Instanziare una Classe: gli oggetti

Gli oggetti sono caratterizzati da

- Classe di appartenenza - tipo (ne descrive attributi e metodi)
- Stato (valore attuale degli attributi)
- Identificatore univoco (reference - handle - puntatore)

Per creare un oggetto occorre

- Dichiarare una istanza
- La dichiarazione non alloca spazio ma solo una riferimento (puntatore) che per default vale null
- Allocazione e inizializzazione
- Riservano lo spazio necessario creando effettivamente l'oggetto appartenente a quella classe

---

Notazioni Puntate

Le notazioni puntate possono essere combinate

- `System.out.println("Hello world!");`
- **System** è una classe del `package java.lang`
- **out** è una variabile di classe contenente il riferimento ad un oggetto della classe **PrintStream** che punta allo standard output
- **println()** è un **metodo** della classe PrintStream che stampa una linea di testo

---

Operazioni su reference

Uso degli operatori relazionali == e !=

- Attenzione: il test di uguaglianza viene fatto sul puntatore (**reference**) e NON sull'oggetto
- Stabiliscono se i **reference** si riferiscono allo stesso oggetto
- È definita l'**assegnazione**
- È definito l'**operatore punto** (notazione puntata)
- **NON** è prevista l'aritmetica dei puntatori

Operazioni su istanze

- Le principali operazioni che si possono effettuare sulle variabili che riferiscono istanze di una classe sono:
    - assegnamento
    - confronto
    - invocazione di metodi

- Il valore di una variabile di tipo strutturato è il riferimento ad un oggetto (istanza di una classe)
- Una stessa variabile può riferire oggetti diversi in momenti diversi a seguito di operazioni di assegnazione sul suo valore
- Se la variabile contiene il valore null non riferisce nessun oggetto in quel momento

---

Accesso a metodi e attributi non static

- La sintassi è simile al caso precedente, ma ovviamente l'accesso/invocazione è possibile solo tramite un'istanza specifica (ed ogni accesso è diversificato):
- Nel corpo di un metodo non `static` si può accedere a qualunque attributo e metodo della stessa classe
- All'interno del corpo di un metodo si possono riferire in modo abbreviato attributi e metodi definiti nella stessa classe
- Se nel corpo di un metodo non `static` appare il nome di un metodo o attributo non `static` della sua classe è sottinteso che sia riferito all'istanza su cui è stato invocato il metodo

---

Oggetti e riferimenti

- Le variabili hanno un nome, gli oggetti no
- Per utilizzare un oggetto bisogna passare attraverso una variabile che ne contiene il riferimento
- Uno stesso oggetto può essere riferito da più variabili e quindi essere raggiunto tramite nomi diversi (di variabili)
- Il rapporto variabili - oggetti riferiti è dinamico, il riferimento iniziale non necessariamente rimane legato all'oggetto per tutta la sua esistenza
- Se un oggetto non è (più) riferito da nessuna variabile diventa irraggiungibile (e quindi interviene il garbage collector)

Confronti tra variabili di tipo strutturato

- E' possibile applicare gli operatori di confronto == e != a variabili di tipo strutturato
- Se uno dei due termini del confronto è il valore null si verifica se una certa variabile riferisce un oggetto oppure no, p.e. saluto3 != null
- Se entrambi i termini del confronto sono variabili, si verifica se hanno lo stesso valore (cioè riferiscono esattamente lo stesso oggetto)

Confronto tra riferimenti vs. confronto tra oggetti

- Usare == fa il confronto tra i riferimenti non fra i valori contenuti negli oggetti (p.e. le sequenze di caratteri contenute nelle istanze di String)
- Di solito si vogliono confrontare i contenuti non i riferimenti: per questo si usa il metodo **equals**
- Il metodo booleano equals della classe String accetta come argomento il riferimento ad un altro oggetto e ritorna true se le stringhe contenute sono uguali (in modo case sensitive), false altrimenti
- Il metodo booleano equalsIgnoreCase fa lo stesso senza distinguere maiuscole/minuscole