PROGRAMMAZIONE ORIENTATA AGLI OGGETTI

Introduzione

IL FERMENTO DEGLI ANNI ‘70

- Hardware (8080, costi, dimensioni)
- Applicare le tecnologie informatiche ad ambiti più complessi.
- Dati più importanti delle procedure.

- Database
- OOP

ALESSANDRO URSOMANDO

UN NUOVO PARADIGMA

Con la programmazione OOP cambiava il punto di vista:
l'obiettivo non era più *risolvere un problema* ma **gestire un sistema**.

Un sistema fatto di **componenti**.

Componenti che **interagiscono** tra loro.

Componenti caratterizzate da **proprietà** e **azioni**.

Sistema per dipingere

SISTEMA DEL MONDO REALE

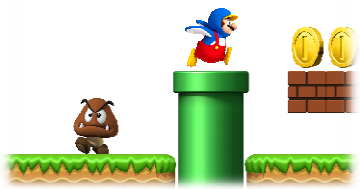
IL CONCETTO DI OGGETTO

Il concetto di oggetto è molto vicino al concetto di record.
L'oggetto – come il record – può avere delle variabili che lo caratterizzano (**proprietà**)
ma può avere – inoltre – delle funzioni (**metodi**).

Un oggetto di tipo *cattivone* può avere le proprietà:
velocità, altezza, aspetto, posizione..

Un oggetto (l'unico) di tipo *personaggio* può avere i metodi: salta, trasformati, cadi, muori, tocca un *cattivone*..

Mario bros



SISTEMA INFORMATICO

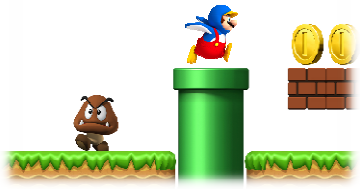
IL CONCETTO DI CLASSE

Con il termine oggetto in realtà si intende una istanza di una **classe**.
La classe quindi è il **tipo** (come intero, stringa..) e **l'oggetto è una istanza della classe**, ovvero una variabile di un tipo definito da noi come classe (per esempio tot, x..).

Un oggetto di ~~tipo~~ **classe** *cattivone* può avere le proprietà: velocità, altezza, aspetto, posizione..

Un oggetto (l'unico) di ~~tipo~~ **classe** *personaggio* può avere i metodi: salta, trasformati, cadi, muori, tocca un *cattivone*..

Mario bros



SISTEMA INFORMATICO

UN ESEMPIO COMPLETO

CaneDiFantasia	
String	**nome**
Color	**colore**
String	**serie**
String	**razza**
Number	**annoNascita**
Number	**puntiVita**
Number	**punti**
void	**siSpostaVerso(Direzione)**
Number	**guadagnaPunti(Number)**
Number	**siFerisce(Number)**
void	**mangia(Number)**
Number	**gioca(Number)**

Cane di fantasia

SISTEMA DEL MONDO REALE

UN ESEMPIO COMPLETO

Dichiaro l'oggetto **cane1** e lo inizializzo come indicato.

cane1		
String	**nome**	Spank
Color	**colore**	bianco
String	**serie**	Hello Spank
String	**razza**	meticcio
Number	**annoNascita**	1980
Number	**puntiVita**	100
Number	**punti**	0

D'ora in avanti posso accedere alle proprietà o ai metodi in questo modo:
- cane1.nome
- cane1.mangia(10)

Cane di fantasia

SISTEMA DEL MONDO REALE

UN ESEMPIO COMPLETO

Dichiaro l'oggetto **cane2** e lo inizializzo come indicato.

cane2		
String	**nome**	Snoopy
Color	**colore**	bianco
String	**serie**	Peanuts
String	**razza**	beagle
Number	**annoNascita**	1950
Number	**puntiVita**	100
Number	**punti**	0

D'ora in avanti posso accedere alle proprietà o ai metodi in questo modo:
- cane2.nome
- cane2.mangia(10)

Cane di fantasia

SISTEMA DEL MONDO REALE

UN ESEMPIO COMPLETO

Dichiaro l'oggetto **cane3** e lo inizializzo come indicato.

cane3		
String	**nome**	Pluto
Color	**colore**	giallo
String	**serie**	Disney
String	**razza**	bracco
Number	**annoNascita**	1930
Number	**puntiVita**	100
Number	**punti**	0

D'ora in avanti posso accedere alle proprietà o ai metodi in questo modo:

- cane3.nome
- cane3.mangia(10)

Cane di fantasia

SISTEMA DEL MONDO REALE

UN ALTRO ESEMPIO COMPLETO

SimCard	
Number	**numTelefono**
Number	**iccid**
Number	**memoria**
Number	**credito**
Number	**annoNascita**
Persona	**intestatario**
Profilo	**profilo**
void	**faTelefonata(Number)**
void	**ricaricaCredito(Number)**
void	**cambiaIntestatario(Persona)**
Number	**memoriaLibera()**

SIM card
(dal punto di vista dell'operatore)

SISTEMA INFORMATICO

UN ALTRO ESEMPIO COMPLETO

faTelefonata(Number) prenderà come parametro il numero del destinatario e modificherà il valore della proprietà credito.

Analogamente gli altri metodi.

SIM card
(dal punto di vista dell'operatore)

SISTEMA INFORMATICO

IL CONCETTO DI COSTRUTTORE

Un oggetto deve **sempre** avere un contenuto **coerente**.

Supponiamo di avere la classe **Data**.

Data	
Number	**giorno**
Number	**mese**
Number	**anno**
String	**getGiornoSettimana()**
void	**aggiungiGiorni(Number)**
void	**stampa()**

Tutti gli attributi di tutti gli oggetti della classe **Data** devono **sempre** avere dei dati validi.

Anche non appena l'oggetto viene istanziato.

Questo lavoro viene svolto dal **costruttore**.

IL CONCETTO DI COSTRUTTORE

Un oggetto deve **sempre** avere un contenuto **coerente**.

Chi sviluppa la classe deve definire un metodo **costruttore** con il quale si dà un valore iniziale agli attributi.

Il costruttore può accettare dei **parametri** oppure può usare dei dati arbitrari.

Il costruttore viene chiamato all'atto della dichiarazione della variabile.

Lo sviluppatore può produrre anche **più costruttori** (che avranno liste di parametri diverse).

IL CONCETTO DI METODI STATICI

Un linguaggio di programmazione che aderisce al paradigma della programmazione strutturata mette a disposizione dei suoi programmatori una serie di strumenti con un meccanismo molto semplice: la **libreria**.

Per esempio una certa libreria **Math** potrebbe contenere funzioni e costanti di uso comune in matematica (per esempio il valore di **pigreco** o la produzione di un numero **pseducasuale**).

Per **libreria** si intende una collezione di funzioni e costanti (relative un certo argomento).

Per fornire simili strumenti ai loro programmatori i linguaggi di programmazione OOP fanno uso dei **metodi statici**.

IL CONCETTO DI METODI STATICI

Per metodo **statico** (o metodo **di classe**) si intende un metodo che non è relativo ad un oggetto ma – appunto – ad una classe.

Pertanto un metodo statico può essere invocato senza la necessità di istanziare alcunché: basta specificare il nome della classe seguita dal nome del metodo.

METODO STATICO

Proseguendo nell'esempio precedente possiamo supporre l'esistenza di una classe **Math** all'interno della quale ci siano una costante che contiene il valore di pigreco e una funzione in grado di produrre un valore pseudocasuale.

L'utilizzo di questi due strumenti avverrà senza istanziare alcun oggetto, ma con istruzioni simili alle seguenti:

- Math.PI
- Math.random()

CARATTERISTICHE DELLA OOP

Indica la capacità di racchiudere all'interno dell'oggetto sia gli attributi che i metodi

INCAPSULAMENTO

Indica la possibilità di sviluppare oggetti complessi a partire da oggetti già sviluppati più semplici

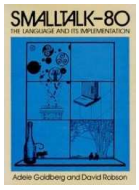
EREDITARIETÀ

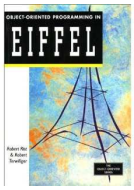
Indica la possibilità di richiamare metodi che a run-time potranno avere implementazioni diverse

POLIMORFISMO

LINGUAGGI PURI E IBRIDI

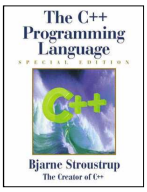
Sono quei linguaggi all'interno dei quali ogni cosa è un oggetto.





LINGUAGGI PURI

Sono quei linguaggi che offrono anche tipi di dati semplici.



LINGUAGGI IBRIDI