Teoria di Java

TIPI DI MODIFICATORI DI VISIBILITA'

Public : Un elemento se è public è accessibile, modificabile e richiamabile dall'interno della classe stessa, dall'esterno della classe e dalle classi che la ereditano.

Private: Un elemento private è accessibile, modificabile e richiamabile solo all'interno della classe stessa. Ho bisogno di metodi di get() e set() per poter modificare o utilizzare i dati esternamente

Protected: Un elemento protected è accessibile, modificabile e richiamabile all'interno della classe stessa, dall'interno delle classi che la ereditano e visibile solo dalle classi dello stesso package e dalle sottoclassi.

Default: Il metodo è come se fosse pubblico ma sarà visibile solo dalle classi appartenenti allo stesso package.

TIPI DI DATO

Primitivi: Sono tipi di dato elementari che hanno una grandezza specifica in memoria. L'inizializzazione per i tipi primitivi è automatica.

Riferimento: Sono detti di tipo oggetto, ad esempio delle classi. L'inizializzazione è a null.
Vi è bisogno dell'utilizzo della new seguita dal nome dell'oggetto da creare. Posso passare valori in input durante la creazione.
La definizione di una costante in Java è possibile con il modificatore final davanti ad una variabile.
Il confronto tra i tipi primitivi è possibile con '==' mentre per i tipi riferimento è possibile attraverso il metodo equals().

DIFFERENZA STATIC E DI ISTANZA

Definendo un parametro come static, il suo valore viene mantenuto con la classe ed è uguale in tutte le istanze. Un metodo statico per esempio può solo riferirsi a variabili o metodi statici e non può interagire con variabili d'istanza.

Definendo un parametro come d'istanza ogni istanza sarà diversa dalle altre come se fossero oggetti differenti. Un metodo d'istanza può riferirsi a tutto in una classe a differenza di quelli static. Un oggetto viene creato con la new che da come risultato il riferimento dell'oggetto appena creato.
Il contesto di esecuzione è relativo all'istanza stessa.

La classe Class ha come istanza tutte le classi esistenti nel programma. E' utile per poter passare la classe ad un metodo.

EREDITARIETA'

In Java è possibile ereditare da una sola classe attraverso la chiave extends. Si eredita tutto il codice tranne blocchi di inizializzazione e i costruttori.
Posso utilizzare quindi un oggetto della sottoclasse al posto di un oggetto della classe. Il tipo di una sottoclasse è compatibile con quello di una classe.
Se non definisco una classe con extends, la classe è sottoclasse di Object. Si possono estendere tutte le classi tranne le classi definite come final.

CLASSE ASTRATTA

Una classe astratta è dichiarata con il modificatore abstract. Tipicamente è una classe incompleta perchè appunto contiene dei metodi astratti, ovvero metodi a cui manca il corpo.
Una classe non astratta per estendere una astratta deve fornire un'implementazione di tutti i suoi metodi. Non si possono creare istanze di una classe astratta ma è possibile assegnare istanze delle sue sottoclassi.
Il vantaggio è la riduzione delle ridondanze quando le informazioni comuni a più classi non definiscono completamente una classe

INTERFACCIA

Un'interfaccia ha una struttura simile ad una astratta, ma i metodi devono essere tutti astratti. Un classe può dichiarare di implementare una o più interfacce però deve fornire un'implementazione per tutti i metodi astratti oppure deve essere dichiarata astratta perchè è come se contenesse i metodi che non implementa.
Un'interfaccia non può estendere una classe. Un'interfaccia può estendere una o più interfacce. Un'interfaccia eredita tutti i metodi astratti delle sue superinterfacce.

OGGETTO - CLASSE - METACLASSSE

OGGETTO:
Un oggetto è una entità che il programmatore può manipolare all'interno del programma mediante l'invocazione di metodi (puoi "manipolare" anche le classi tramite l'invocazione di metodi. Un oggetto e' una istanza di una classe aventi variabili dati "propri" e dati comuni (statici)):
Quando è invocato un metodo di un oggetto, vengono svolte all'interno dell'oggetto una serie di attività con lo scopo di realizzare la funzione per cui il metodo è stato implementato. Tutti gli oggetti dello stesso tipo sono istanze della stessa classe.
CLASSE:
Si intende la descrizione astratta di una classe di oggetti, ovvero le variabili e le funzioni membro specificate nella classe sono quelle che poi potranno essere invocate utilizzando l'oggetto Che differenza c'è tra classe ed oggetto ? La classe è la descrizione astratta di un tipo di dato, specifica cioè i metodi e le variabili che quel

tipo di dato possiede. L’oggetto è una istanza della classe, ovvero quando istanziamo una variabile definendola di una certa classe, noi creiamo un oggetto di quella classe rappresentato dal nome della variabile che abbiamo istanziato.
La classe rappresenta l'astrazione di qualcosa di concreto. L'oggetto invece è qualcosa di concreto che viene rappresentato da una classe.
METACLASSE:
Una metaclasse è una classe le cui istanze sono a loro volta classi. In Java tutte le classi sono concettualmente considerate istanze dell'unica metaclasse Class

THIS E SUPER

This non fa altro che puntare all’oggetto a cui appartiene risolvendo possibili problemi di ambiguità. La parola super è utilizzata in Java per riferirsi alla superclasse. Può essere usato solo con ot-notation e non può essere iterato: *super.super.x.*
La parola super viene usato tipicamente per accedere a un metodo della superclasse non accessibile perché sovrascritto (Ad esempio per richiamare il costruttore).