per mezzo della clausola `throws`
- ☐ Mettere le istruzioni che potrebbero lanciare le due eccezioni in un blocco `try`, e segnalare nella dichiarazione del metodo entrambe le eccezioni per mezzo della clausola `throws`
- ☐ Mettere le istruzioni che potrebbero lanciare l'eccezione controllata in un blocco `try`, seguito da un blocco `catch`, e le istruzioni che potrebbero lanciare l'eccezione non controllata dopo il blocco `catch`

**29. Qual è l'output del seguente programma ?**

```
public class Test {  
    public static void a() throws Exception {  
        throw new Exception();  
    }  
    public static void main(String args[]) {  
        try { a(); System.out.print("exception "); }  
        catch (Exception e) { System.out.print("caught "); }  
        finally { System.out.print("finished"); }  
    }  
}
```

- ☒ caught finished
- ☐ finished
- ☐ Si ottiene un errore di compilazione
- ☐ exception caught finished
- ☐ exception finished

**30. Quale delle seguenti affermazioni è falsa ?**

- ☒ La classe `FileNotFoundException` è una sottoclasse di `RuntimeException` e i suoi oggetti sono eccezioni di tipo non controllato
- ☐ Le eccezioni della classe `RuntimeException` e delle sue sottoclassi sono di tipo non controllato
- ☐ Le eccezioni della classe `IOException` e delle sue sottoclassi sono di tipo controllato
- ☐ La classe `IndexOutOfBoundsException` è una sottoclasse di `RuntimeException` e i suoi oggetti sono eccezioni di tipo non controllato
- ☐ La classe `FileNotFoundException` è una sottoclasse di `IOException` e i suoi oggetti sono eccezioni di tipo controllato