

Programmazione I

A.A. 2002-03

Programmazione Orientata agli Oggetti:

Lavorare con gli oggetti (Lezione XXVII)

Prof. Giovanni Gallo

Dr. Gianluca Cincotti

Dipartimento di Matematica e Informatica

Università di Catania

e-mail : { gallo, cincotti } @dmi.unict.it

Facciamo gli “oggetti” Prima del codice, chiedersi

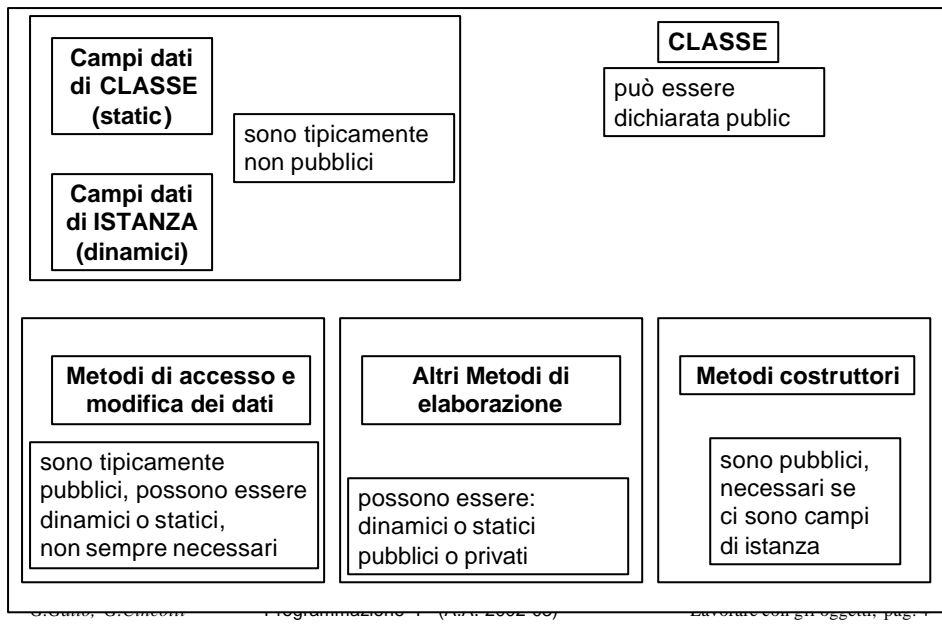
- Cosa rappresenta l’oggetto che voglio costruire?
- Quali funzioni svolge l’oggetto?
- Sono abbastanza semplici o dovrei spezzare l’oggetto in due oggetti diversi?
- L’oggetto richiede dati o è solo una collezione di metodi utili ad altre classi?
- Quali dati sono necessari per la sua descrizione?
- Quali dati possono essere derivati dai dati necessari?
- Quali dati debbono essere assegnati con parametro alla costruzione e quali possono avere valori di default?
- Quali dati e informazioni dell’oggetto sono utili all’esterno dell’oggetto e quali servono solo al programmatore per la sua particolare implementazione e vanno dichiarati privati?

Facciamo gli “oggetti”

Cosa è necessario nel codice

- Un nome descrittivo dell’oggetto;
- Quali le variabili di classe (static) e quelle di istanza?
- Costruttori (default di Java?, uno solo? Più d’uno?)
- Metodi propri dell’oggetto (per calcoli, elaborazioni, ecc.)
- Metodi di accesso per le variabili che si vogliono pubblicare.
- Metodo toString(), se utile (dettagli in seguito)

Struttura di un oggetto (repetita juvant)



Come diventare OOProgrammer: consigli per lo studio e la professione

- In uno sport è l'allenamento che fa il campione;
- Nella musica è l'esercizio che fa il virtuoso;
- Nella cucina è la pratica quotidiana che fa lo chef;
- Nel...

La programmazione è una arte/scienza di natura PRATICA.

**PER PROGRAMMARE BENE BISOGNA
PROGRAMMARE MOLTO** (non basta aver capito!)

Esercizi

- Disegnare una classe per gli intervalli aperti della retta reale; (*vedi cartella "intervalli" in L27*)
- Disegnare una classe per i rettangoli del piano;
- Disegnare una classe per gli studenti di Informatica del primo anno (anagrafe, esami e voti);
- Disegnare una classe per i dipendenti di una ditta (anagrafe, qualifica, stipendio, ferie);
- Disegnare una classe per le squadre del campionato;
- La tombola! (*vedi cartella "tombola" in L27*)

Esercizi più complessi e divertenti

- Studiare una famiglia di oggetti: il caso degli intervalli (vedi cartella “intervalli” in L27):
- Studiare famiglie di oggetti che modellino:
 - I rettangoli del piano;
 - Gli studenti del primo anno di Informatica (matricola, esami, voti);
 - I punti dello spazio terne (x,y,z) e le loro reciproche distanze;
 - I poligoni del piano (assegnati con i loro vertici);
 - eccetera (fantasia!)

Vedi anche...

- La tombola (ad oggetti!) (cartella “tombola” in L27)
- I progetti proposti nel file progetti_proposti di L27. (dagli anni precedenti)

Un “vero” programma JAVA è...

Un ambiente ove diversi oggetti vengono:

- creati, e distrutti,
- mantengono informazioni e dati,
- si scambiano messaggi e informazioni,
- elaborano informazioni.

Uno degli oggetti (quello chiamato per primo dalla JVM) è il “registra” delle operazioni.

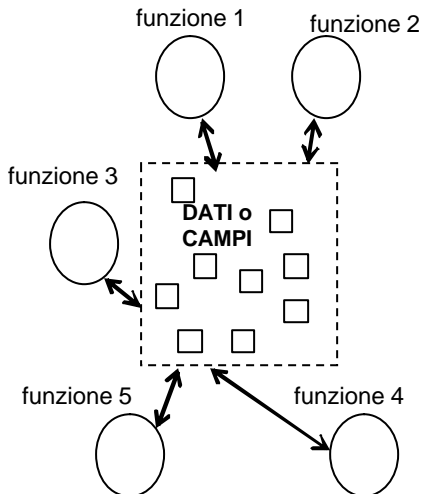
Attori sempre presente ma poco visibili?

Gli oggetti della “famiglia System” !

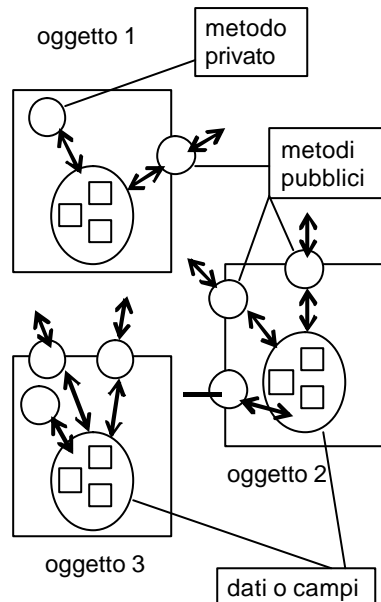
graficamente

Programmazione strutturata

Tutte le funzioni "condividono" i dati.



Programmazione ad oggetti



Relazioni tra oggetti (già osservate)

➤ **Contenimento.**

Un oggetto A contiene un oggetto B se tra le variabili di A è stato dichiarata e viene costruita una istanza dell'oggetto B.

Esempio: la classe gioco contiene una istanza di giocatore e due di dado.

➤ **Utilizzo.**

Un oggetto A utilizza i metodi pubblici di un altro oggetto B che non ha dichiarato o costruito.

Esempio: abbiamo utilizzato spesso la classe Math, la classe System.

Relazioni tra oggetti: nuova!

➤ **Eredità.**

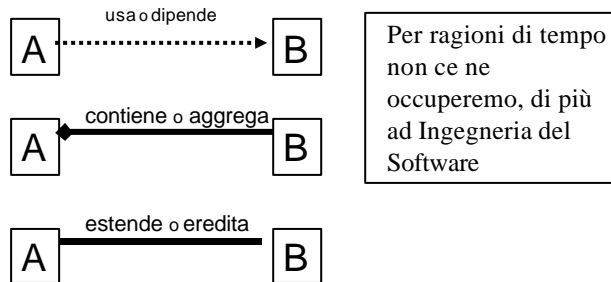
Una classe A eredita la classe B se assume tutte i suoi dati e metodi e li estende o modifica ulteriormente.

La pratica della eredità è di fondamentale importanza e dà all'OOProgramming una grande potenza...

Sarà oggetto delle prossime lezioni.

Linguaggi grafici per il progetto dei programmi OO

Esistono numerose convenzioni la più diffusa è quella proposta dall'Unified Modeling System (UML):



Fine