

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**

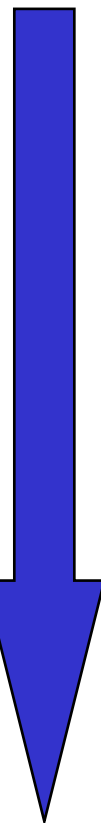
PROGRAMMAZIONE ALGORITMICA



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]



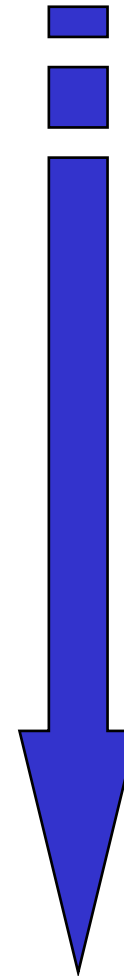
VOGLIO IN USCITA IL VETTORE ORDINATO



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

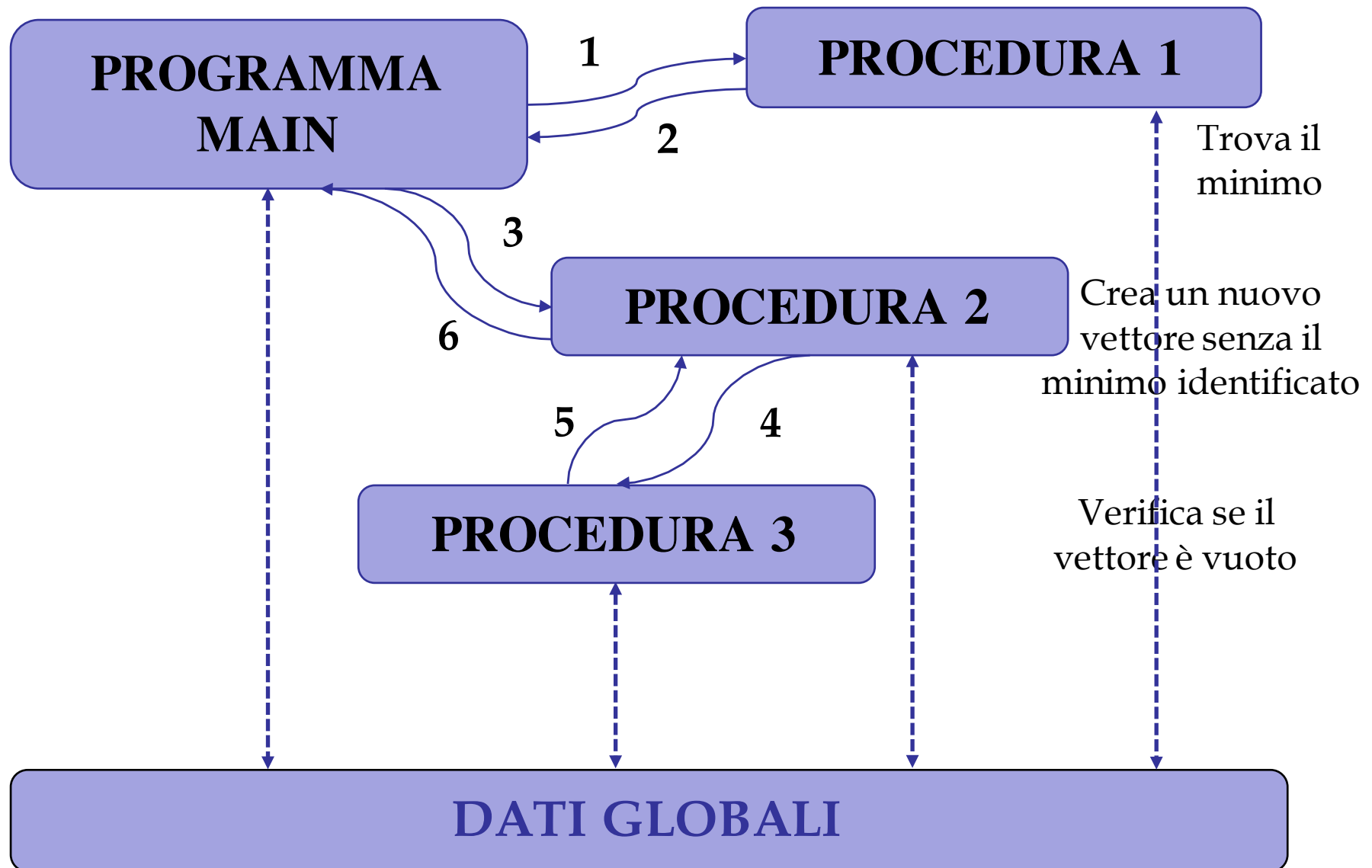


UNIVERSITÀ
DEGLI STUDI DI TRIESTE

**STRUTTURE
DATI**

**PROGRAMMA
MAIN**

PROGRAMMAZIONE PROCEDURALE

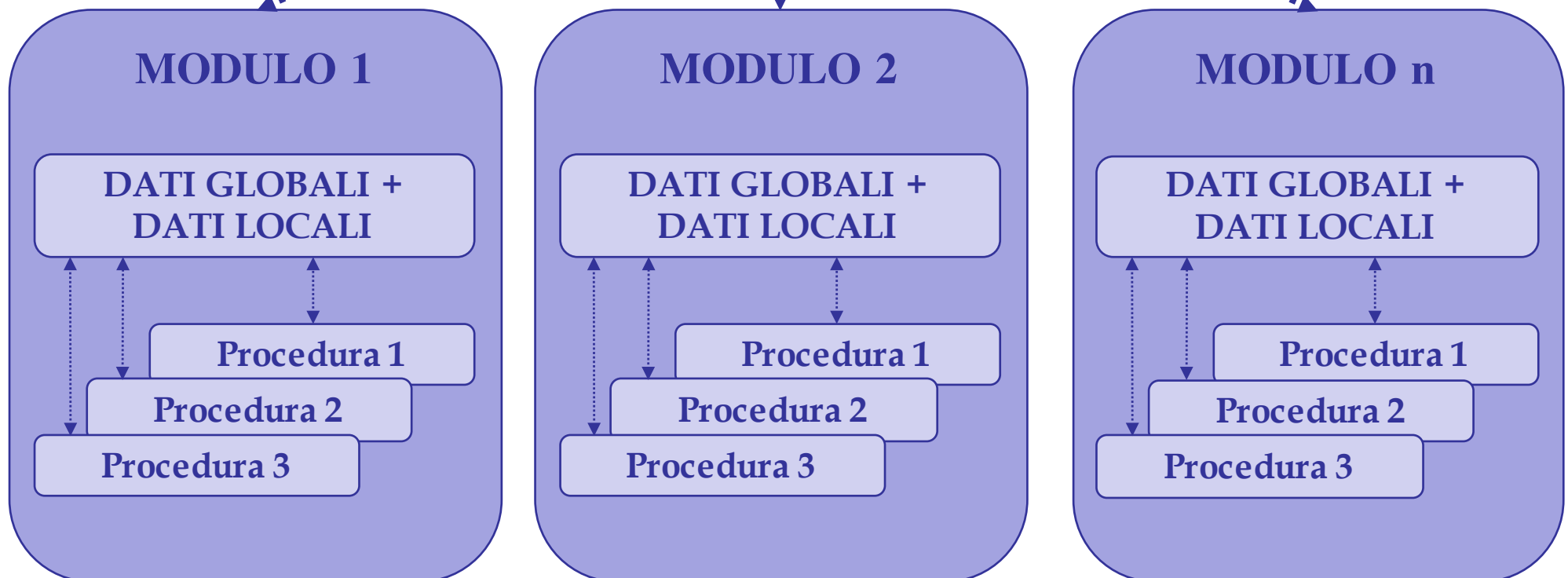


PROGRAMMAZIONE MODULARE



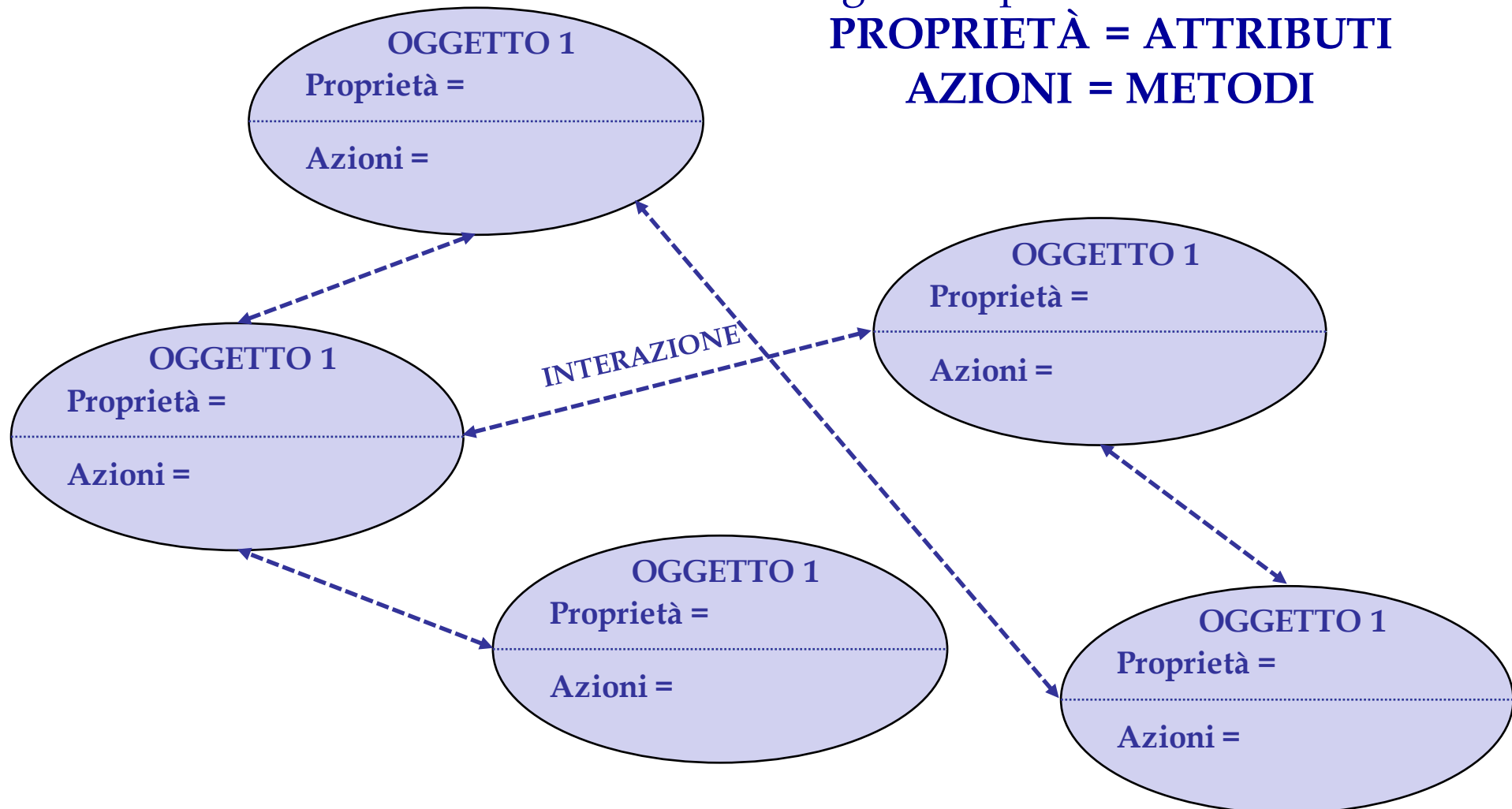
UNIVERSITÀ
DEGLI STUDI DI TRIESTE

MAIN + DATI GLOBALI



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

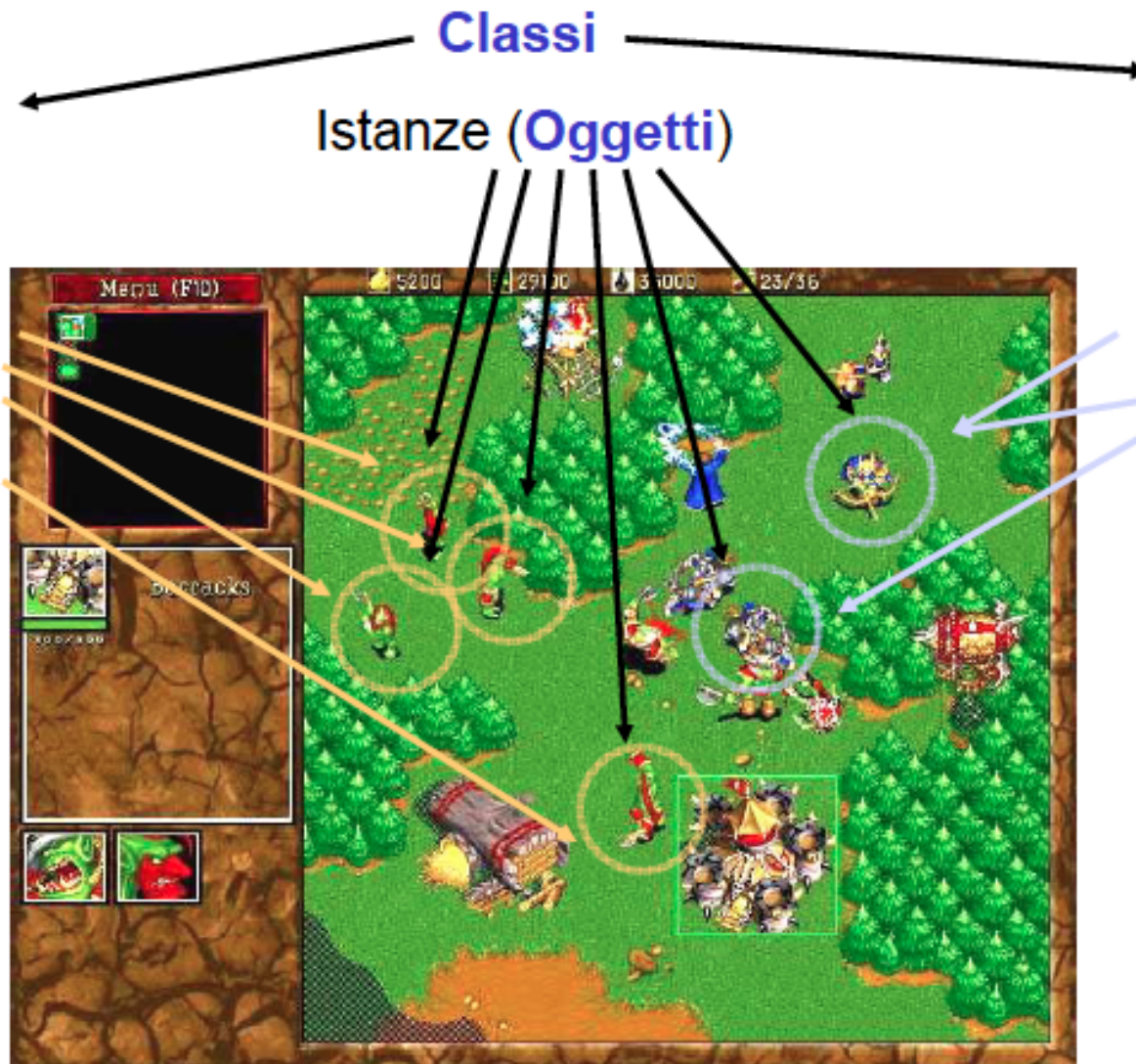
OGGETTI E CLASSI



430 1050 700 200 IRON AGE Menu Chat Diplomacy

Phalanx
120/120
Attack 19
Armor 5
Range 0

OGGETTI E CLASSI



ESEMPIO

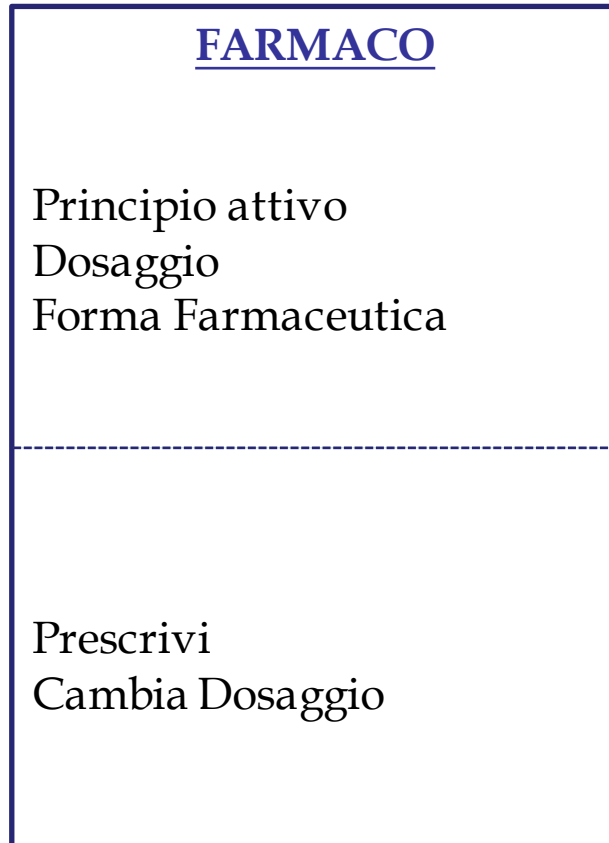


OGGETTI

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



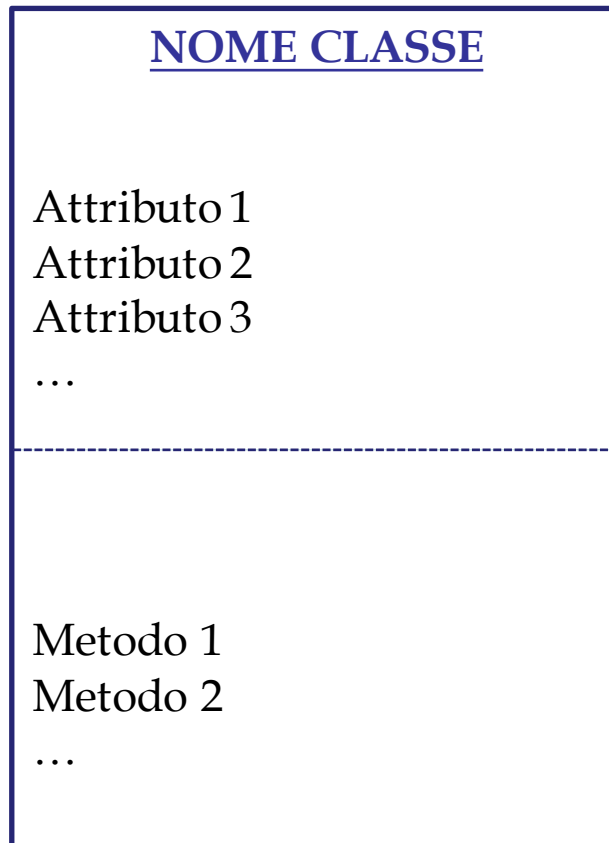
CLASSI E OGGETTI: DEFINIZIONE

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**

EREDITARIETÀ



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Classe genitrice

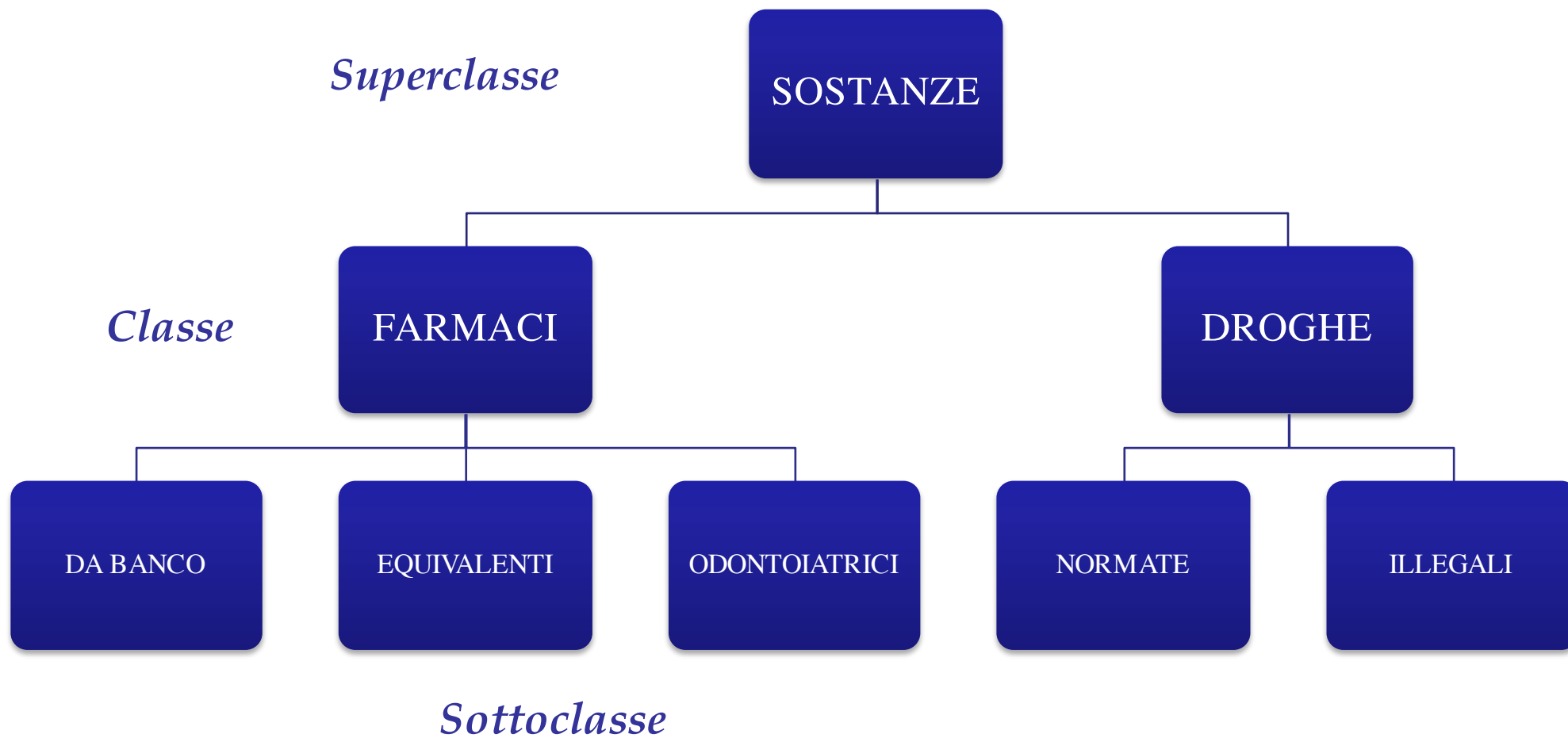
<i>farmaco</i>
Principio attivo
Dosaggio
Forma Farmaceutica
Costo SSN
Prescrivi
Cambia Dosaggio



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>

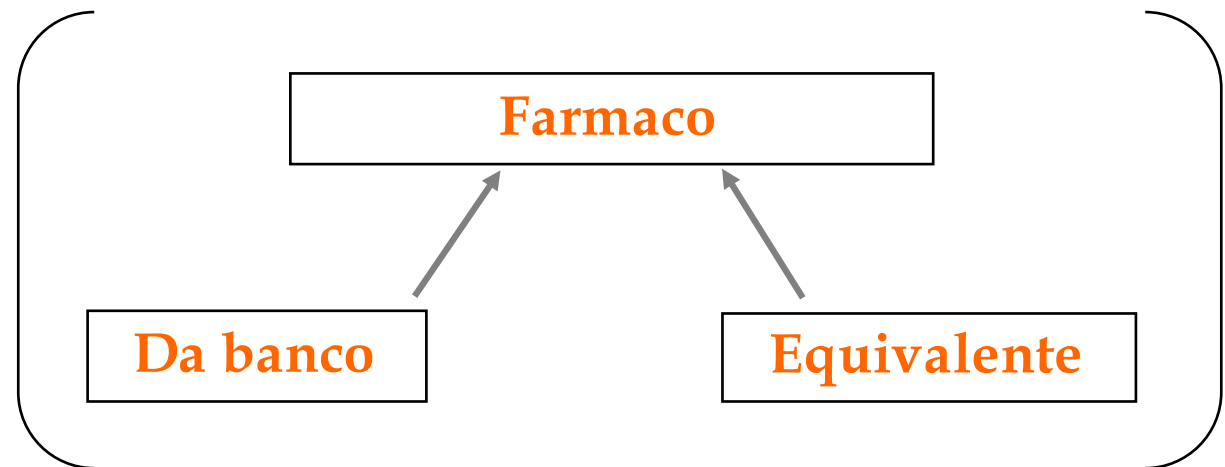
GERARCHIE



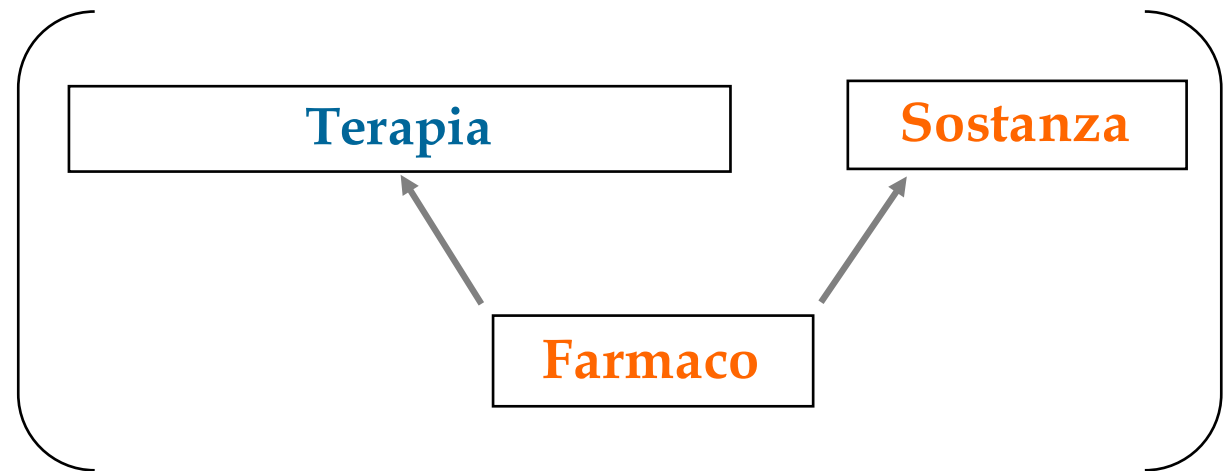
TIPI DI EREDITARETÀ



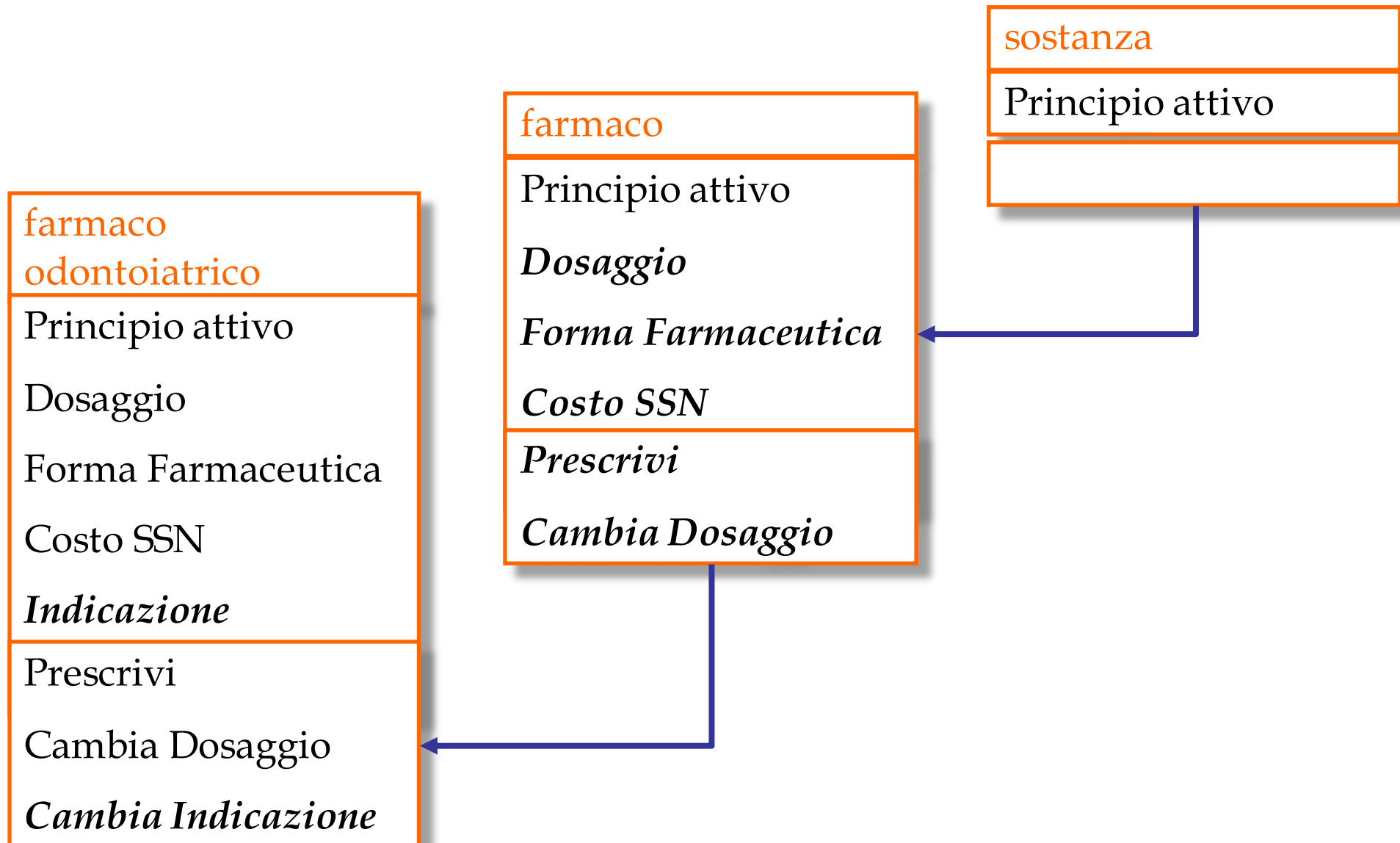
Ereditarietà
singola



Ereditarietà
multipla



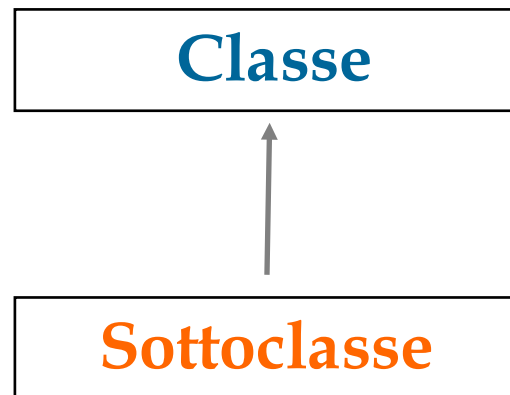
ESEMPIO



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

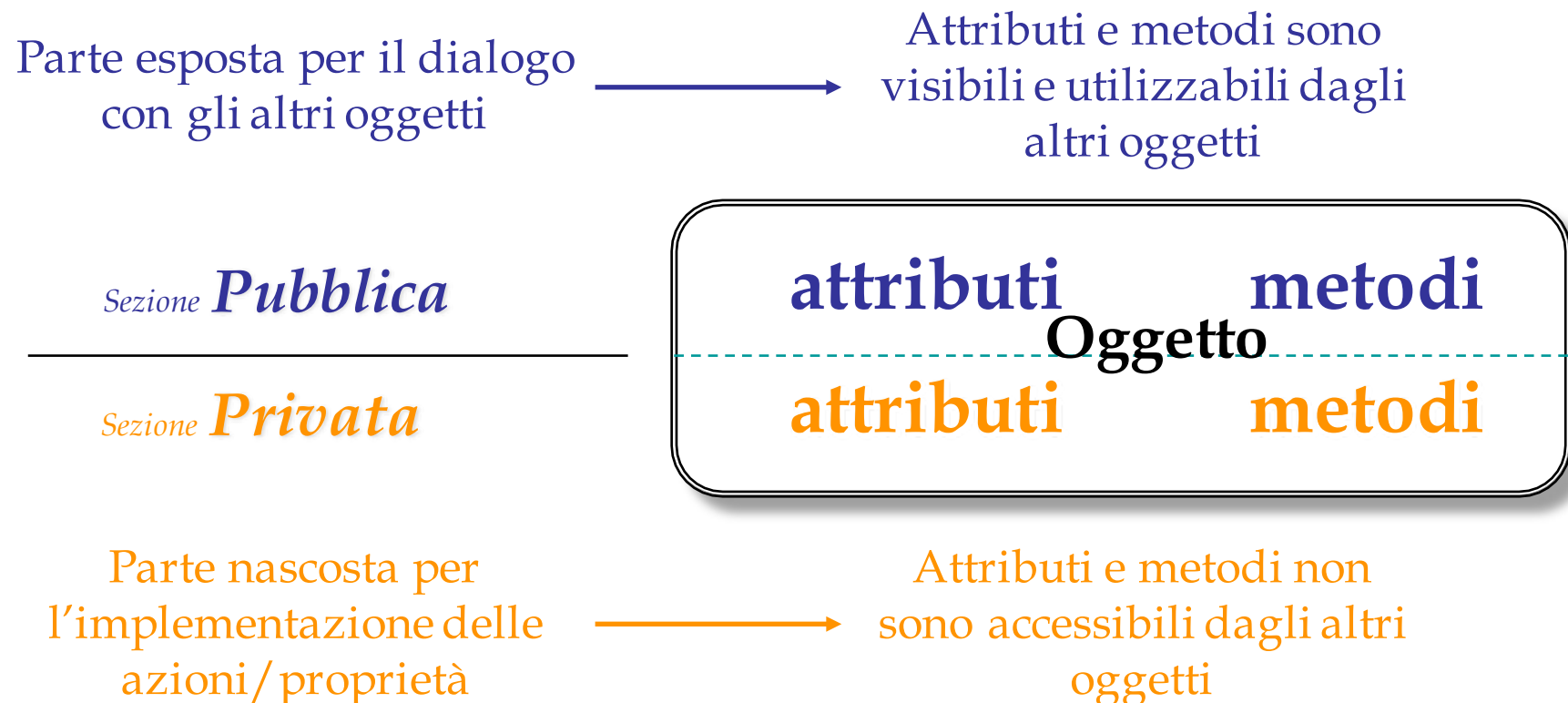
OVERLOADING = i *metodi* possono *assumere forme diverse* (cioè *implementazioni diverse*) all'interno della stessa classe

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



INCAPSULAMENTO

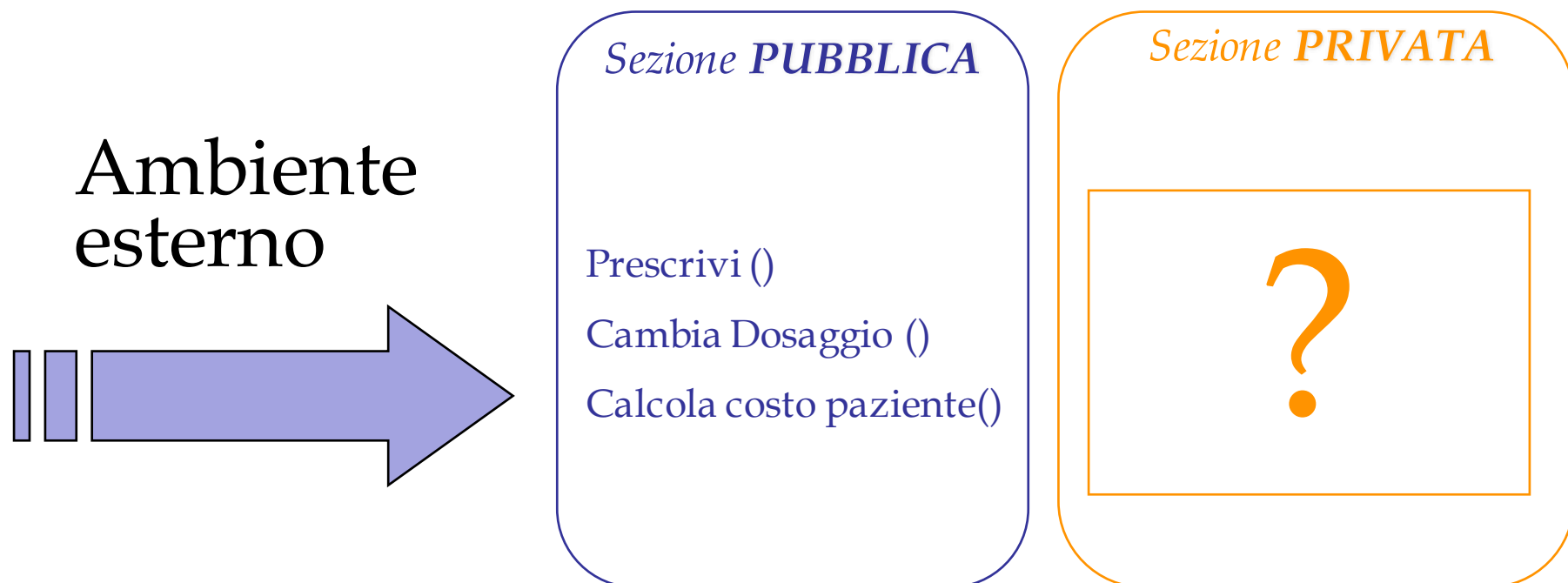
- **Incapsulamento** = Proprietà dell'oggetto di incorporare al suo interno attributi e metodi
- **Information hiding** = mascheramento dell'informazione all'interno dell'oggetto → espone solo metodi e attributi della sezione pubblica



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 ()`



ESERCIZIO



- Ambiente di lavoro: Eclipse
- Se non avete Eclipse:
 - Scaricare e installare JDK (java development kit) da <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
 - [OPZIONALE] Scaricare editor di testo (ad es TextWrang oppure EJE)
 - Scaricare e installare Eclipse (ad es, Eclipse Classic, Juno Package)

ESERCIZIO



- Implementare le classi:
 - Sostanza
 - Farmaco
 - Farmaco da banco
- La classe “Sostanza” ha come attributo solo il principio attivo
- La classe “Farmaco” ha:
 - Attributi: nome, principio attivo, dosaggio, forma farmaceutica, costo SSN
 - Metodi: prescrivere, cambia dosaggio, cambia forma farmaceutica
- La classe “Farmaco da Banco” ha, rispetto alla classe “Farmaco”,:
 - Attributi: costo al pubblico
 - Metodi: calcola differenza costo, il metodo “prescrivere” deve ritornare un’allerta in cui si dice che il farmaco non necessita di prescrizione



CLASSE SOSTANZA

```
public class Sostanza
{
    public String principioAttivo;
    private String tipoSostanza;

    //Constructor 1: senza tipo sostanza
    public Sostanza(String pA) {
        principioAttivo = pA;
    };

    //Constructor 2: con tipo sostanza
    public Sostanza(String pA, String type) {
        principioAttivo = pA;
        tipoSostanza = type;
    };
};
```



CLASSE FARMACO (ATTRIBUTI)

```
public class Farmaco extends Sostanza{
```

```
    public String formaFarmaceutica;
```

```
    double dosaggio;
```

```
    int costoSSN;
```

```
    public String nome;
```

```
    //Constructor con il solo attributo di Sostanza
```

```
    public Farmaco(String pA) {
```

```
        super(pA);
```

```
        // TODO Auto-generated constructor stub
```

```
    }
```

```
    //Constructor con tutti gli attributi --> OVERLOAD
```

```
    public Farmaco (String pA, double dose, int cost, String form, String name)
```

```
    {
```

```
        super(pA);
```

```
        formaFarmaceutica = form;
```

```
        costoSSN = cost;
```

```
        dosaggio = dose;
```

```
        nome = name;
```

```
    }
```



CLASSE FARMACO (METODI)

//Metodo Stampa Farmaco

```
public void stampaFarmaco(Farmaco f){  
    System.out.println ("Nome farmaco = " + f.nome);  
    System.out.println ("Principio Attivo = " + f.principioAttivo);  
    System.out.println ("Dosaggio = " + f.dosaggio);  
    System.out.println ("Forma Farmaceutica = " +  
f.formaFarmaceutica);  
    System.out.println ("Costo previsto dal SSN = " + f.costoSSN);  
}
```

//Metodo PRESCRIVI

```
public void prescrivi (String nomePaz, String nomeDott)  
{  
    System.out.println ("Il Farmaco " + nome + " è stato prescritto a " +  
nomePaz + " da " + nomeDott);  
}
```



CLASSE FARMACO (METODI)

//Metodo cambia Dosaggio (setter)

public void cambiaDosaggio (double newDose)

{

dosaggio = newDose;

}

//Metodo cambia Forma (setter)

public void cambiaForma (String newForma)

{

formaFarmaceutica = newForma;

}

}



SETTER E GETTER

- Se impongo che un attributo sia “private”, per poter accedere devo avere dei metodi appositi

```
public class Farmaco extends Sostanza{  
    public String formaFarmaceutica;  
    public double dosaggio;  
    private int costoSSN;  
    public String nome;  
  
    //Setter → posso cambiare il valore dell'attributo  
    public void setCostoSSN (int newCost){  
        costoSSN = newCost; }  
  
    //Getter → posso leggere il valore dell'attributo  
    public int getCostoSSN(){  
        return costoSSN;}  
}
```

CLASSE FARMACO DA BANCO



```
public class FarmacoDaBanco extends Farmaco {

    public int costoAlPubblico;

    public FarmacoDaBanco(String pA, String name, double dose, String form,
                           int cost, int costoPubblico) {
        super(pA, name, dose, form, cost);
        costoAlPubblico = costoPubblico;
        // TODO Auto-generated constructor stub
    }

    public int calcolaCosto () {

        int differenza = costoAlPubblico-costoSSN;
        return differenza;

    }
}
```


CLASSE FARMACO DA BANCO



//Override metodo prescrivi

```
public void prescrivi () {
```

```
    System.out.println("Il farmaco " + nomeFarmaco + " non è prescrivibile");
```

```
}
```

//Override metodo stampa

```
public void stampaFarmaco (){
```

```
    System.out.println ("Nome farmaco = " + nomeFarmaco);
```

```
    System.out.println ("Principio Attivo = " + principioAttivo);
```

```
    System.out.println ("Dosaggio = " + dosaggio);
```

```
    System.out.println ("Forma farmaceutica = " + formaFarmaceutica);
```

```
    System.out.println ("Costo per il SSN = " + costoSSN);
```

```
    System.out.println ("Costo al pubblico = " + costoAlPubblico);
```

```
}
```

```
}
```



MAIN ESEMPIO

```
public class Primary{
public static void main(String[] args)
    {
        Sostanza drug1 = new Sostanza("levodopa");
        System.out.println(drug1.principioAttivo);
        Farmaco farmaco1 = new Farmaco ("ibuprofene");
        //farmaco1.costoSSN=15;
        farmaco1.setCostoSSN(21);
        Farmaco farmaco2 = new Farmaco ("paracetamolo","tachipirina",250,"Sciroppo",7);
        farmaco2.stampaFarmaco();
        farmaco1.dosaggio=300;
        farmaco1.nomeFarmaco="Moment 300";
        farmaco1.cambiaForma("Compresse dispersibili");
        farmaco1.stampaFarmaco();
        // int costo1 = farmaco1.costoSSN;
        int costo1 = farmaco1.getCostoSSN();
        System.out.println(costo1);
        FarmacoDaBanco fb = new FarmacoDaBanco ("ketoprofene", "Ketoprofene Sandoz",5,
        "Gel",9,10);
        fb.stampaFarmaco();
        System.out.println ("SCONTRINO");
        System.out.println ("Prezzo farmaco " + fb.nomeFarmaco + " = " + fb.calcolaCosto() + "
        euro");
        farmaco1.prescrivi("Mario Rossi", "Alberto Bianchi");
        fb.prescrivi(); }}}
```