

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

## Linguaggi di Programmazione

a.a. 2013-2014

Esame del 24-02-2014

**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 per scrivere un commento che si estende su una sola riga ?

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

2. Quale di questi non è un tipo fondamentale di Java ?

- ☒ array
- ☐ double
- ☐ boolean
- ☐ short
- ☐ byte

3. Quale delle seguenti affermazioni è falsa ?

- ☒ Un metodo non può contenere variabili
- ☐ Una classe può contenere variabili
- ☐ Un metodo può contenere variabili
- ☐ Una classe può contenere metodi
- ☐ Una classe può non contenere variabili

4. Con quale/i modificatore/i 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 falsa ?

- ☒ Una variabile locale può essere contrassegnata come `static`
- ☐ Una variabile locale non è inizializzata per default
- ☐ Una variabile locale può contenere un oggetto
- ☐ Le variabili locali cessano di esistere alla chiusura del metodo in cui sono dichiarate
- ☐ Un metodo non può accedere alle variabili locali di un altro metodo

6. 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) { }`

7. Quale delle seguenti affermazioni è falsa ?

- ☒ Un metodo `static` non può essere sovraccarico (overloading)
- ☐ Un metodo `static` non può accedere alle variabili di istanza, a meno che queste non siano anch'esse `static`
- ☐ Un metodo `static` non si invoca su un oggetto
- ☐ Un metodo `static` può essere invocato anche se nessun oggetto è stato istanziato
- ☐ Un metodo costruttore non può essere `static`

8. Quale dei seguenti è un metodo costruttore ?

- ☒ `public A () { }`
- ☐ `public static A () { }`
- ☐ `public void A () { }`
- ☐ `public static void A () { }`
- ☐ `public new A () { }`

9. Una variabile di istanza di tipo boolean 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

10. Quale delle seguenti affermazioni è falsa ?

- ☒ Il costruttore di default può essere sovrascritto (overriding)
- ☐ Il costruttore di default ha lo stesso nome della classe
- ☐ Il costruttore di default è implicitamente public
- ☐ Il costruttore di default è implicitamente final
- ☐ Il costruttore di default può essere sovraccaricato (overloading)

11. 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 + " ");  
        }  
    }  
}
```

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

12. Cosa stampa l'istruzione `System.out.println( +0+"+0" );` ?

- ☒ 0+0
- ☐ 00
- ☐ Niente, perché provoca un errore di compilazione
- ☐ +00
- ☐ 0++0

13. Sia `s` un riferimento alla stringa "prova 123". Cosa restituisce l'espressione `s.substring(s.length()-3,s.length())` ?

- ☒ La stringa "123"
- ☐ La stringa " 123"
- ☐ Un errore di compilazione
- ☐ La stringa "23"
- ☐ L'intero 6

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

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

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

- ☒ Per creare a riferimenti a oggetti di classe A
- ☐ Per creare a oggetti di classe A
- ☐ Per creare un oggetto di classe A[]
- ☐ Per creare il riferimento b per un oggetto di classe A[]
- ☐ Per creare il riferimento b per un oggetto di classe A

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

- ☒ 6
- ☐ 4
- ☐ 1+3
- ☐ 2+4
- ☐ Nulla, perché il compilatore segnala un errore

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` come primo elemento (e spostare tutti gli altri a destra) è corretta ?

- ☒ `a.add(0,b);`
- ☐ `a.add(b,0);`
- ☐ `add(a,b);`
- ☐ `b.add(a,0);`
- ☐ `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 dei seguenti metodi contenuti nella classe `java.util.Arrays` permette di riempire gli elementi di un array con uno stesso valore ?

- ☒ `fill()`
- ☐ `equals()`
- ☐ `toString()`
- ☐ `sort()`
- ☐ `copyOf()`

20. Sia B una sottoclasse di A (entrambe classi pubbliche e contenute nello stesso package). Quale delle seguenti affermazioni è corretta ?

- ☒ B eredita tutti gli attributi di A
- ☐ B eredita tutti gli attributi di A ad eccezione di quelli **private**
- ☐ B eredita solo gli attributi **public** di A
- ☐ B eredita solo gli attributi **protected** di A
- ☐ Nessuna delle altre risposte è corretta

21. 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

22. Quale delle seguenti affermazioni è falsa ?

- ☒ Una variabile può essere contemporaneamente **public** e **protected**
- ☐ Una variabile può essere contemporaneamente **public** e **static**
- ☐ Una variabile può essere contemporaneamente **protected** e **static**
- ☐ Una variabile può essere contemporaneamente **public** e **final**
- ☐ Una variabile può essere contemporaneamente **protected** e **final**

23. 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

**24. L'overloading di metodi è :**

- ☒ La possibilità di avere metodi con lo stesso nome ma con diversi parametri
- ☐ La possibilità di sovrascrivere un metodo definito in una superclasse
- ☐ La possibilità di ereditare i metodi non **private** della superclasse
- ☐ La possibilità di avere metodi con lo stesso nome in classi diverse
- ☐ La possibilità di avere metodi con lo stesso nome ma con valori di ritorno diversi

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

- ☒ Una classe **abstract** può avere al più una sottoclasse
- ☐ Una classe può estendere una sola classe **abstract**
- ☐ Una classe **abstract** non può essere istanziata
- ☐ Oltre a metodi **abstract**, una classe **abstract** può contenere dati e metodi non **abstract**
- ☐ Una classe **abstract** può avere un costruttore non **abstract**

**26. Cosa succede se si prova a compilare il seguente programma ?**

```
class A {  
    public static void main (String[] args) {  
        Object a = new A();  
        a = new Object();  
    }  
}
```

- ☒ Non si ottiene nessun errore di compilazione
- ☐ Si ottiene un errore di compilazione, perché c'è incompatibilità di tipi tra **Object** e **A**
- ☐ Si ottiene un errore di compilazione, perché **A** non è sottoclasse di **Object**
- ☐ Si ottiene un errore di compilazione, perché il riferimento **a**, una volta usato per un oggetto di classe **A**, non può essere usato per un oggetto di un'altra classe
- ☐ Si ottiene un errore di compilazione, perché il riferimento **a** può essere usato solo per un oggetto di classe **Object**

**27. Quale delle seguenti affermazioni sulla serializzazione è falsa ?**

- ☒ Una classe con attributi **static** non può essere serializzata
- ☐ Per potere serializzare oggetti, una classe deve implementare l'interfaccia **Serializable**
- ☐ L'interfaccia **Serializable** non contiene nessun metodo da implementare
- ☐ Una volta serializzati, gli oggetti possono essere scritti mediante l'uso della classe **ObjectOutputStream**
- ☐ Cambiamenti al codice di una classe che serializza oggetti rendono gli oggetti serializzati in precedenza illegibili

28. Sia a un metodo suscettibile di lanciare un'eccezione, e nel quale non è stato installato un gestore per tale eccezione. Allora nella dichiarazione del metodo :

- ☒ bisogna dichiarare l'eccezione che il metodo potrebbe lanciare, per mezzo della clausola **throws**, solo se l'eccezione è di tipo controllato
- ☐ bisogna dichiarare l'eccezione che il metodo potrebbe lanciare, per mezzo della clausola **throws**, sia se l'eccezione è di tipo controllato che non controllato
- ☐ bisogna dichiarare l'eccezione che il metodo potrebbe lanciare, per mezzo della clausola **throw**, solo se l'eccezione è di tipo controllato
- ☐ bisogna dichiarare l'eccezione che il metodo potrebbe lanciare, per mezzo della clausola **throw**, sia se l'eccezione è di tipo controllato che non controllato
- ☐ non è necessario dichiarare l'eccezione che il metodo potrebbe lanciare, perché questa verrà propagata all'esterno

29. Qual è l'output del seguente programma ?

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

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

30. Quale delle seguenti affermazioni è vera ?

- ☒ Se **ClasseEccezione** è una sottoclasse di **Exception**, l'istruzione **catch(ClasseEccezione e)** può catturare eccezioni di **ClasseEccezione** e di qualsiasi sua sottoclasse
- ☐ La classe **Error** è una sottoclasse di **RuntimeException**
- ☐ Le eccezioni di **RuntimeException** sono controllate
- ☐ Ogni istruzione che può lanciare un'eccezione deve trovarsi all'interno di un blocco **try**, altrimenti si ottiene un errore di compilazione
- ☐ Le eccezioni di **IOException** sono non controllate