

Università degli Studi di Trieste

Corso di Laurea Magistrale in
INGEGNERIA CLINICA

RICHIAMI DI PROGRAMMAZIONE A OGGETTI

Corso di Informatica Medica
Docente Sara Renata Francesca MARCEGLIA



Dipartimento di Ingegneria e Architettura



**UNIVERSITÀ
DEGLI STUDI DI TRIESTE**

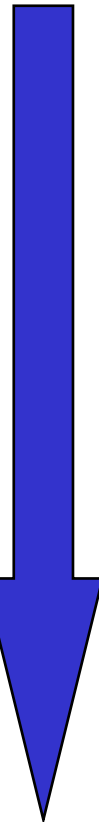
PROGRAMMAZIONE ALGORITMICA



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Algoritmo

Dati di input →



Istruzioni



Output = Risultati

PROGRAMMAZIONE ALGORITMICA (2)



DATO UN VETTORE IN INGRESSO

[2 77 1 935 11 19 773 15 3]



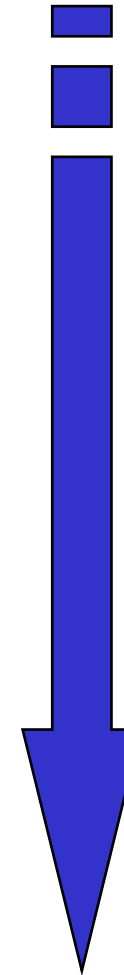
VOGLIO IN USCITA IL VETTORE ORDINATO



Cerco il minimo
Lo metto da parte
Lo elimino dal vettore
Cerco il minimo nel vettore rimanente



VETTORE IN USCITA
[1 2 3 11 15 19 77 773 935]



PROGRAMMAZIONE NON STRUTTURATA

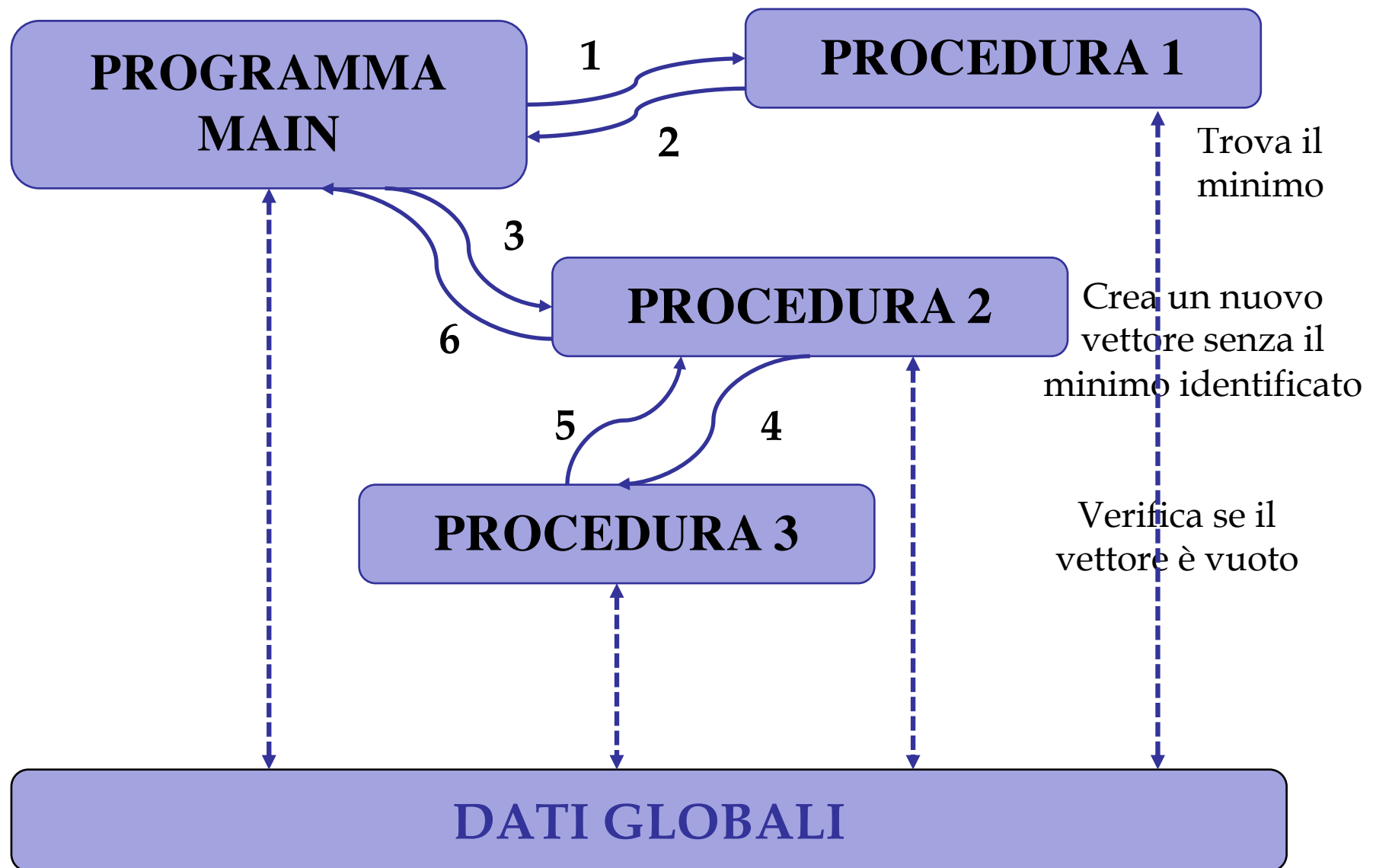


UNIVERSITÀ
DEGLI STUDI DI TRIESTE

**STRUTTURE
DATI**

**PROGRAMMA
MAIN**

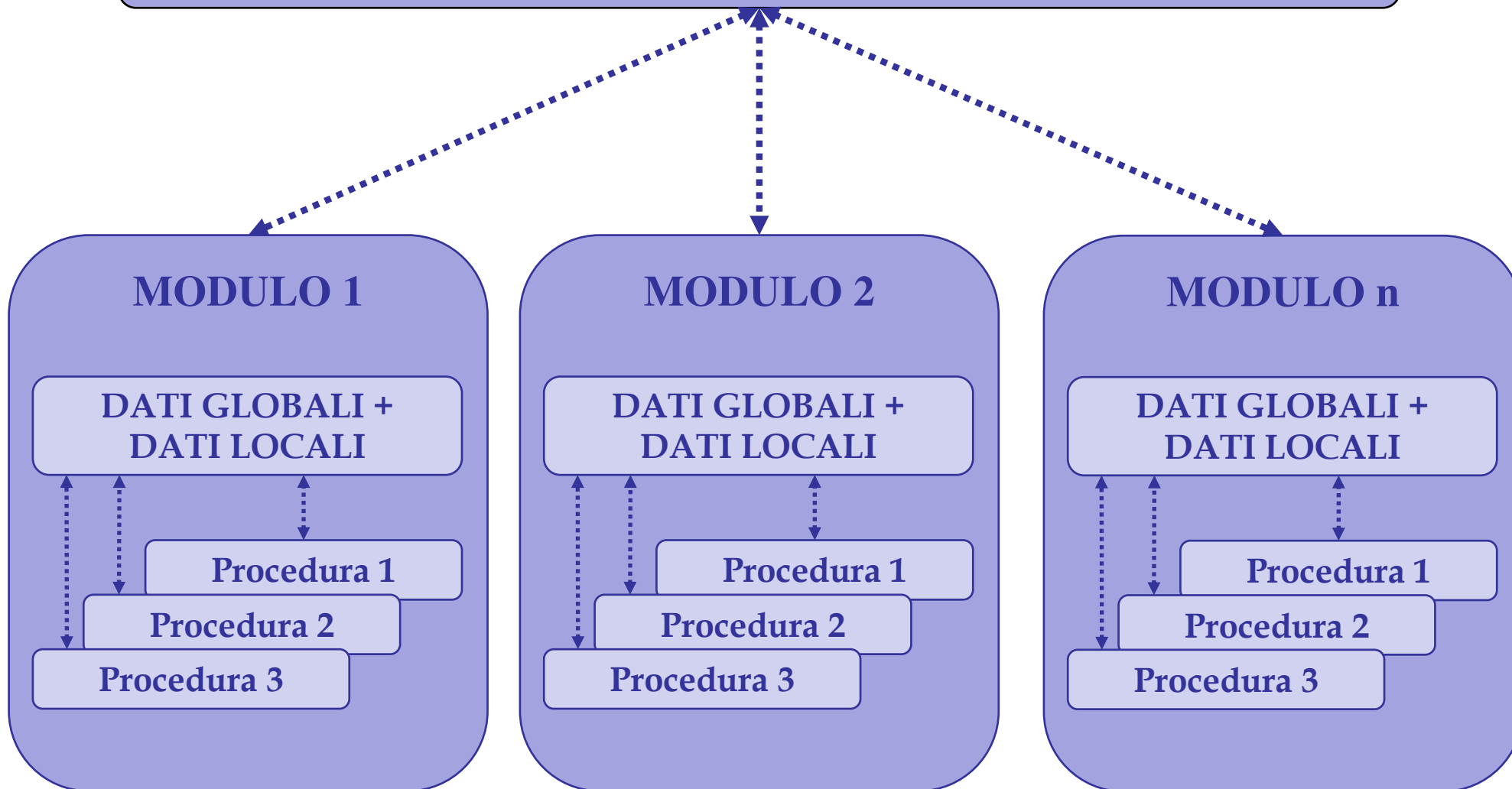
PROGRAMMAZIONE PROCEDURALE



PROGRAMMAZIONE MODULARE



MAIN + DATI GLOBALI

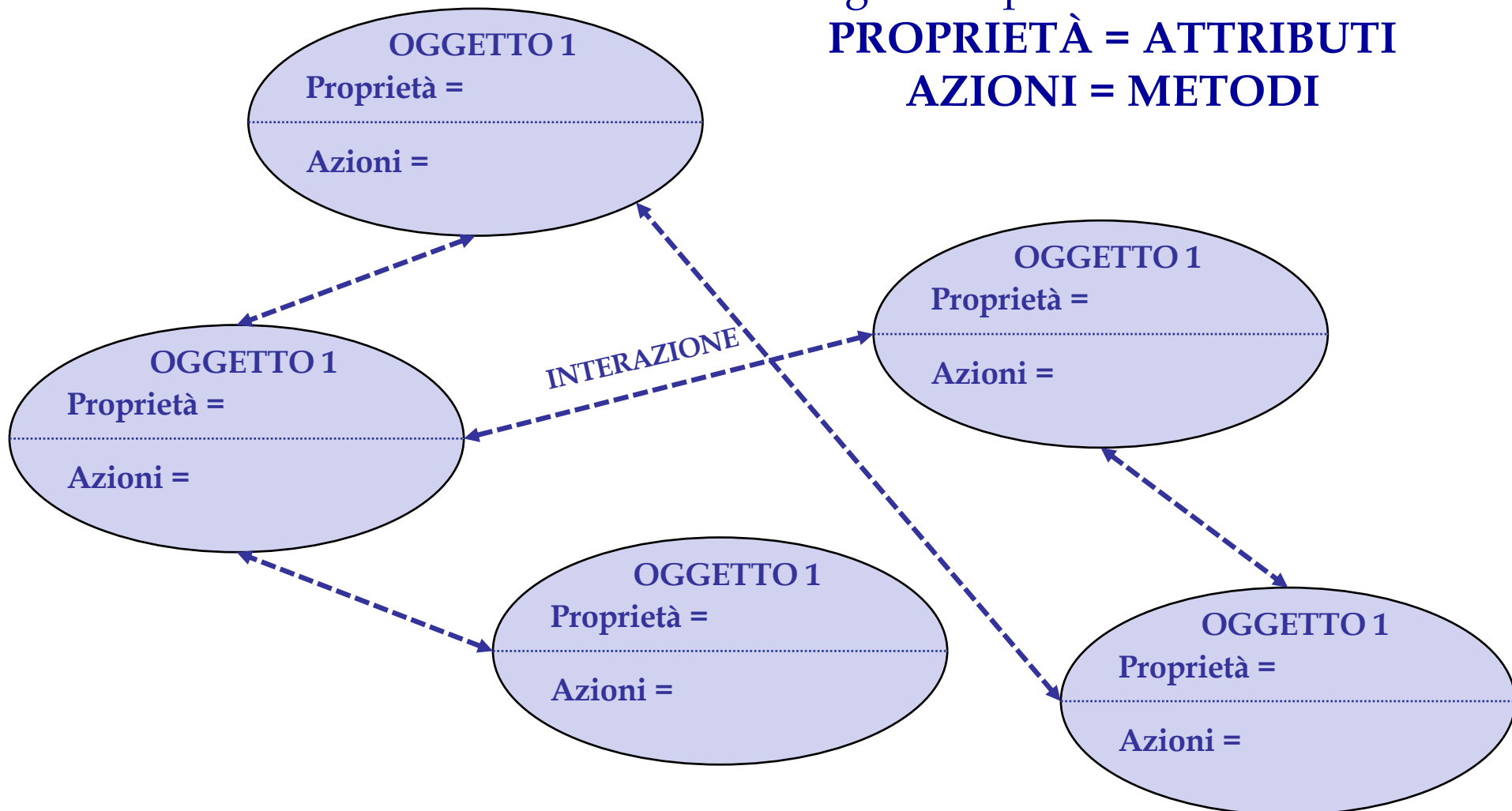




PROGRAMMAZIONE A OGGETTI

SISTEMA

Un sistema è un insieme di
entità interagenti = OGGETTI
in cui ogni componente è caratterizzato da
PROPRIETÀ = ATTRIBUTI
AZIONI = METODI





ALGORITMI vs OGGETTI

PROGRAMMAZIONE ALGORITMICA

- Sequenza di azioni
- Basato su DATI e FUNZIONI = PROGRAMMI
- Obiettivo: risolvere un PROBLEMA

PROGRAMMAZIONE A OGGETTI

- Sistema = Insieme di oggetti
- Basato su OGGETTI fatti da AZIONI e ATTRIBUTI
- Obiettivo: gestire un SISTEMA

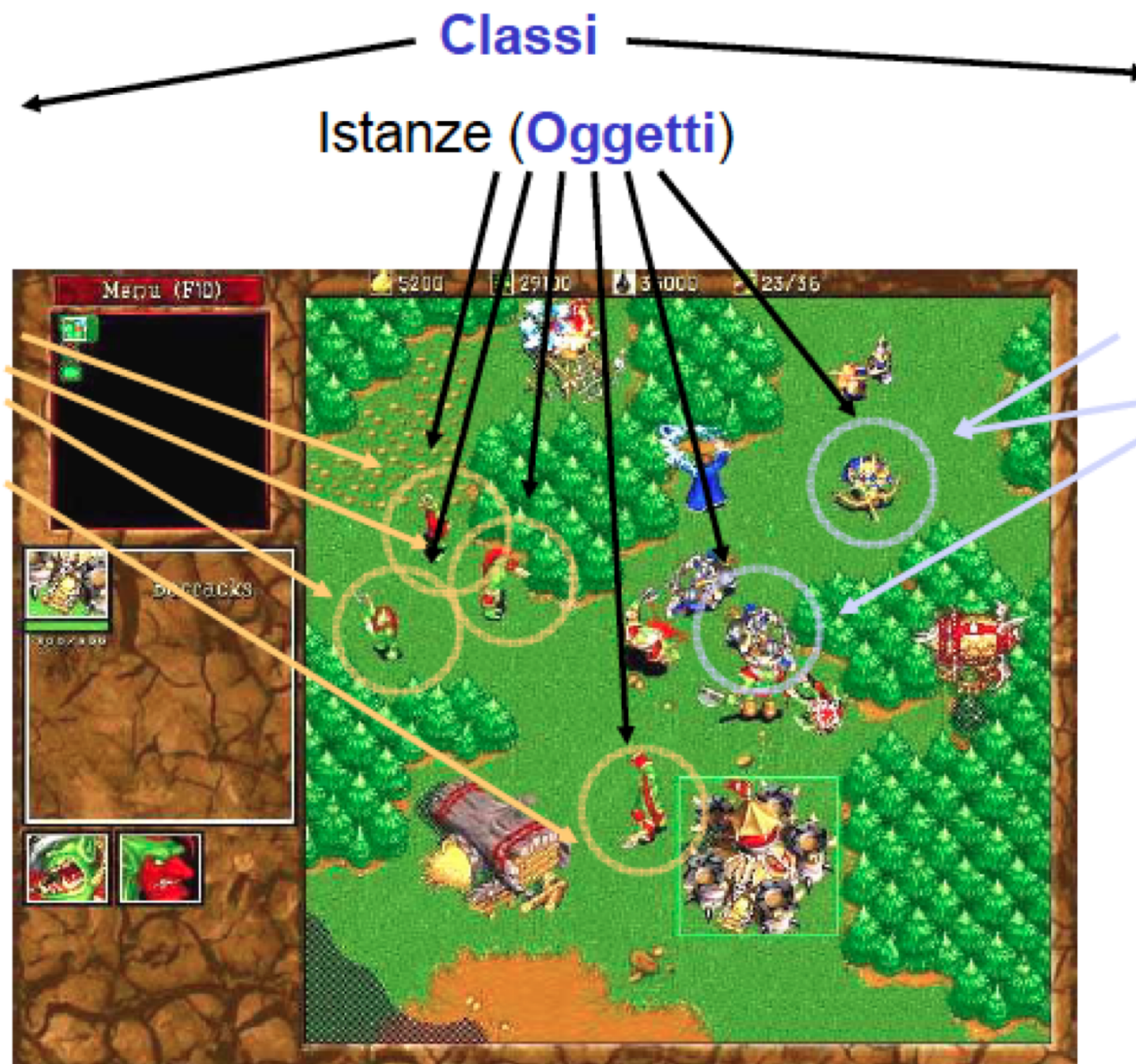
OGGETTI E CLASSI



430 1050 700 200 IRON AGE Menu Chat Diplomacy

Phalanx
120/120
Attack 19
Armor 5
Range 0

OGGETTI E CLASSI





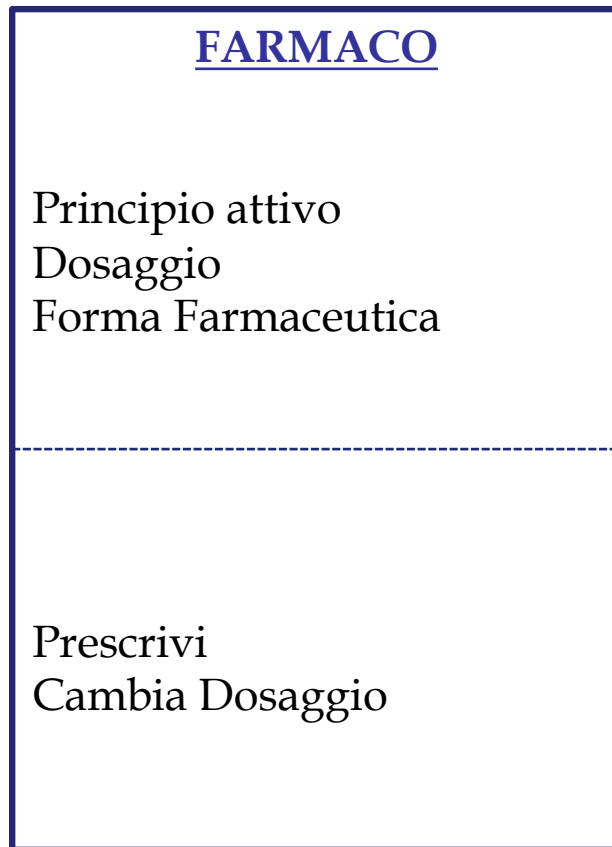
ESEMPIO

OGGETTI

CLASSE

ATTRIBUTI =
Caratteristiche

METODI =
Comportamenti



**OGGETTO =
ISTANZA DI UNA CLASSE**

MOMENT

Principio Att = Ibuprofene
Dosaggio = 200 mg
Forma = Compresse

Prescrivi
Cambia Dosaggio

TACHIPIRINA

Principio Att = Paracetamolo
Dosaggio = 1000 mg
Forma = Compresse effervescenti

Prescrivi
Cambia Dosaggio

ESEMPIO: OSSERVAZIONI



MOMENT

Principio Att = Ibuprofene
Dosaggio = 200 mg
Forma = Compresse

Prescrivi
Cambia Dosaggio

I VALORI DEGLI ATTRIBUTI
SONO SPECIFICI
DELL'OGGETTO ISTANZIATO
(ogni oggetto ha il suo insieme di
valori)

LE AZIONI SONO COMUNI A
TUTTE LE ISTANZE DELLA
CLASSE



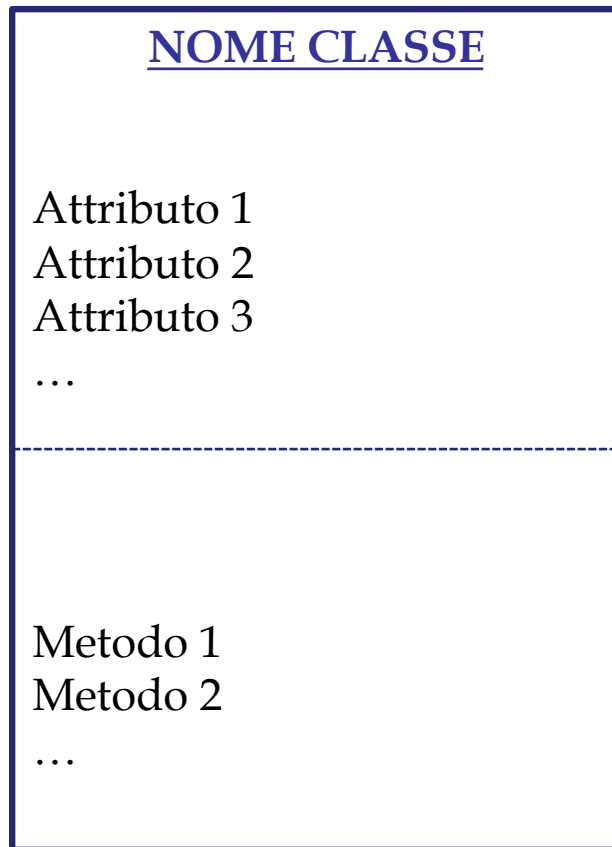
CLASSI E OGGETTI: DEFINIZIONE

OGGETTI

CLASSE

ATTRIBUTI =
Caratteristiche

METODI =
Comportamenti



**OGGETTO =
ISTANZA DI UNA CLASSE**

OGGETTO 1

Att 1 = Val 1
Att 2 = Val 2
Att 3 = Val 3

Metodo 1
Metodo 2

OGGETTO 2

Att 1 = Val 4
Att 2 = Val 5
Att 3 = Val 6

Metodo 1
Metodo 2



ATTRIBUTI E METODI

ATTRIBUTI

- Descrivono le proprietà **statiche** dell'oggetto
- Nella **programmazione** gli attributi vengono realizzati attraverso l'uso delle **variabili** utilizzate dall'oggetto per memorizzare i dati

METODI

- Descrivono le proprietà **dinamiche** dell'oggetto
- Nella **programmazione** i metodi vengono realizzati attraverso la scrittura di codice (**procedure e funzioni**) che implementano le operazioni dell'oggetto

PROPRIETÀ DELLE CLASSI



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

- **Ereditarietà** e **Gerarchia** *di ereditarietà*
- **Tipi** *di ereditarietà*
- **Polimorfismo**

Classe genitrice

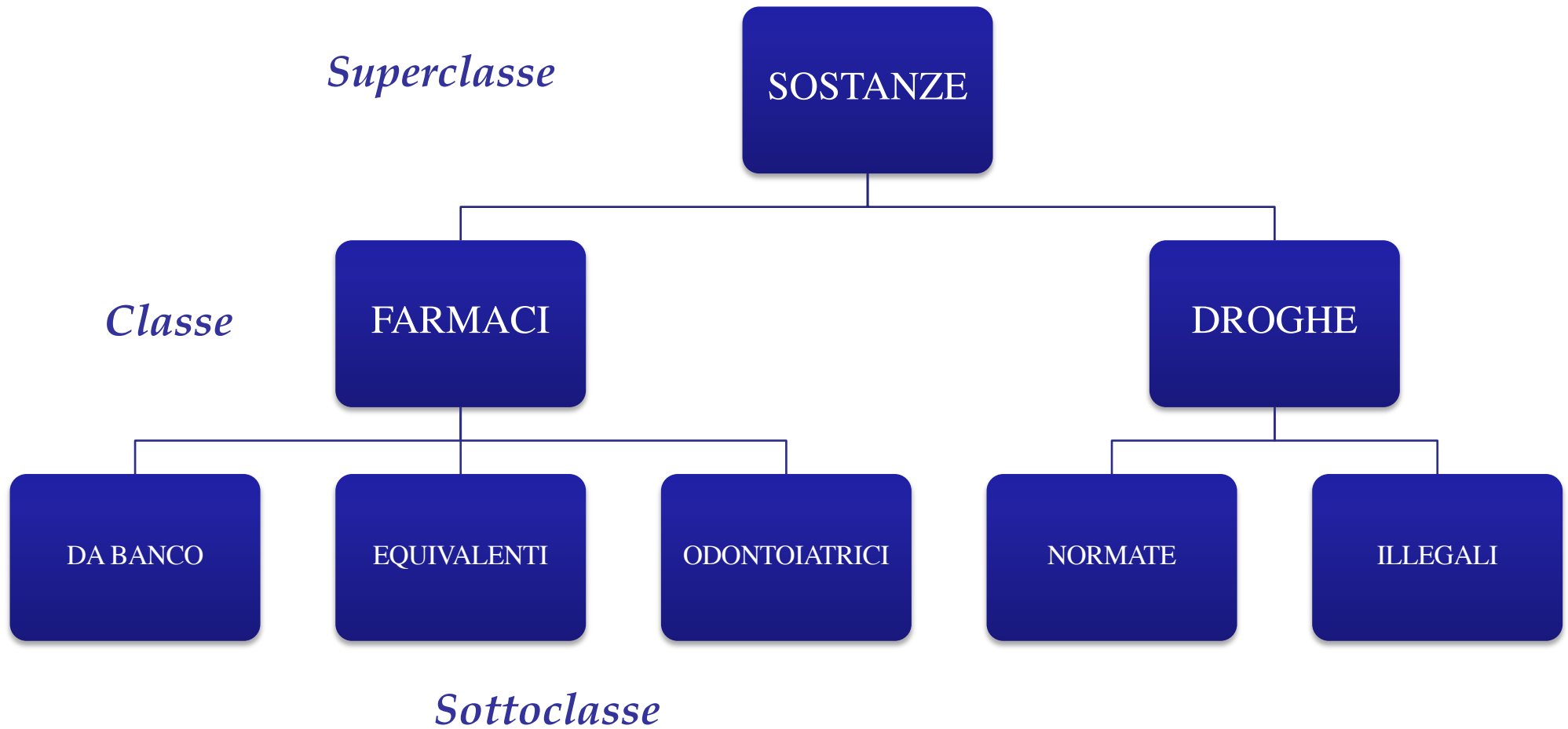
<i>farmaco</i>
Principio attivo
Dosaggio
Forma Farmaceutica
Costo SSN
Prescrivi
Cambia Dosaggio



Nuova Classe

<i>Farmaco di marca</i>
Principio attivo
Dosaggio
Forma Farmaceutica
<i>Nome Commerciale</i>
Costo SSN
<i>Prezzo Pubblico</i>
Prescrivi
Cambia Dosaggio
<i>Calcola costo paziente</i>

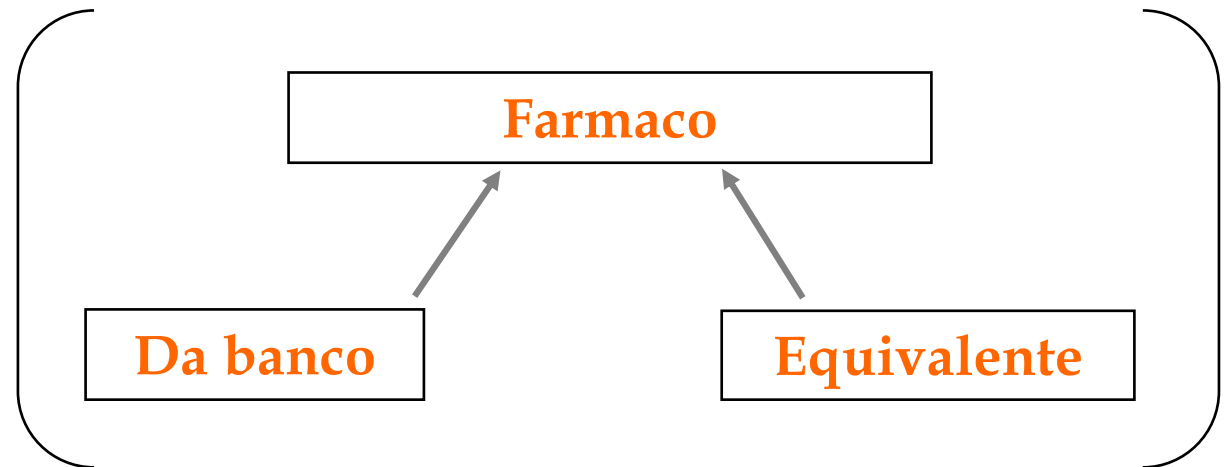
GERARCHIE



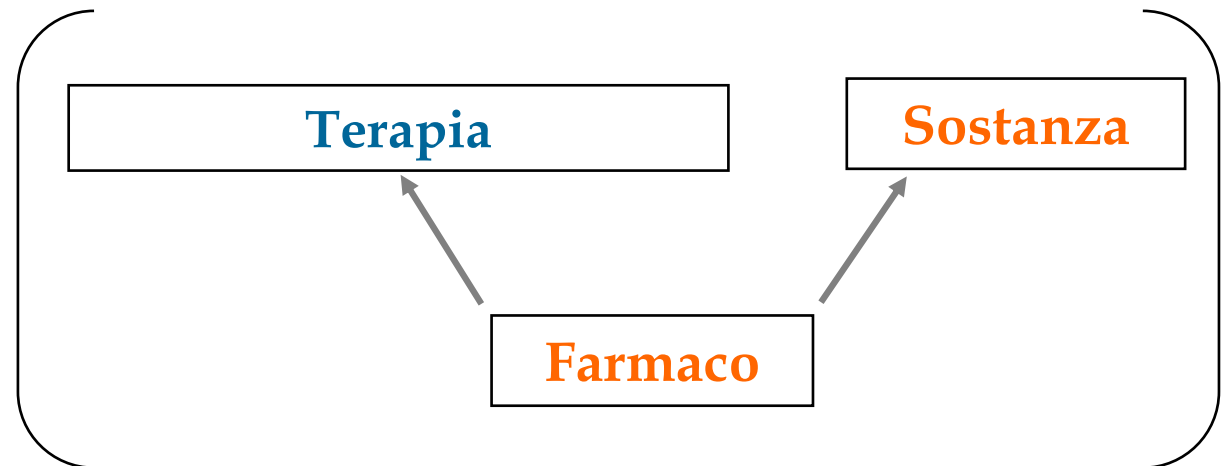
TIPI DI EREDITARETÀ



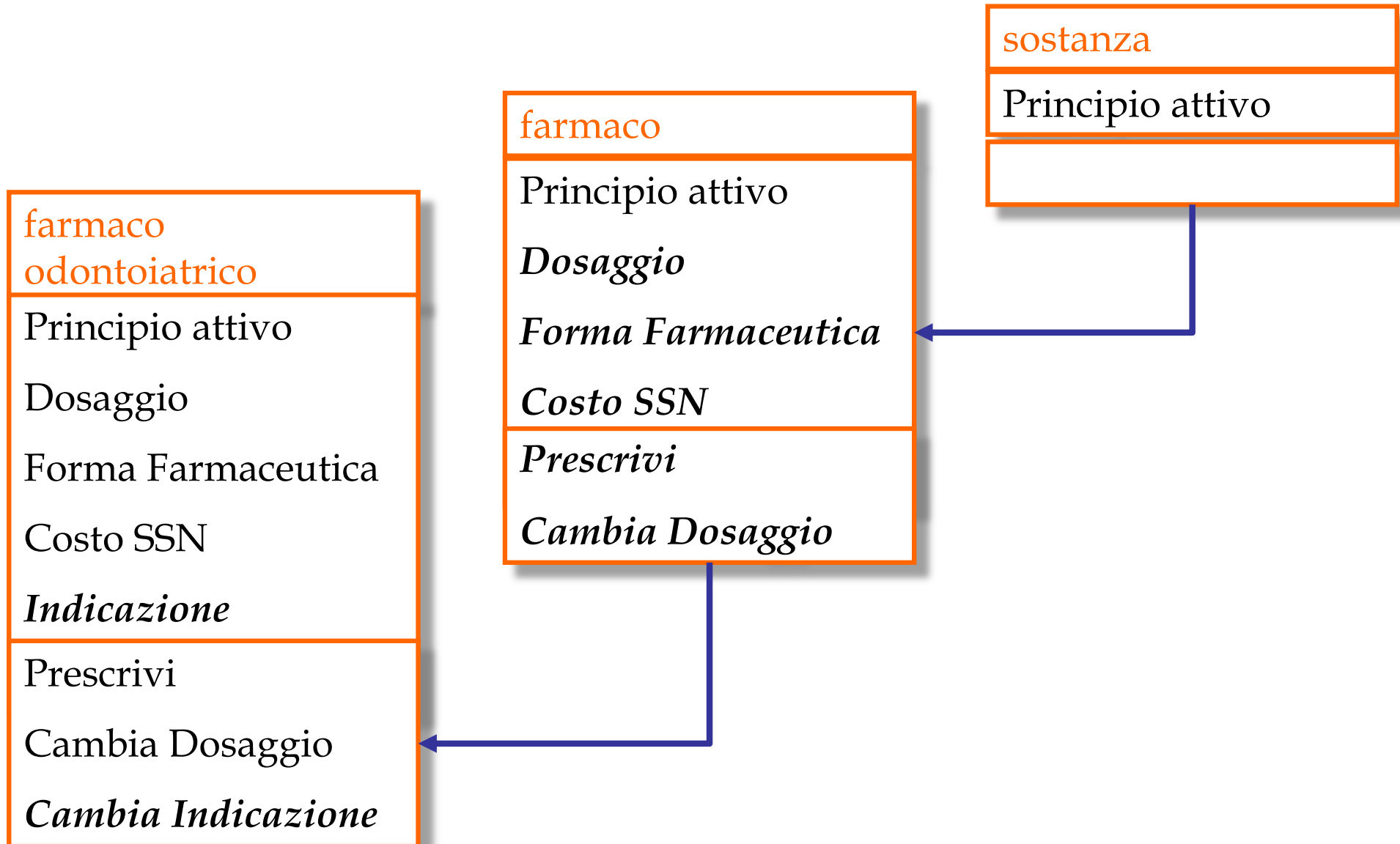
Ereditarietà
singola



Ereditarietà
multipla



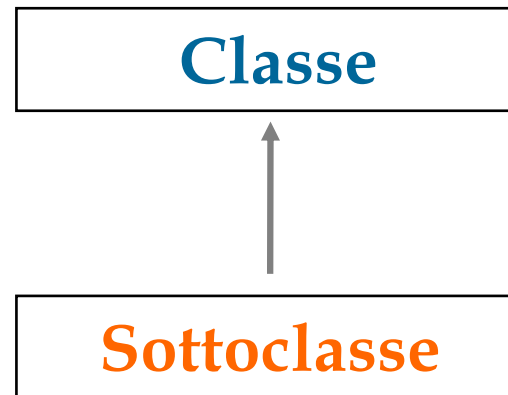
ESEMPIO



PROPRIETÀ DELL'EREDITARIETÀ



UNIVERSITÀ
DEGLI STUDI DI TRIESTE



ESTENSIONE

la sottoclasse
AGGIUNGE NUOVI
METODI/ATTRIBUTI

RIDEFINIZIONE

la sottoclasse
RIDEFINISCE I
METODI

OVERRIDING = riscrittura
del codice del metodo



POLIMORFISMO

OVERRIDING = i *metodi* possono *assumere forme diverse* (cioè *implementazioni diverse*) all'interno della gerarchia delle classi

ES. IL METODO "PRESCRIVI" SARÀ IMPLEMENTATO DIVERSAMENTE NEL FARMACO DA BANCO E NEL FARMACO GENERICO

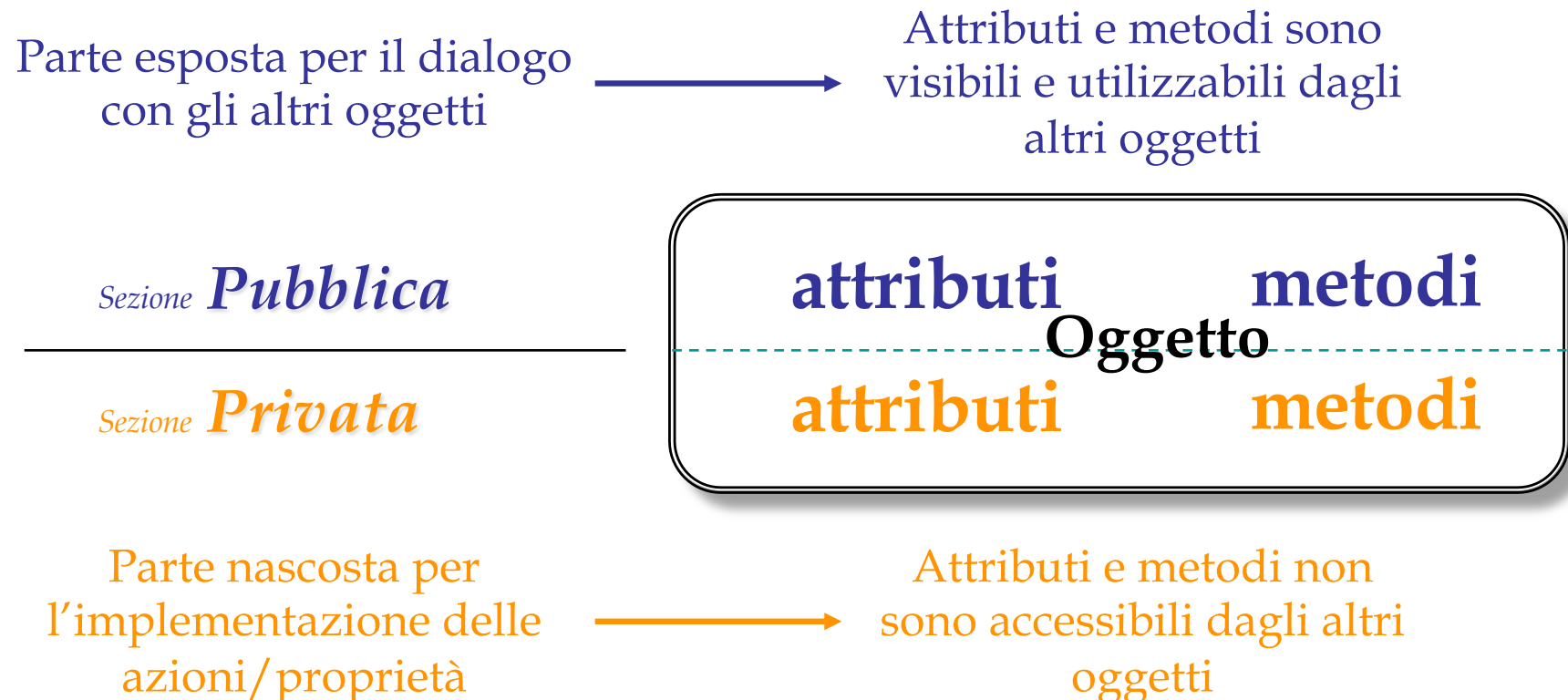
OVERLOADING = i *metodi* possono *assumere forme diverse* (cioè *implementazioni diverse*) all'interno della stessa classe

ES. IL METODO "CAMBIA DOSAGGIO" PUÒ RICHIEDERE O DI CAMBIARE IL NUMERO DI ASSUNZIONI O DI CAMBIARE IL NUMERO DI DOSI PER ASSUNZIONE



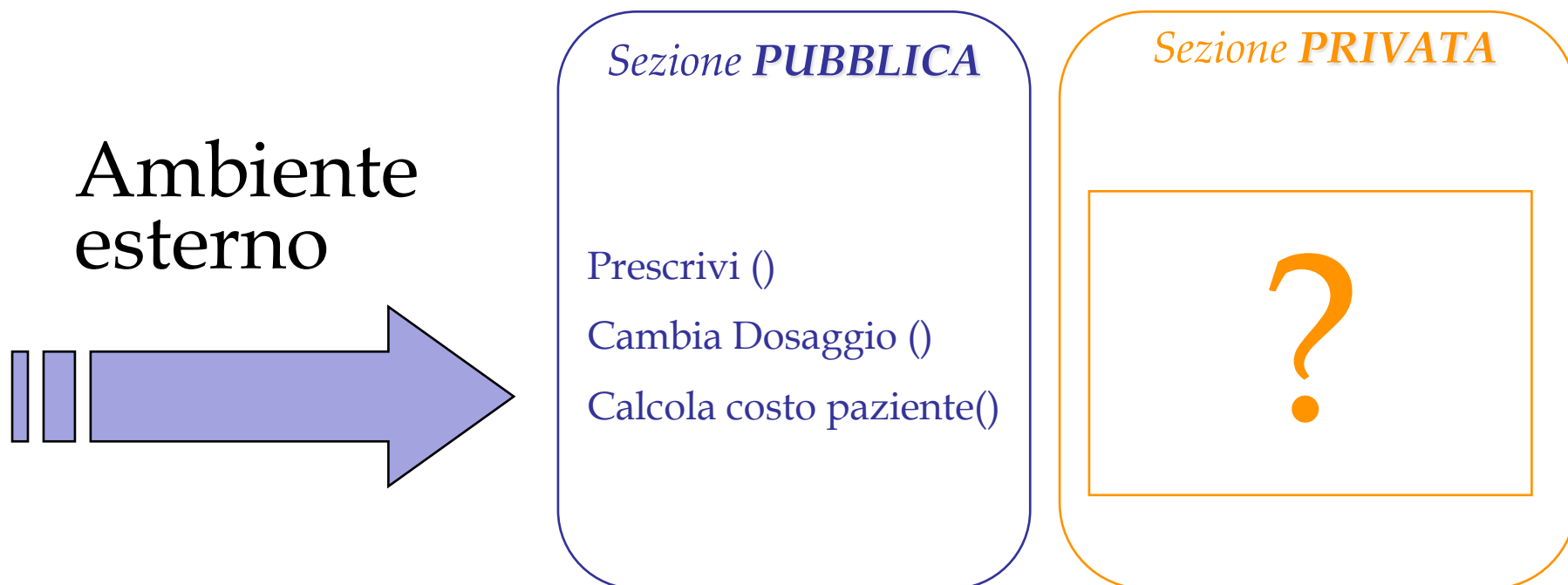
INCAPSULAMENTO

- **Incapsulamento** = Proprietà dell'oggetto di incorporare al suo interno attributi e metodi
- **Information hiding** = mascheramento dell'informazione all'interno dell'oggetto → espone solo metodi e attributi della sezione pubblica



INTERFACCIA

- **INTERFACCIA** = insieme dei messaggi inviabili all'oggetto/ricevibili dall'oggetto
- L'interfaccia **non consente di vedere come sono implementati i metodi**, ma ne permette il loro utilizzo e l'accesso agli attributi pubblici





INTERAZIONE TRA OGGETTI: MESSAGGI

- Un programma ad oggetti è caratterizzato dalla presenza di tanti oggetti che **interagiscono fra loro attraverso il meccanismo dello scambio di messaggi**
- I messaggi possono:
 - Richiedere un'informazione su un oggetto
 - Modificare lo stato di un oggetto

Dott Rossi

Data Nascita = 3/1/69
Matricola = 12345
Specialità = **neurologo**
Ospedale = **maggiore**

Sinemet

Principio attivo = levodopa
Dosaggio = 150 mg
Forma Farmaceutica = compresse

Messaggio:

`Sinemet.prescrivi()`

Metodo:

`prescrivi ()`





DEVELOPMENT CHOICE

Stress full capabilities
of the device

Better performances

Compatibility with upgrade

Native

Specific for a device

Needs of knowledge for the
adopted development language

Low reusability for common
libraries/functionalities

Based on a standard development
Language (HTML5) and libraries

Portable on different architecture

Could be compiled or interpreted by
The web browser on the device

WebApp?

Need to recompile/package

Not stressing the full capabilities
of the physical device

Not full control over the
API/primitives used in the App

or

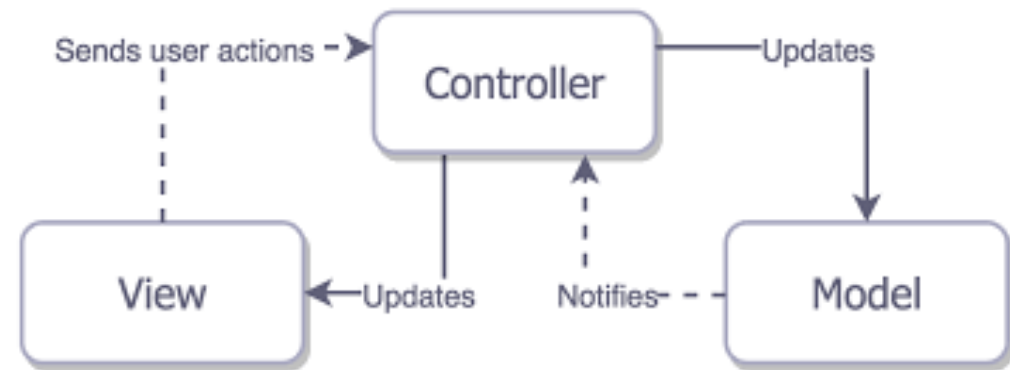




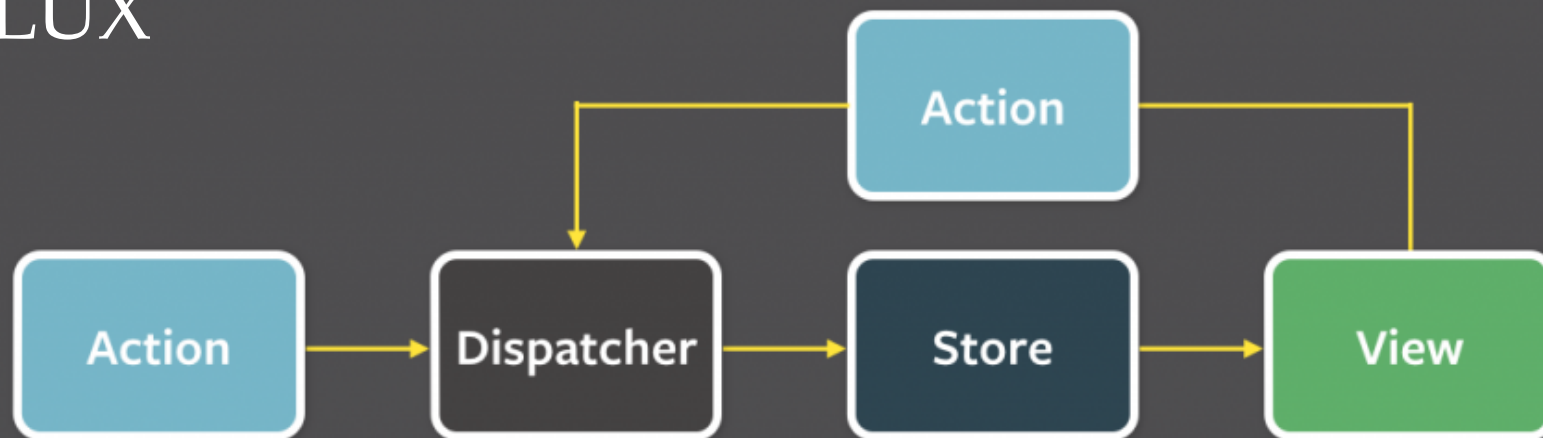
ARCHITECTURAL PATTERNS

MVC

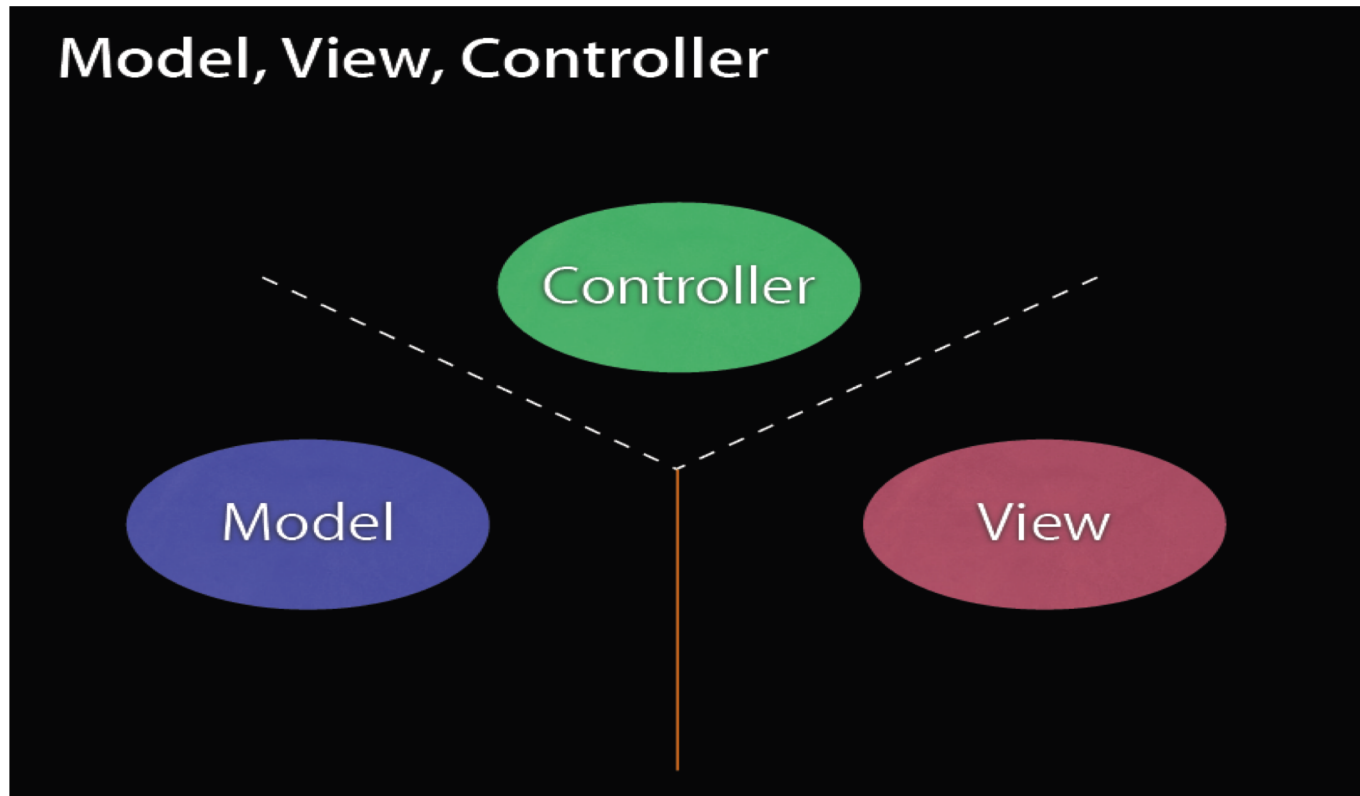
Set of structures necessary to reason about the system, which includes the software elements, the relationships between them, the properties they have.



FLUX



THE MVC PATTERN



View : the **UI interface** of the pattern

Model : the **application logic**

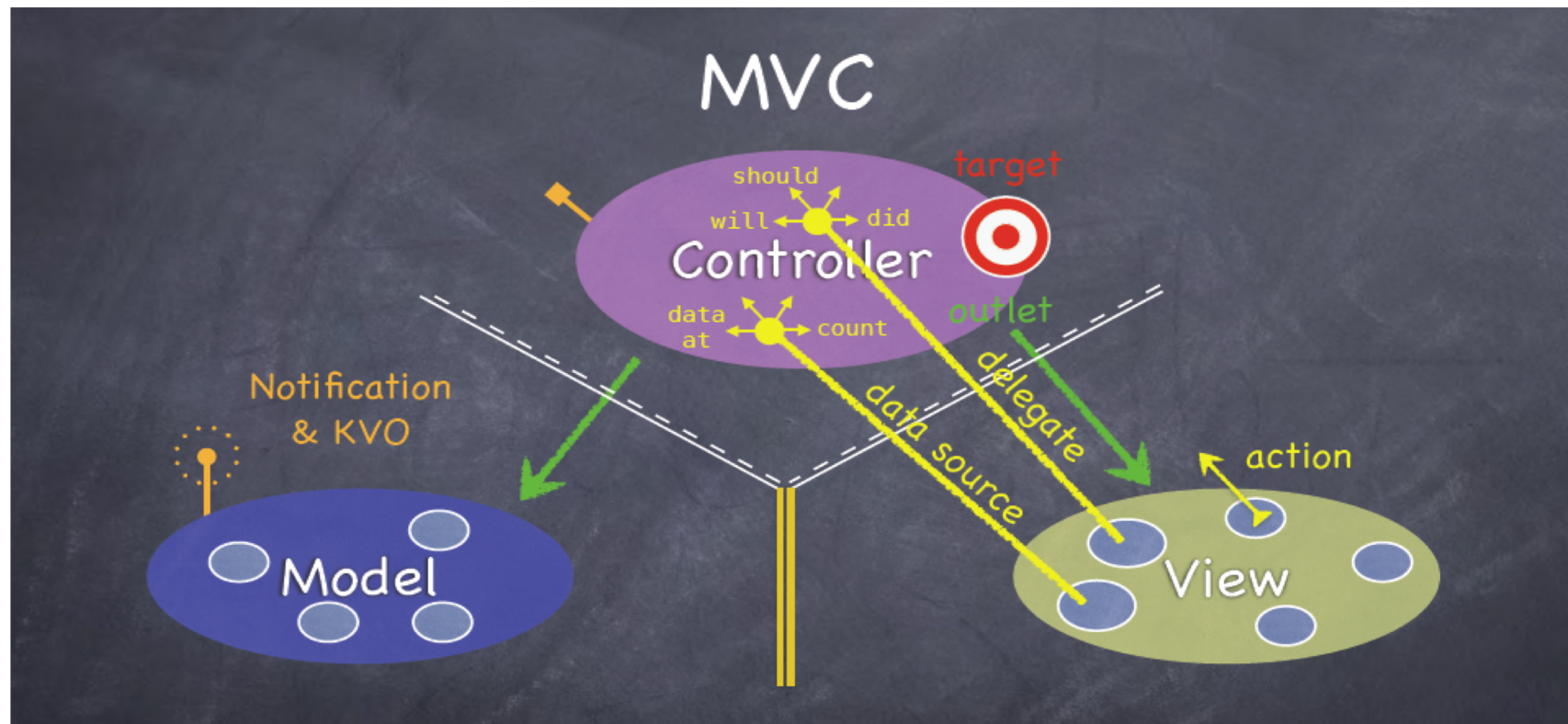
Controller : the “source” of the data for model and view. It takes care of the **internal communication** and provide the **external coordination**. It could handle more than one Model/View

THE MVC PATTERN IN iOS

One big difference with SW development for the pattern:

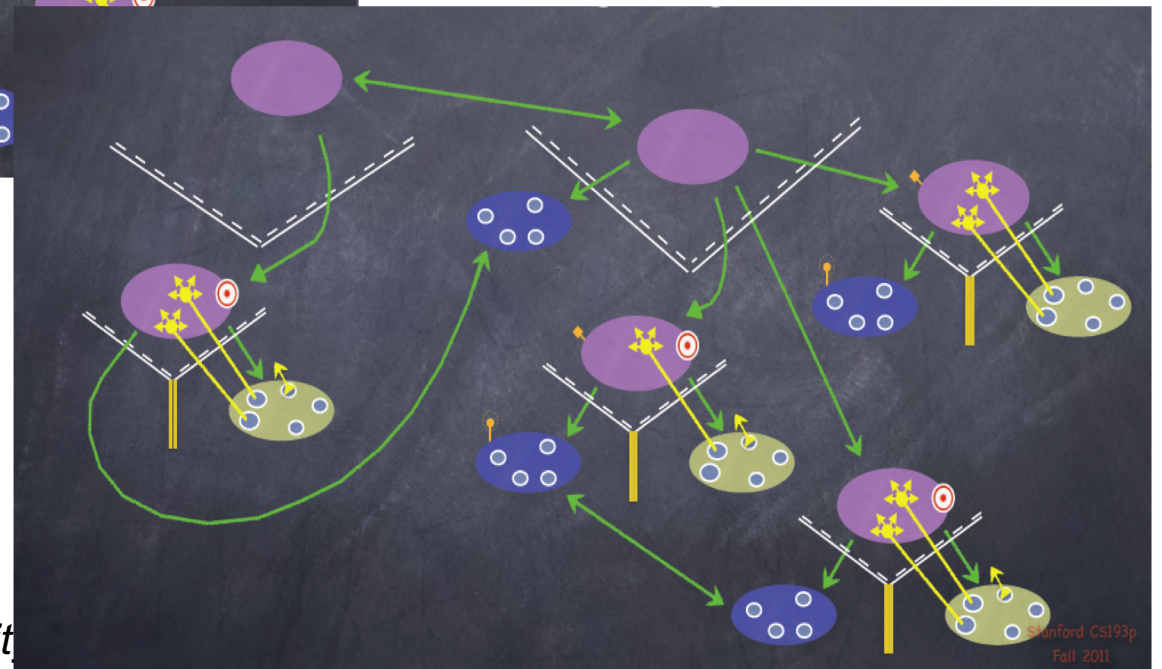
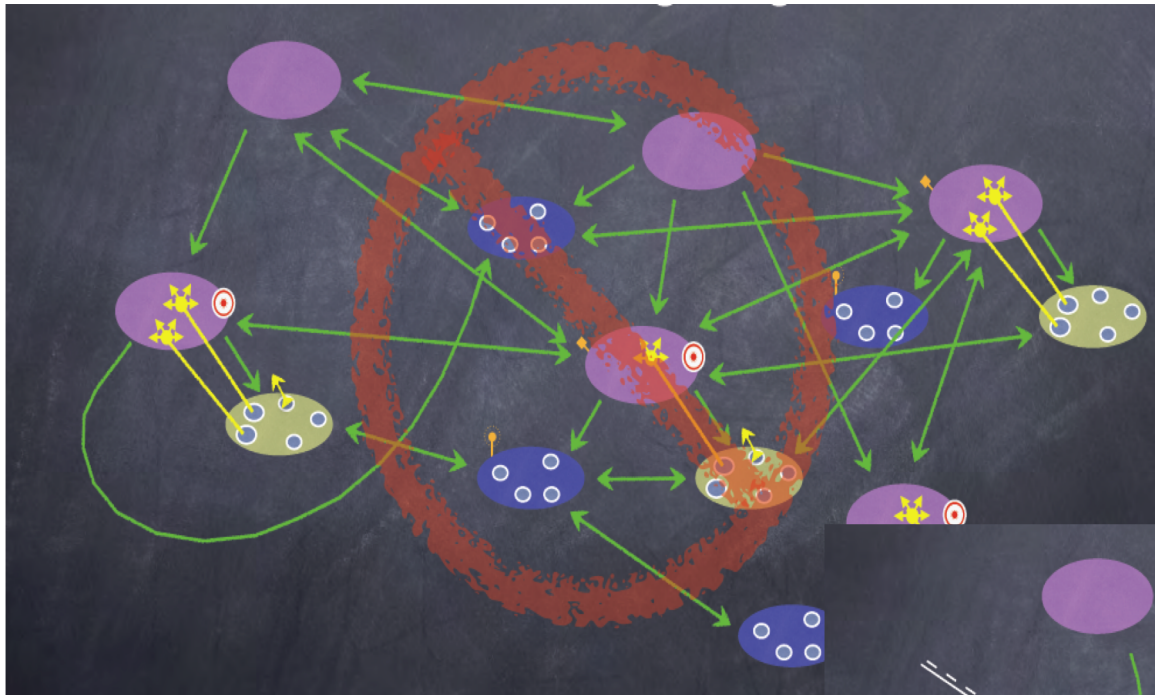
SW → driver is the application logic (functionalities offered)

Apps → driver are UI / available devices and
the possible interactions



By Paul Hegarty, Stanford University

THE MVC PATTERN IN iOS



By Paul Hegarty, Stanford University