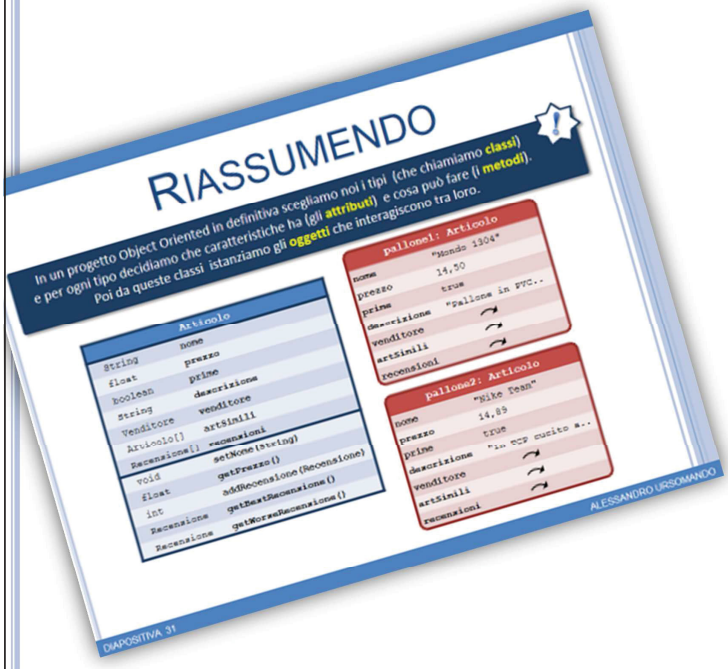
OOP

UML

UNIFIED MODELING LANGUAGE



UML è un **formalismo grafico**
molto potente
che permette di rappresentare
sia gli **aspetti statici**
che gli **aspetti dinamici**
di un **sistema**

Tra gli aspetti statici
ci sono
i diagrammi delle classi e
i diagrammi degli oggetti.

DIAGRAMMA UML DI UNA CLASSE

Data	
- int	giorno
- int	mese
- int	anno
+	Data()
+	Data(int,int,int)
+ void	setData(int,int,int)
+ void	setGiorno(int)
+ void	setMese(int)
+ void	setAnno(int)
+ int	getGiorno()
+ int	getMese()
+ int	getAnno()
+ boolean	isMaggiore(Data)
+ int	differenzaInGiorni(Data)
+ String	getDataComeStringa()
+ void	aggiungiUnGiorno()
- boolean	annoBisestile()
- int	getUltimoGiorno()
+ boolean	isUguale(Data)

Ci sono infiniti modi
per rappresentare una classe
con un diagramma UML.
A noi basterà questo.

- ❑ Tre zone
- ❑ Specificatore di visibilità
- ❑ Il tipo
- ❑ Il nome dell'attributo o del metodo
- ❑ L'elenco dei parametri (solo il tipo)

ALESSANDRO URSOMANDO

DIAGRAMMA UML DI UN OGGETTO

Data	
- int	giorno
- int	mese
- int	anno
+	Data()
+	Data(int,int,int)
+ void	setData(int,int,int)
+ void	setGiorno(int)
+ void	setMese(int)
+ void	setAnno(int)
+ int	getGiorno()
+ int	getMese()
+ int	getAnno()
+ boolean	isMaggiore(Data)
+ int	differenzaInGiorni(Data)
+ String	getDataComeStringa()
+ void	aggiungiUnGiorno()
- boolean	annoBisestile()
- int	getUltimoGiorno()
+ boolean	isUguale(Data)

Ha l'obiettivo di mostrare
solo il nome dell'oggetto,
il nome della classe
e lo stato attuale dei suoi attributi

dataDiNascita: Data	
giorno	1
mese	4
anno	1976

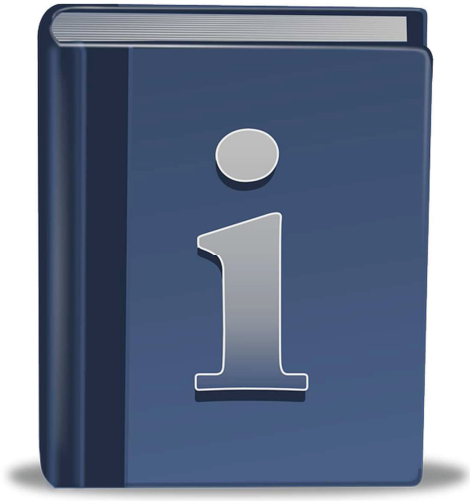
ALESSANDRO URSOMANDO

OOP

Interfaccia
Information hiding
Incapsulamento

INTERFACCIA

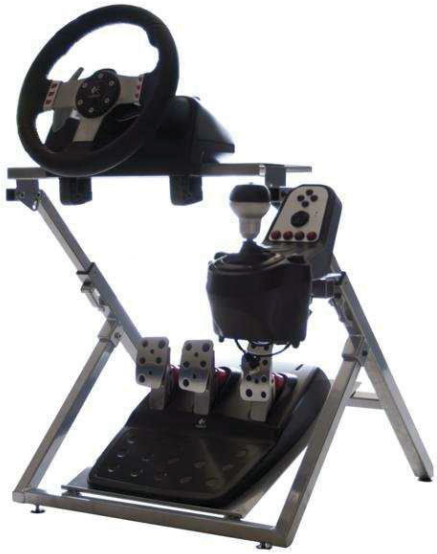
Rappresenta il libretto di istruzioni
che l'utente della classe (il programmatore) deve conoscere per poter interagire con essa.



INTERFACCIA (DELLA CLASSE)

Rappresenta il libretto di istruzioni
che l'utente della classe (il programmatore) deve conoscere per poter interagire con essa.

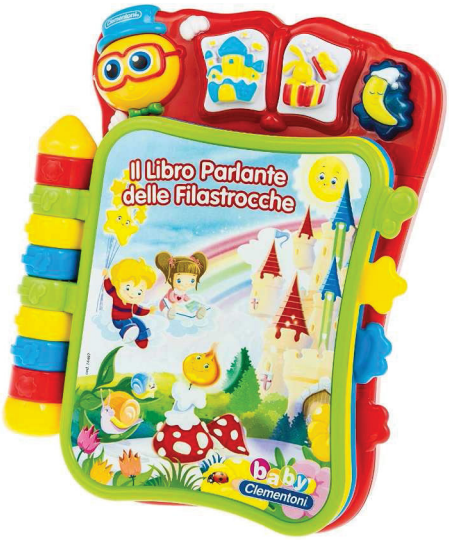
In altre parole,
tutto ciò che è **pubblico**
(ovvero disponibile all'esterno)
rappresenta **l'interfaccia**.



INFORMATION HIDING

Abbiamo il duplice obiettivo di
proteggere il nostro oggetto da un uso incauto del programmatore
e di garantire al programmatore che tutto funzioni bene senza che lui debba occuparsi di nulla.

In altre parole,
con **information hiding**
intendiamo la condizione
con la quale i dettagli implementativi
(dei metodi)
sono nascosti all'utente



INCAPSULAMENTO

Vogliamo esprimere il concetto di
più elementi
visti come una unità indivisibile

L'**incapsulamento**
è la prima delle tre caratteristiche
fondamentali della programmazione OOP.

Mette l'accento sul fatto che
la classe è un contenitore atomico
di metodi e attributi.

OOP

Metodi speciali: toString

L'OGGETTO COME STRINGA

Abbiamo già visto nei nostri esercizi che è comodo avere un metodo che ci fa un riassunto in formato stringa di un oggetto.

Persona	
- String	nome
- String	cognome
- Data	dataDiNascita
...	
+ String	getPersonaComeStringa()
...	

```
// istanzio due oggetti di tipo Persona
Persona a, b;
a = new Persona();
b = new Persona();

// riempio a e b in qualche modo
// ...

// mostro i due oggetti
System.out.println(a.getPersonaComeStringa());
System.out.println(b.getPersonaComeStringa());
```

```
Alessandro Ursomando (41)
Vasco Rossi (65)
```


L'OGGETTO COME STRINGA

Sarebbe comodo poter scrivere direttamente:

```
System.out.println(a);
System.out.println(b);
```

Ma questo darebbe in output un'informazione relativa all'indirizzo dell'oggetto:

```
Persona@3961d1d6
Persona@409d5502
```

```
// istanzio due oggetti di tipo Persona
Persona a, b;
a = new Persona();
b = new Persona();

// riempio a e b in qualche modo
// ...

// mostro i due oggetti
System.out.println(a.getPersonaComeStringa());
System.out.println(b.getPersonaComeStringa());
```

```
Alessandro Ursomando (41)
Vasco Rossi (65)
```

Per fare in modo di ottenere una stringa stabilita da noi andiamo ad implementare il metodo speciale **toString()** al posto di getPersonaComeStringa()

IL METODO toString()

```java
public String toString() {
    // recupero la data di oggi
    // (in qualche modo)

    // calcolo età (in qualche modo)
    int eta;

    // restituisco la stringa creata
    return nome + " " + cognome + " (" + eta + ")";
}
```

DENTRO LA CLASSE

```java
Persona a = new Persona();
System.out.println(a);
```

ESTERNAMENTE ALLA CLASSE

```
Alessandro Ursomando (41)
```

OUTPUT