

Java e Object-Oriented Programming, Richiami

- Programmazione Procedurale vs OOP
 - Equals, Comparable

Programmazione Procedurale

- Esiste un programma principale ed eventuali procedure o moduli a sua disposizione
- Il programma, in sostanza, esegue una serie di istruzioni, possibilmente chiamando procedure o funzioni
- Durante il calcolo il programma può interagire con dispositivi di input e/o output collegati all'utente, al file system, o ad altri programmi
- In caso di parallelismo, diversi programmi, autonomi, possono interagire tra di loro (processi)

Programmazione Orientata agli Oggetti

- Si definiscono delle classi che sono “stampi” per creare oggetti
- Gli oggetti hanno dei metodi associati che, solo se attivati, eseguono il codice corrispondente
- Esistono uno o più Thread (il main più eventuali altri) che creano oggetti e li “stimolano” chiamando i loro metodi
- I metodi di solito ottengono il loro input da parametri e restituiscono in output un valore di ritorno
- L'interazione con l'utente o con dispositivi di input e output è affidata ad oggetti e thread particolari (GUI, streams, ...)

Oggetti come entità che forniscono servizi

- Gli oggetti, raggruppati in classi, modellano entità del dominio del discorso
- Essi possiedono uno stato interno e possono essere creati in numero imprecisato dallo “stampo”, la classe
- Ogni oggetto, se non stimolato, non fa nulla
- L'oggetto è reattivo: esegue del codice, previsto dalla sua classe, solo se viene chiamato su di esso un metodo

Memoria: Stack e Heap

- La memoria, in particolare la parte denominata Heap, contiene un certo numero di oggetti
- Normalmente esiste un solo Thread in esecuzione, corrispondente al main
- A tale Thread è associata una zona di memoria, lo Stack, che contiene variabili locali al main
- All'attivazione di un metodo lo stack si congela e gli viene sovrapposta una attivazione, un altro stack, che contiene i parametri e le variabili locali del metodo
- Al termine del metodo l'attivazione scompare e si ritorna a quella precedente

Equazioni di secondo grado

- Nella programmazione procedurale:
 - I parametri sono chiesti in input o assegnati nel codice
 - Il programma fa i calcoli e restituisce in output i risultati
- Nella programmazione ad oggetti:
 - L'equazione diventa un oggetto di una classe
 - Un oggetto è in grado, su richiesta, di fornire i risultati

Equazioni di Secondo Grado

- Vediamo il codice...

Uguaglianza

- All'interno di una classe è opportuno definire un criterio di uguaglianza fra oggetti
- Due oggetti fisicamente allocati in zone diverse di memoria possono quindi essere considerati uguali
- L'uguaglianza implica che i due oggetti rappresentano la stessa entità nel dominio del discorso
- Ridefinizione dei metodi `equals` e `hashCode` della classe `Object`

Uguaglianza

- All'interno di una classe è opportuno definire un criterio di uguaglianza fra oggetti
- Due oggetti fisicamente allocati in zone diverse di memoria possono quindi essere considerati uguali
- L'uguaglianza implica che i due oggetti rappresentano la stessa entità nel dominio del discorso
- Ridefinizione dei metodi equals e hashCode

Ordinamento Naturale

- Può essere utile anche definire un ordinamento totale tra gli elementi di una classe
- Uso dell'interfaccia `Comparable<T>`
- Metodo `int compareTo(T o)` restituisce un valore < 0 se l'oggetto `this` precede `o` nell'ordinamento
- `compareTo(T o)` restituisce un valore > 0 se l'oggetto `this` segue `o` nell'ordinamento
- `compareTo(T o)` restituisce un `0` se e solo se `this.equals(o)`

Pretty Print

- Di solito si ridefinisce anche il metodo `toString()` della classe `Object`
- Esso deve fornire una descrizione intellegibile dell'oggetto sotto forma di stringa
- Utile in molti contesti, soprattutto Test e simili

-
-
- Vediamo il codice...