

Corso di Laurea Ingegneria Informatica

Fondamenti di Informatica 1

Dispensa 2

Programmi e Oggetti Software

Alfonso Miola
Settembre 2007

Contenuti

- ❑ **Cosa è un programma**
- ❑ **Cosa significa programmare**
 - Il punto di vista del programmatore
 - Il punto di vista dell'utente
- ❑ **Paradigmi di programmazione**
- ❑ **La programmazione ad oggetti**
- ❑ **Oggetti software – Esempi**
 - L'oggetto System.out
 - L'oggetto Math
 - Gli oggetti String

Prerequisiti

Questo capitolo **presuppone** la conoscenza degli argomenti già trattati nelle **precedenti lezioni** di questo corso e del corso di **Laboratorio di Informatica**, con particolare riferimento al **capitolo 1** del **libro di testo**

Che cosa è un programma

- ❑ Un **programma (o applicazione)** è usato da un utente per risolvere un problema con l'ausilio di un calcolatore
- ❑ Un programma mostra a un utente la rappresentazione, nel calcolatore, di una porzione di mondo reale fatto di **oggetti**
 - la porzione di mondo rappresentata da un programma è la **realtà di interesse** del programma
 - un programma **rappresenta** nel calcolatore gli **oggetti** di una certa realtà di interesse, intesi e quindi rappresentati come **dati e operazioni**

Che cosa è la programmazione

La programmazione è :

- **scrittura (sintesi) di programmi**
- **controllo** — un calcolatore fa quello che gli viene detto di fare
- **concretezza** — un calcolatore per eseguire un compito ha bisogno di istruzioni dettagliate
- **insegnamento** — un calcolatore impara a eseguire nuove operazioni **quando e solo se** gli viene detto come vanno realizzate
- **modellazione** — un programma rappresenta nel calcolatore una porzione di mondo, la realtà di interesse
- **astrazione** — il programmatore deve identificare le caratteristiche essenziali della realtà di interesse da modellare, evitando di descrivere dettagli inutili

Il punto di vista del programmatore

- ❑ Per un **programmatore**, un programma è un insieme di frasi (**istruzioni**) che descrivono una certa realtà di interesse in un linguaggio di programmazione
- ❑ Le **istruzioni** di un programma servono a descrivere il modo con cui devono essere rappresentati i **dati** della realtà di interesse del programma e le **azioni** che devono essere svolte da parte del calcolatore quando un utente chiede l'esecuzione di una certa operazione

Il calcolatore per un programmatore

- Un programmatore è normalmente interessato solo agli strati più esterni dell'architettura di un calcolatore (secondo lo schema a cipolla)



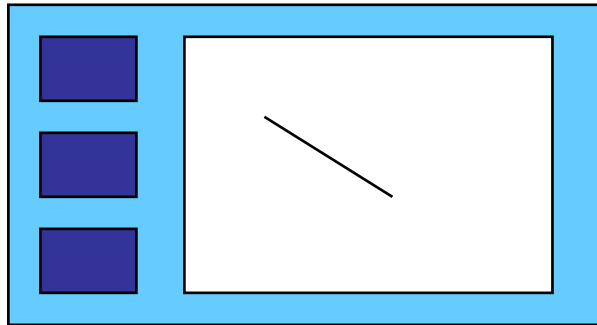
Gli strati più interni riguardano altre figure dell'informatica

Il punto di vista dell'utente

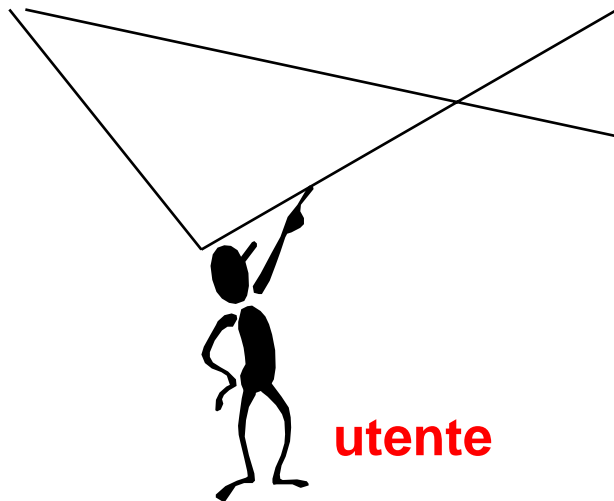
- Un utente è quindi interessato ad usare un programma per risolvere un problema facendo eseguire un insieme di operazioni a un calcolatore
 - ciascuna **operazione** costituisce uno **strumento** per l'utente del programma
 - l'**esecuzione** di ciascuna operazione corrisponde allo svolgimento di una **sequenza di azioni** da parte del calcolatore
 - l'**effetto finale** dell'esecuzione delle operazioni porta al raggiungimento di un **risultato** che costituisce per l'utente la soluzione del problema

I punti di vista dell'utente e del programmatore

interfaccia del programma



strati del programma nascosti dall'interfaccia



utente



programmatore

Paradigmi di programmazione

Esistono diversi approcci alla programmazione, chiamati paradigmi di programmazione

- nel paradigma di **programmazione imperativa**, un programma specifica le azioni che devono essere eseguite in sequenza per calcolare i risultati a partire dai dati
- nel paradigma di **programmazione funzionale**, un programma è la definizione di una funzione, parametrica rispetto ai dati, che calcola i risultati
- nel paradigma di **programmazione logica**, un programma è la descrizione delle proprietà che devono essere verificate dai risultati sulla base delle proprietà verificate dai dati

Programmazione orientata agli oggetti

Il paradigma di programmazione orientata agli oggetti è basato sul seguente punto di vista

- il mondo reale è fatto di **oggetti**
- dato che un programma è la rappresentazione nel calcolatore di una certa realtà di interesse, allora anche il programma è composto da oggetti, chiamati **oggetti software**
- un programma modella una realtà di interesse come una collezione di **oggetti software che cooperano**

Oggetti e oggetti software

□ Ciascun oggetto della realtà è:

- identificabile con un nome
- ha caratteristiche (proprietà) ben precise
- svolge funzioni specifiche

□ Un **oggetto software** non è quindi altro che una adeguata **rappresentazione virtuale**, nel **calcolatore**, **di un oggetto della realtà**, sia esso **concreto** oppure **astratto**

Oggetti software . . .

□ Un oggetto software ha:

- un **nome** — che permette di referenziarlo con una identificazione univoca
- uno **stato** — un insieme di proprietà che lo caratterizzano
- un **comportamento** — un insieme di operazioni che sa eseguire
 - si può richiedere a un oggetto software di eseguire una operazione mediante l'**invio di un messaggio**
 - l'uso di un oggetto software è descritto dalla sua **interfaccia**

. . . Oggetti software

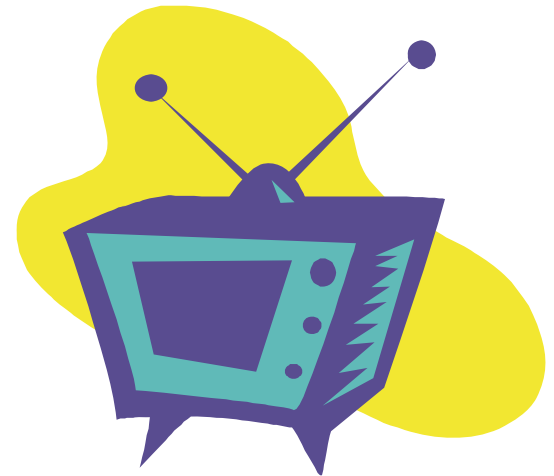
- **Un oggetto software è quindi un vero e proprio esecutore specializzato che:**
 - **risolve specifici problemi**
 - **può essere **riusato** ogni volta che serve**
 - **può richiedere ad **altri oggetti** di svolgere **funzioni di supporto** per completare i propri scopi**
 - **può essere **composto** con altri oggetti software **per cooperare** alla soluzione di problemi più complessi**

Programmi orientati agli oggetti

- ❑ La **programmazione** nel paradigma orientato agli oggetti consiste nella:
 - identificazione delle tipologie di **oggetti software necessari** per la soluzione di un dato problema
 - definizione del loro rispettivo **progetto**
- ❑ Un programma **coordina** le funzioni dei **diversi oggetti coinvolti**, per raggiungere il suo scopo finale
- ❑ L'**esecuzione** di un programma è quindi la **cooperazione di più oggetti**, ciascuno specializzato a risolvere un sottoproblema del problema posto

Un oggetto reale

- Nel mondo reale esiste un oggetto denominato “**televisore**”, nome che permette di far **referimento** all’oggetto, ovvero, di **referenziare** l’oggetto
- Il televisore ha un suo **comportamento**
 - sa fare delle cose
 - sa eseguire delle **operazioni**



Operazioni e messaggi

- Si può chiedere al televisore di eseguire una operazione premendo un tasto del telecomando
 - il telecomando invia al televisore la richiesta di esecuzione dell'operazione sotto forma di segnale elettromagnetico
- nella terminologia della programmazione orientata agli oggetti, la richiesta dell'esecuzione di una operazione viene fatta mediante l'**invio di un messaggio** a un oggetto



Interfaccia e uso

- L'**interfaccia** di un oggetto è la descrizione dell'insieme delle operazioni che l'oggetto sa eseguire
- L'interfaccia di un oggetto è il “**manuale d'uso**” dell'oggetto:
 - l'**elenco delle operazioni** possibili, cioè **ammissibili** e quindi **permesse**
 - la **forma sintattica dei messaggi** da inviare per richiedere una determinata operazione, cioè **come formulare la richiesta**
 - Il **significato** dell'operazione, la **semantica**, cioè **cosa si ottiene** dall'esecuzione dell'operazione

Proprietà e stato

- ❑ In ogni istante, un oggetto è in un certo **stato**, descritto da un insieme di **proprietà**
- ❑ Ciascuna **proprietà** è caratterizzata da un **nome**, dal **valore** corrente e dall'insieme dei valori ammessi
- ❑ L'**effetto** di una operazione può consistere nel **cambiamento dello stato** (cioè, del valore delle proprietà) dell'oggetto che la esegue

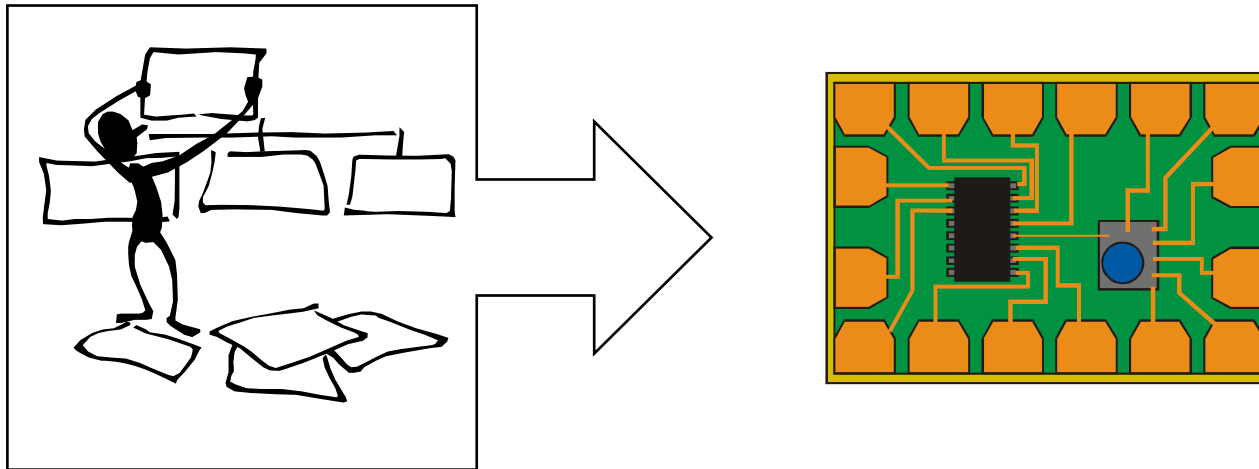
Progettazione

- La **progettazione** di un oggetto è la definizione delle caratteristiche dell'oggetto e dei dettagli per la sua realizzazione
 - viene prima definita l'interfaccia, cioè come l'oggetto potrà essere usato dall'utente
 - viene definita ciascuna delle operazioni dell'interfaccia
 - vengono definite le proprietà che determinano lo stato dell'oggetto

. . . Progettazione

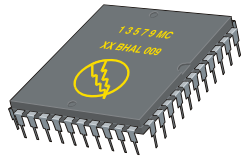
□ Ad esempio nella **progettazione** di un televisore

- il progettista stabilisce come l'oggetto potrà essere usato dall'utente
- il progettista definisce lo schema elettrico del televisore



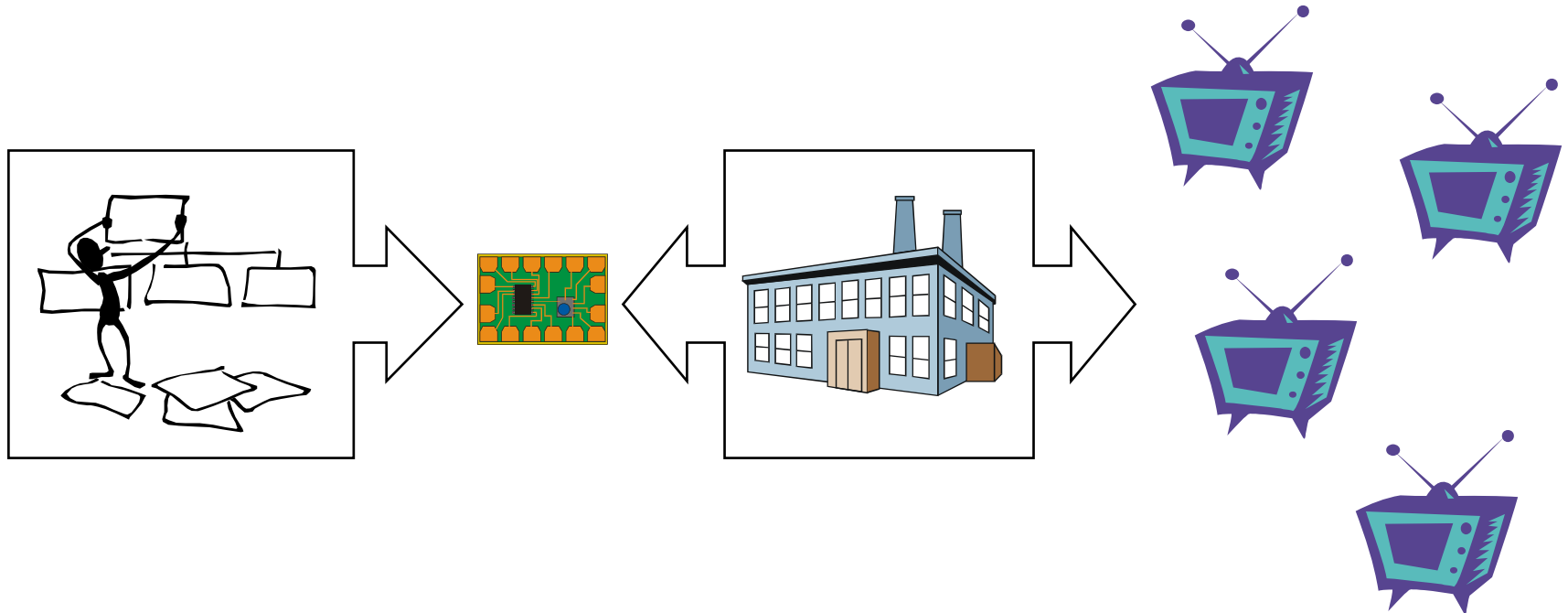
Progettazione e riuso

- Nella progettazione di un oggetto si considera l'esistenza di altri **oggetti già disponibili** il cui **riuso** può essere funzionale allo scopo
 - ad esempio, nella progettazione di un televisore si procede all'assemblaggio di
 - oggetti (componenti elettronici) standard, disponibili sul mercato, che vengono usati conoscendone l'interfaccia
 - componenti elettronici progettati in precedenza che vengono riusati senza modifiche
 - componenti elettronici progettati appositamente



Costruzione

- **La costruzione del televisore viene fatta da una fabbrica**
 - **la fabbrica costruisce televisori sulla base del progetto del televisore**



Progettazione e costruzione di oggetti software

□ Un oggetto software

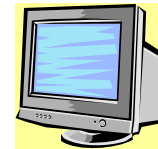
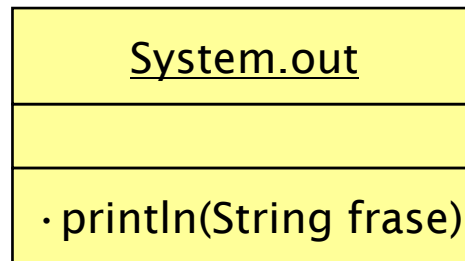
- è **progettato** e realizzato **da un programmatore**
 - la progettazione e realizzazione di componenti software è chiamata **implementazione**
 - una **classe** è il progetto di un oggetto software
 - i linguaggi per la definizione di classi sono i **linguaggi di programmazione orientati agli oggetti**
- è **costruito da un calcolatore**
 - gli oggetti software esistono solo nei calcolatori, cioè **esistono in modo virtuale**
 - il calcolatore viene usato come una **macchina virtuale** che sa gestire degli oggetti software, mediante l'esecuzione di istruzioni di appositi linguaggi di programmazione orientati agli oggetti, come ad esempio **Java**

Esempi di oggetti software

- ❑ **Gli oggetti software vengono descritti mediante un formalismo grafico (il linguaggio UML acronimo di Unified Modeling Language)**
- ❑ **Anche la cooperazione tra oggetti software viene descritta utilizzando il linguaggio UML**
 - **va sottolineato che il linguaggio UML non è un linguaggio di programmazione**
 - **UML è quindi solo un linguaggio per la descrizione di oggetti e classi**

L'oggetto software System.out

- ❑ Lo schermo del calcolatore è un oggetto che viene rappresentato nei programmi da un oggetto software
 - un oggetto software che modella lo schermo del calcolatore è **System.out**

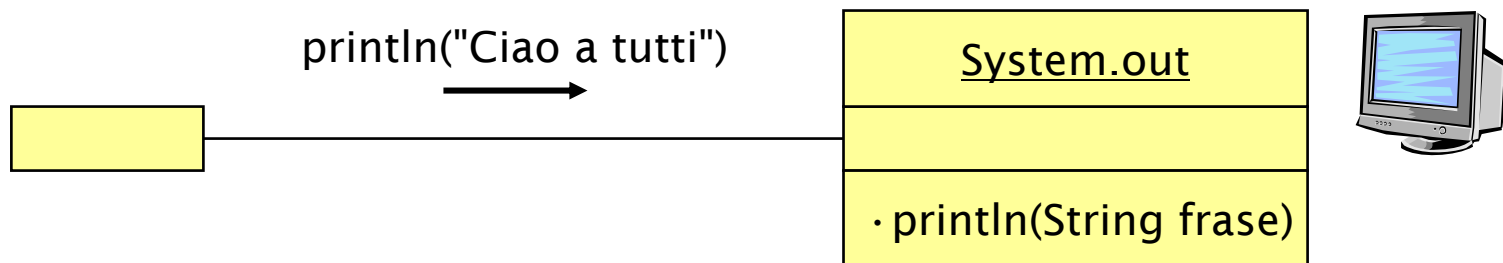


- ❑ Ciascun oggetto software è caratterizzato da
 - un **nome**
 - le **operazioni** identificate da un nome, eventualmente con dei **parametri** per i quali è definito un **tipo**

System.out — invio di un messaggio . . .

□ Per far visualizzare a **System.out** una frase, ad esempio la frase **Ciao a tutti**, è necessario inviargli il seguente messaggio

- `println("Ciao a tutti")`



Utenti e uso di oggetti

□ L'esempio ha mostrato l'uso di un oggetto software

- **l'uso di un oggetto software**
 - è descritto dalla sua interfaccia
 - si concretizza mediante l'invio di messaggi
- **per l'utente di un oggetto software è sufficiente conoscere l'interfaccia dell'oggetto software**
 - **l'utente di un oggetto software può ignorare come l'oggetto software sia fatto internamente**

Classi come progetto di oggetti software

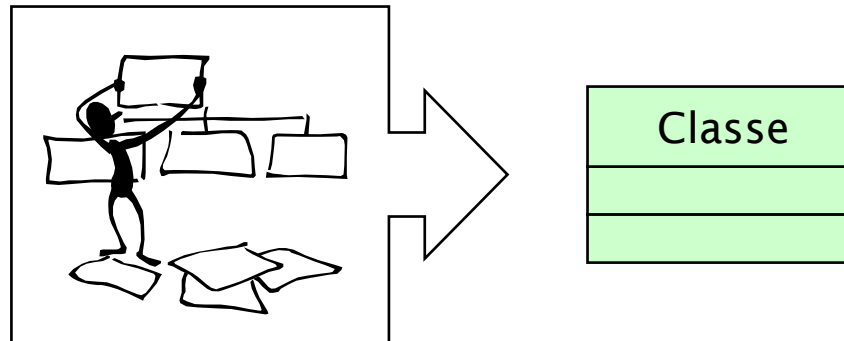
□ L'implementazione di un oggetto software consiste della

- definizione dell'interfaccia dell'oggetto software
 - la modalità d'uso dell'oggetto software deve essere definita in modo preciso
- definizione del progetto completo dell'oggetto software
 - definizione del **progetto dello stato e del comportamento** dell'oggetto software (ovvero della sua interfaccia)

Il progetto di un oggetto software si chiama
classe

Cosa è una Classe

- Una **classe** è il progetto di un oggetto software
 - l'implementazione di un oggetto software consiste nella definizione di una classe
- Il lavoro del progettista di un oggetto software
 - definire la classe per l'oggetto software



Costruzione di oggetti software

□ **Prima di poter utilizzare un oggetto software, è necessario costruirlo a partire dalla classe che ne definisce il progetto**

- si tratta evidentemente di una costruzione di tipo software, puramente virtuale, infatti gli oggetti software esistono solo nella memoria del calcolatore

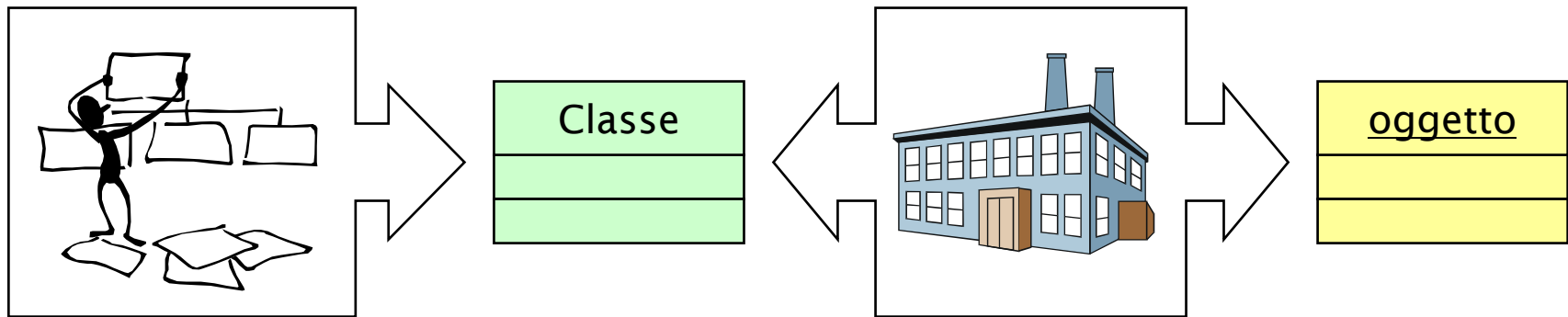
□ **Come si costruisce un oggetto software?**

- la **definizione della classe** per un oggetto software è scritta dal **programmatore** usando come formalismo un linguaggio di programmazione (orientato agli oggetti)
- la **costruzione di oggetti software** viene svolta dal **calcolatore**, usato come macchina virtuale in grado di eseguire programmi scritti nel linguaggio di programmazione scelto

Modalità di costruzione statica

□ Nella modalità di costruzione statica di oggetti software

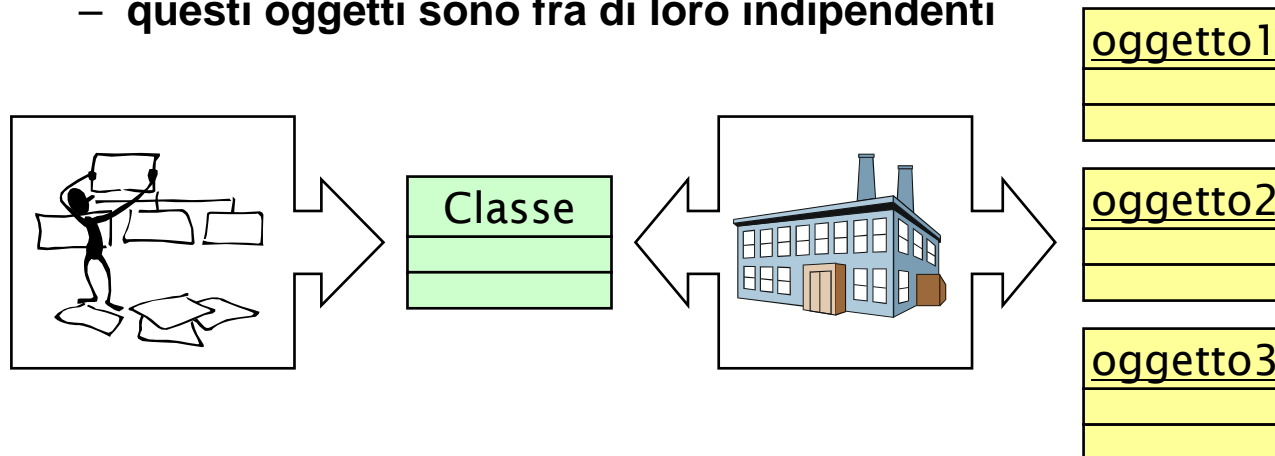
- una classe è il progetto di un singolo oggetto software
- la **definizione** della classe **corrisponde** essenzialmente **anche alla costruzione** dell'oggetto software dalla classe
 - l'aver definito una classe rende disponibile un oggetto software che ha le caratteristiche definite dalla classe



Modalità di costruzione dinamica

□ Nella modalità di costruzione dinamica di oggetti software

- la classe è il progetto di una tipologia di oggetti software
- la **costruzione** di un oggetto software a partire dalla classe **deve essere** solitamente **richiesta in modo esplicito**
 - al momento della definizione della classe non viene costruito nessun oggetto software di quel certo tipo
- è possibile costruire tanti oggetti software di quel tipo
 - questi oggetti sono fra di loro indipendenti



Oggetti

□ Un oggetto software è chiamato

▪ oggetto classe

- se è l'oggetto software **costruito** automaticamente con **modalità statica** a partire dalla definizione di una classe

▪ oggetto istanza

- se è un oggetto software **creato** con **modalità dinamica** da una classe

Nota bene: si parla genericamente (e più semplicemente) di oggetti quando la distinzione tra oggetto classe e oggetto istanza non è necessaria oppure è chiara dal contesto

Componenti di una classe

- Una classe è il progetto di una tipologia di oggetti
 - un oggetto software consiste di **stato** e **comportamento**
- In corrispondenza, la definizione di una classe comprende
 - la dichiarazione di **variabili** — che rappresentano le **proprietà**
 - la definizione di **metodi** — che implementano le **operazioni**

Modalità di costruzione e componenti di una classe

□ Nella modalità di costruzione **statica**

- variabili di classe
- metodi di classe
- blocchi di inizializzazione statica

□ Nella modalità di costruzione **dinamica**

- variabili di istanza
- metodi di istanza
- costruttori – **metodi speciali per** richiedere al calcolatore, quando servono, la **costruzione** di oggetti istanza

La classe Math

□ La classe **Math** segue la modalità di costruzione **statica**

- **Math** è un oggetto che per sua natura è unico
- **Math** è un oggetto classe
- le proprietà di **Math** sono alcune costanti notevoli

Math
static E : double static PI : double
static double sqrt(double n) static double pow(double b, double e) static double log(double n) static double random() ...

«oggetto classe» <u>Math</u>
E = 2.71... PI = 3.14...
double sqrt(double n) double pow(double b, double e) double log(double n) double random() ...

La classe String

- La classe **String** segue la modalità di costruzione **dinamica**
 - un oggetto **String** rappresenta una **stringa**
 - ogni stringa è un oggetto istanza della classe **String**
 - gli oggetti **String** sanno eseguire operazioni utili per la loro gestione facenti parte dell'interfaccia della classe
 - gli oggetti **String** vengono costruiti con il costruttore denominato **String** come la classe

"calcolatore" : String

calcolatore

... le operazioni ...

Cosa abbiamo visto finora

- ❑ Cosa è un programma
- ❑ Cosa significa programmare
- ❑ Cosa è la programmazione ad oggetti
- ❑ Cosa sono gli oggetti software
 - come si progettano
 - come si costruiscono
- ❑ Cosa è una classe
- ❑ Costruzione statica e dinamica di classi
- ❑ Esempi di oggetti software
 - **System.out – Math - String**

Riferimenti al libro di testo

- Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al **capitolo 2 sugli oggetti software**, tenendo conto che
 - il paragrafo **2.3.3 – Un oggetto per la lettura da tastiera** – lo vedremo in seguito e quindi per ora può essere saltato
 - gli argomenti trattati nei seguenti **paragrafi non faranno parte del programma d'esame** e quindi devono essere omessi nello studio del capitolo
 - 2.3.4 – Robot e labirinti
 - 2.6.2 – Le classi Robot e Labirinto