

F.A.Q
su EREDITARIETÀ e VISIBILITÀ

DOMANDA 1

D: Può una classe derivata ridefinire *pubblico* un metodo o una funzione statica che nella classe base non lo era?

R: Sì: la classe derivata può ridefinire metodi e funzioni ampliandone la visibilità rispetto al suo “status” precedente; *non può invece restringerla*.

- quindi, un metodo protected (o package) può essere ridefinito come pubblico in una sottoclasse
- l’inverso non è consentito e dà errore di compilazione:
“attempting to assign weaker access privileges”

DOMANDA 2

D: Un dato *statico* è di fatto condiviso fra tutte le istanze di una classe? Chi può accedervi?

R: Il dato statico appartiene alla classe come componente software: come tale è condiviso fra tutte le sue istanze. Perciò è accessibile sia (ovviamente) alle funzioni statiche della classe, sia a tutte le istanze della classe medesima:

- quindi, ogni modifica fatta a tale dato da qualunque istanza sarà visibile anche dalle altre istanze...
- ... e anche, ovviamente, dalla classe medesima.

DOMANDA 3

D: Può un *metodo (non statico)* accedere a un dato *statico* (di questa o di un'altra classe)?

R: Sì: trattandosi di un dato di un componente software (la classe), purché non sia privato, *chiunque* può accedervi:

- da dentro la stessa classe, un metodo o una funzione statica possono accedervi semplicemente per nome:
ad esempio, `flag = 8;`
- da un'altra classe, si accede con il nome assoluto qualificato: ad esempio, `Prova.flag = 7;`

DOMANDA 3 bis

D: E il viceversa? Può una *funzione statica* accedere a un dato *non-statico* o invocare un metodo (non statico)?

R: No: la funzione statica può fare riferimento solo a enti che esistono staticamente: non può accedere a enti di natura dinamica, che staticamente *non esistono*.

DOMANDA 4

D: Se una classe base possiede un dato *statico*, in una classe derivata esso *è replicato* o *si condivide* lo stesso dato della classe base?

R: Si condivide lo stesso dato della classe base:

- le istanze della classe base e della classe derivata *fanno tutte riferimento allo stesso dato statico* della classe base
- ogni modifica, da chiunque fatta, si ripercuote su tutte.

DOMANDA 5

D: Si può, da una classe base, accedere a un dato *statico* di una classe derivata da essa?

R: Sì: la classe derivata non è diversa da qualunque altra classe. L'accesso è quindi possibile mediante il nome assoluto qualificato (sempre che la visibilità lo consenta - vedi domande 2 e 3):

– ad esempio, dalla classe **Prova**:

```
Prova2.flag = 7;
```

DOMANDA 6

D: Può una classe definire *due versioni* di uno stesso metodo, con diverso tipo di ritorno?

R: No: l'overloading consente la definizione di metodi (o funzione statiche) con egual nome, ma solo se distinguibili dalla lista dei parametri: il tipo di ritorno non viene considerato. Quindi, un simile tentativo dà luogo a errore di compilazione:
“method is already defined”

DOMANDA 7

D: Neanche se una versione è *pubblica*
e l'altra *no*?

R: Neppure: il livello di visibilità non può essere sfruttato per “nascondere” ambiguità.

D'altronde, sarebbe uno “sporco trucco”...

... e creerebbe evidenti incongruenze col principio generale appena esposto in varie situazioni

-- basti pensare al caso in cui le due versioni fossero in una stessa classe...

DOMANDA 8

D: Può una classe *derivata* definire una nuova versione di un metodo già esistente *nella classe base*, ma con diverso tipo di ritorno?

R: No: la situazione è identica al caso già esposto. Il fatto che la ridefinizione sia tentata in una sotto-classe, anziché nella stessa classe, è irrilevante. Anche qui si ha perciò errore di compilazione:
“cannot override - attempting to use incompatible return type”

DOMANDA 9

D: Può una classe *base* contenere un main, dentro al quale si istanziano oggetti di una classe *derivata*?

R: Sì: il main ha il solo requisito di stare in una classe pubblica. Più in generale, *qualunque metodo* di una classe può istanziare oggetti di una classe derivata -- come di ogni altra classe: il fatto che le due classi siano “parenti” è irrilevante (fermo restando il rispetto dei diritti di accesso legati ai livelli di visibilità dei vari metodi e dati).

DOMANDA 10

D: Sempre nel caso di una classe *base* che contiene un main, può tale main accedere a dati *statici* di una classe *derivata*?

R: Sì: qualsiasi metodo o funzione statica della classe base (*e di ogni altra classe*, visibilità permettendo) può accedere a dati statici della classe derivata, con il nome qualificato.
(*si vedano anche le risposte alle domande 2 e 3*)