

Vero o falso?

# 1 - Vero o falso?

- `int a[]` è un oggetto
- In Java non esistono le variabili globali

## 2 - Vero o falso?

- Il costruttore chiama automaticamente il costruttore della superclasse con gli stessi parametri. Se nella superclasse non è disponibile un costruttore con la stessa firma, viene chiamato il costruttore vuoto.
- Il metodo `finalize()` chiama automaticamente il corrispondente metodo della superclasse

## 3 - Vero o falso?

- L'esistenza in una classe di un metodo `f(int x)`, e in una sua superclasse di un metodo `f(String s)` è un esempio di overriding
- Se `a.equals(b)` è vero, deve essere `a.hashCode()==b.hashCode()`

## 4 - Vero o falso?

- In un progetto ci possono essere più classi con lo stesso nome
- In una classe ci può essere un solo metodo chiamato main

## 5 - Vero o falso?

- L'esistenza in una classe di un metodo `f(int x)` e di uno `f(String s)` è un esempio di overloading
- Se `a.equals(b)` è falso, deve essere `a.hashCode()!=b.hashCode`

## 6 - Vero o falso?

- In un progetto ci possono essere tanti metodi `public static void main(String a[]);`
- Il main può sempre accedere a qualunque variabile di istanza della classe in cui è contenuto.

# 7 - Vero o falso?

- Nel main non posso definire variabili non statiche
- Le variabili dichiarate static non sono modificabili dai metodi privati



## 8 - Vero o falso?

- Nei metodi statici non è possibile leggere e scrivere le variabili di istanza
- Se una classe è astratta è permesso usarla per effettuare ereditarietà multipla

## 9 - Vero o falso?

- Una Interface può estendere una classe
- Non è possibile istanziare una Collection

# 10 - Vero o falso?

- Il garbage collector di Java sospende l'esecuzione del programma finchè non ha finito di liberare la memoria.
- Di default l'operatore == e il metodo equals fanno la stessa cosa.

# 11 - Vero o falso?

- Se A è padre di B la scrittura `B a=new A();` genera errore a runtime
- Se A è padre di B la scrittura `B a=(B)(new A());` genera errore a compile time

## 12 - Vero o falso?

- Poichè Java usa sempre dynamic binding, esso usa sempre la heap e mai lo stack.
- Java usa solo la heap perché tutta la memoria è allocata dinamicamente con le new()

# 13 - Vero o falso?

- In C++ le variabili globali non stanno nè in Heap nè in Stack
- Gli identificatori di oggetto sono concettualmente equivalenti ai puntatori

# 14 - Vero o falso?

- Non è possibile chiamare il metodo equals su una classe generica a meno che essa non dichiari l'interfaccia Comparable
- 
- Non è possibile chiamare il metodo clone su una classe generica a meno che essa non dichiari l'interfaccia Cloneable
  - Un oggetto ed un suo clone sono identici.