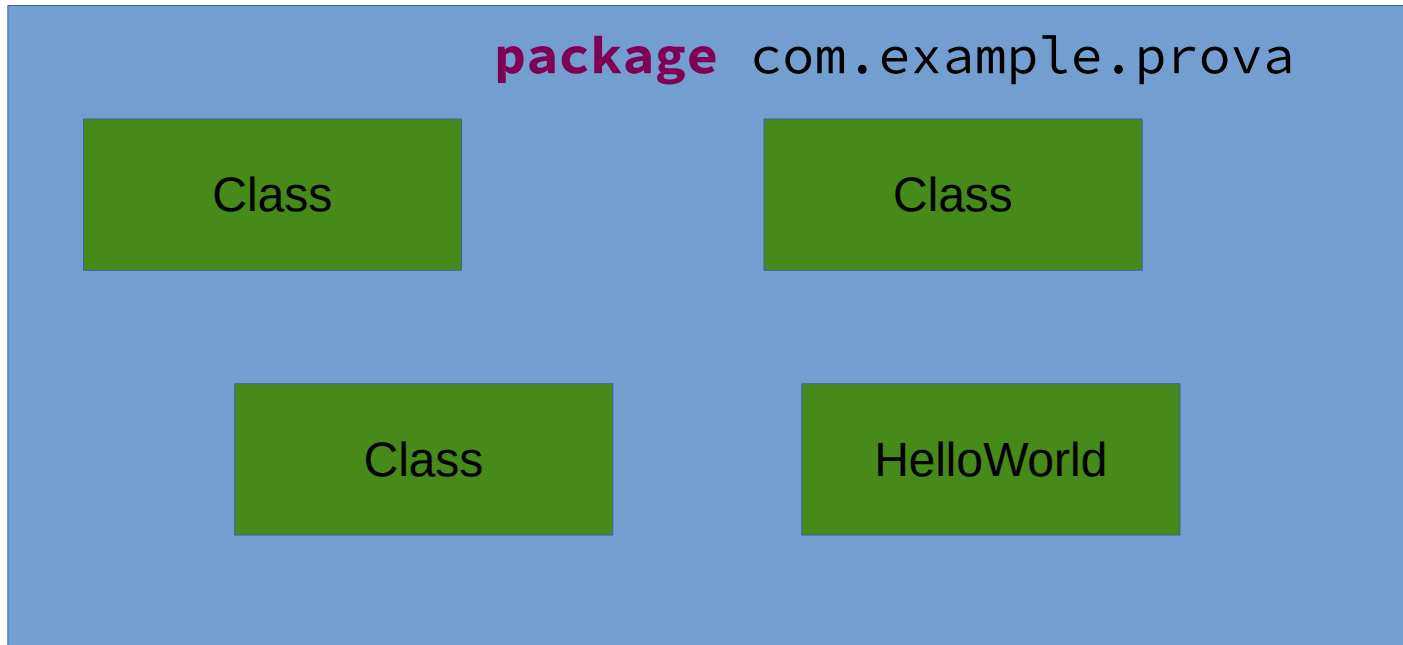


Java

15 Luglio 2021

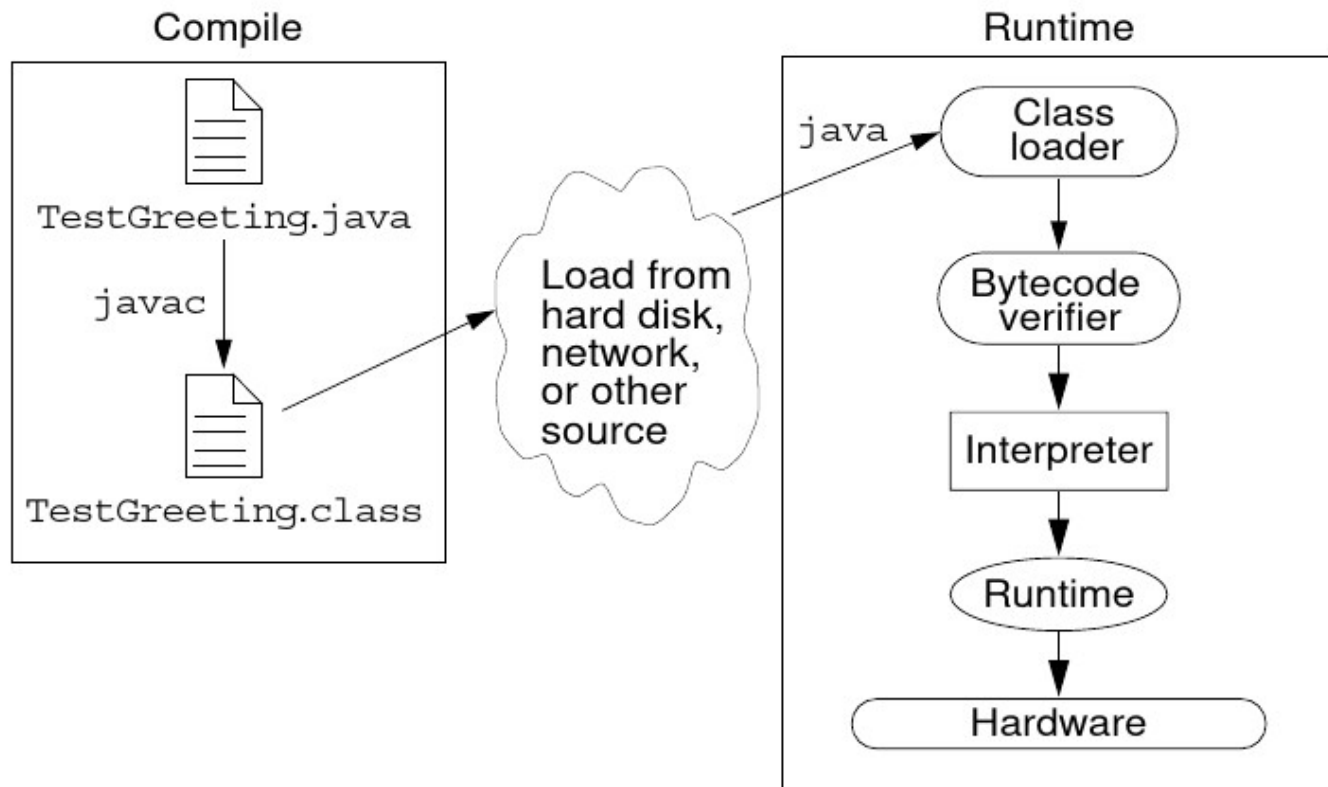
Java - Sorgenti

Package (pkg1.pkg2.pkg3....pkgn => /pkg1/pkg2/pkg3/...)



The Java Runtime Environment

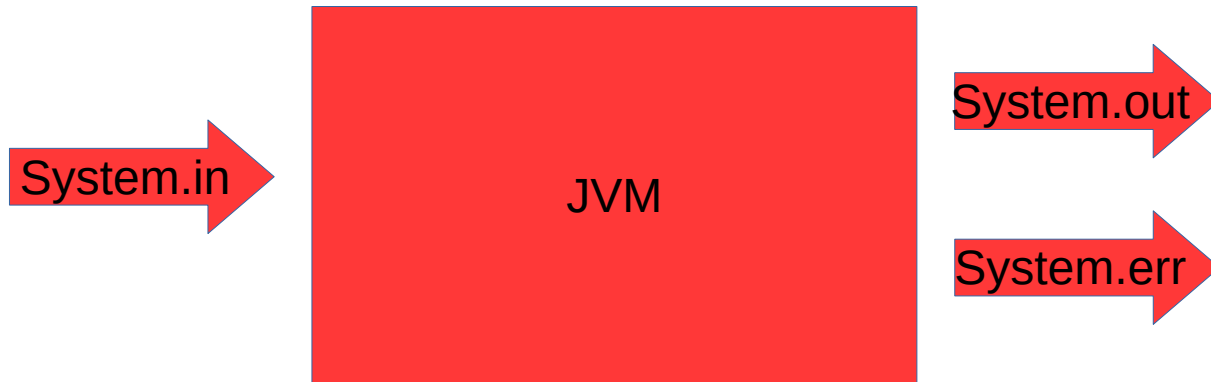
The Java application environment performs as follows:



Package

- Raggruppamento di “classi” in qualche modo collegate logicamente.
- Il package viene realizzato tramite una struttura di directory nel file-system.

Canali i/o standard



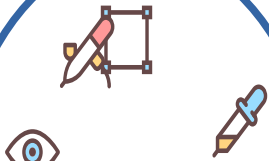
Objects



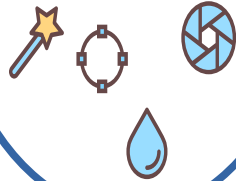
What is an Object?



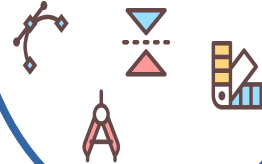
Writing Tools



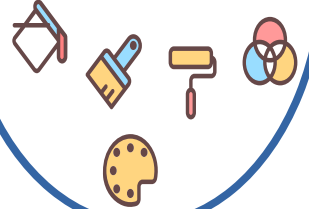
Unknown Tools



Measuring Tools



Painting Tools



Drawing Tools



Magnetic Tools



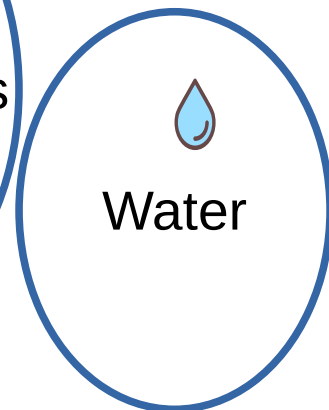
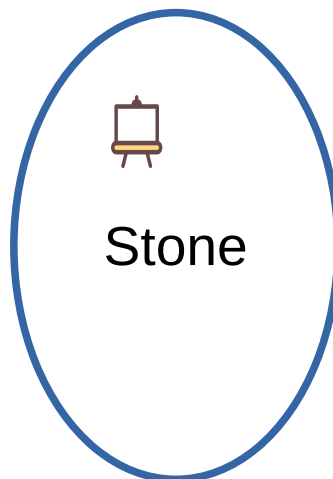
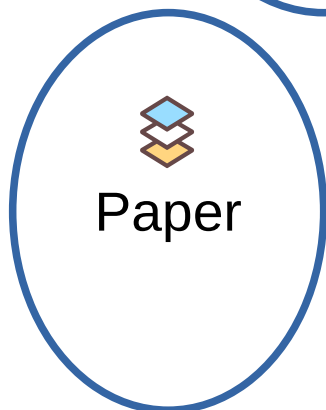
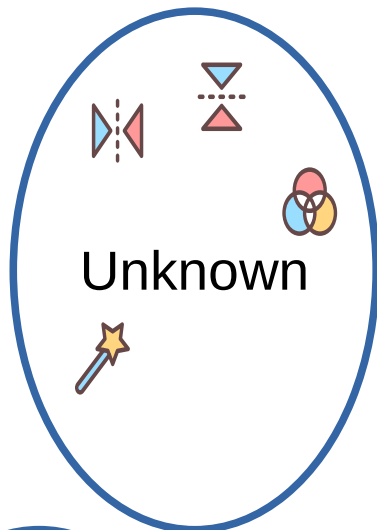
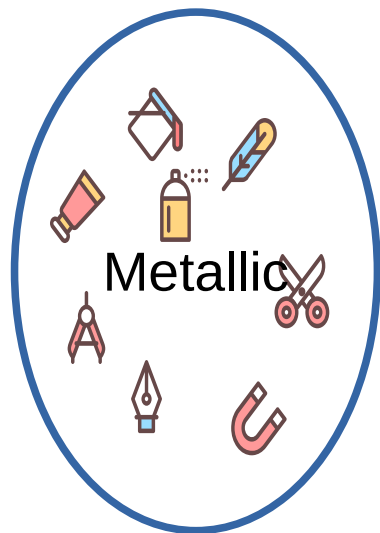
Office Tools



School Tools



Raggruppamento per **utilizzo**



Raggruppamento per **materiale**

Objects

Raggruppare =
Classificare

Objects

What is a **Class**?

= Blueprint (Progetto)
(Modello)(Astrazione)

=

Tipo

Objects

What is an Object?
qualcosa che si
costruisce a partire
dalla classe.

Oggetto?

- Un oggetto è **l'istanza** di una classe
- In JAVA si realizza tramite la parola chiave **new**

OOP → Incapsulamento

NomeClass {

public

private

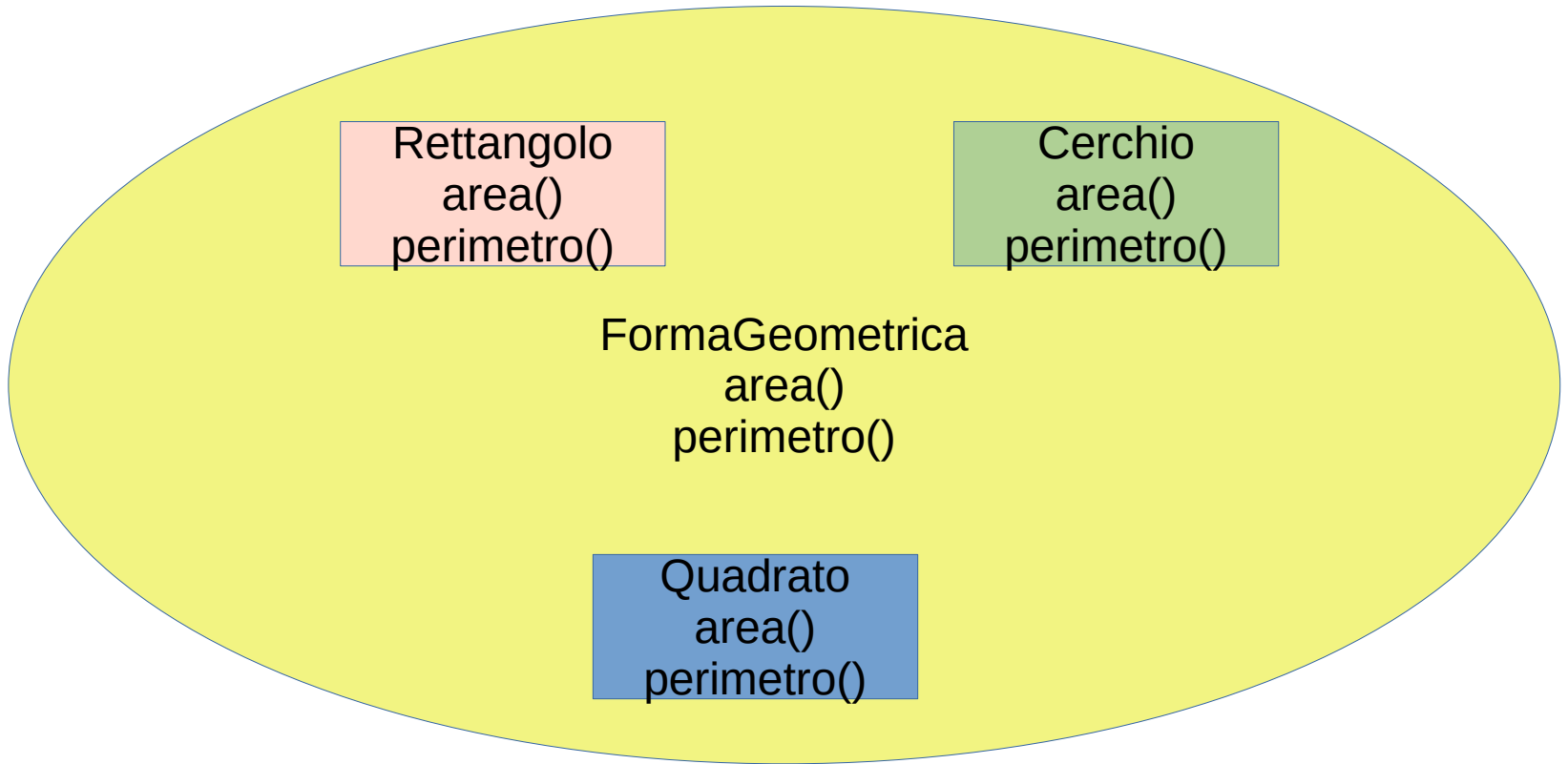
Capsula che contiene
dati
metodi

}

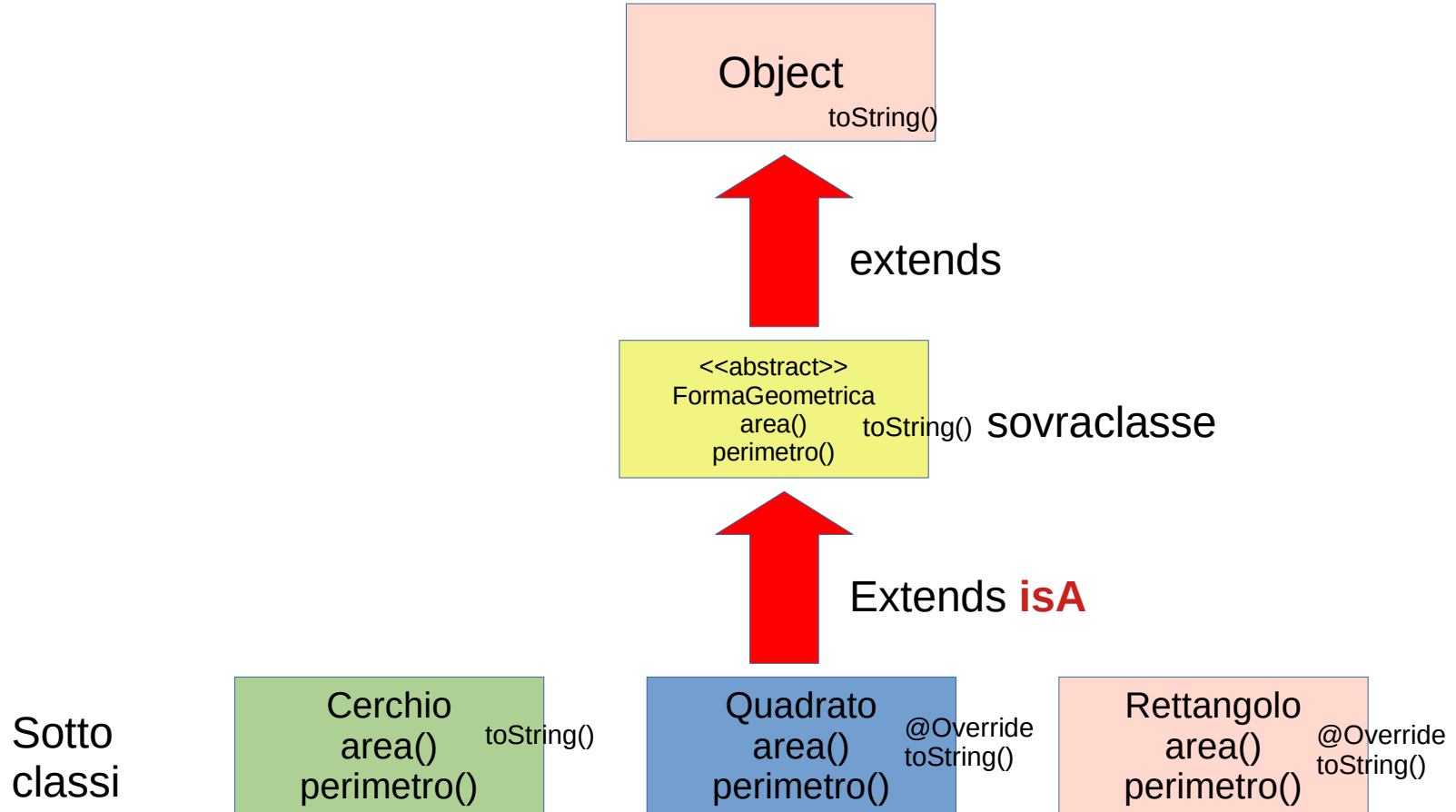
protected Visibile solo nella gerarchia

default Visibile solo nel package

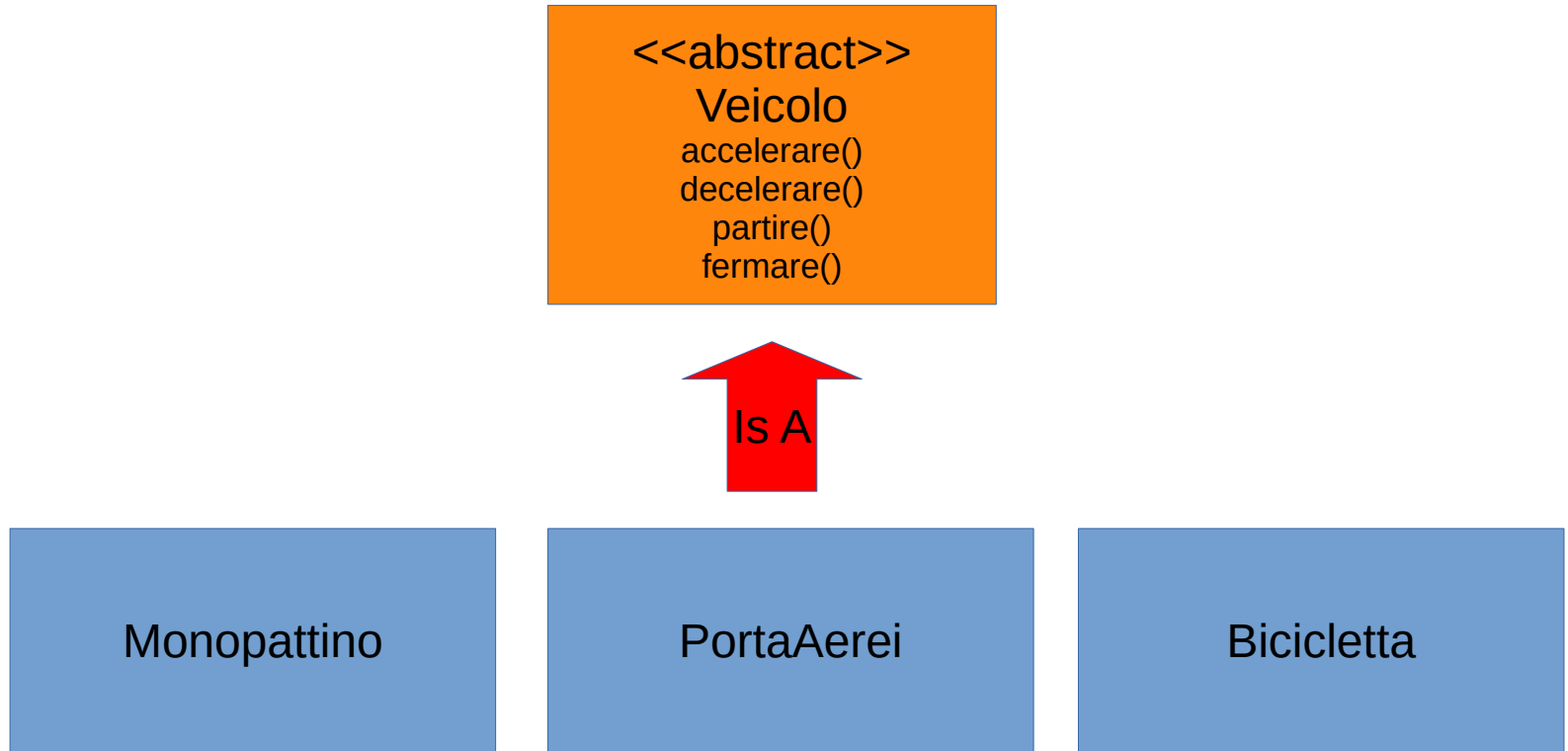
OOP → Ereditarietà



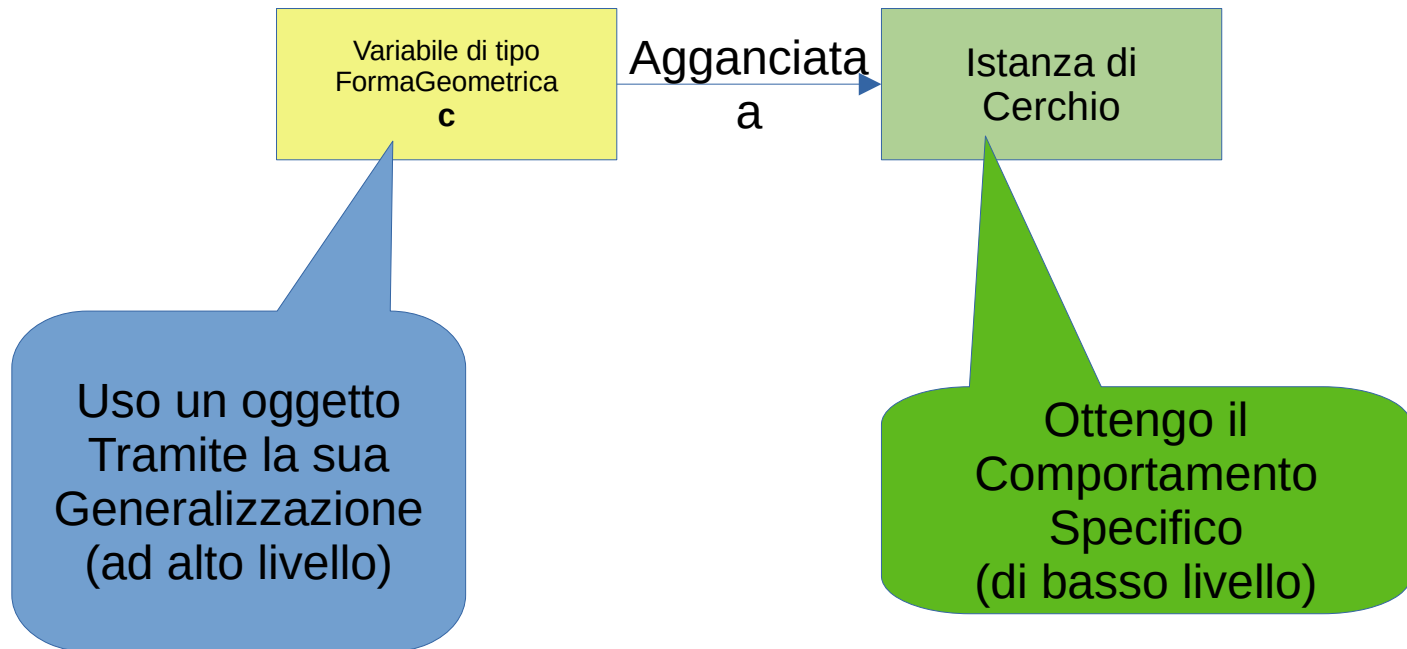
OOP → Ereditarietà



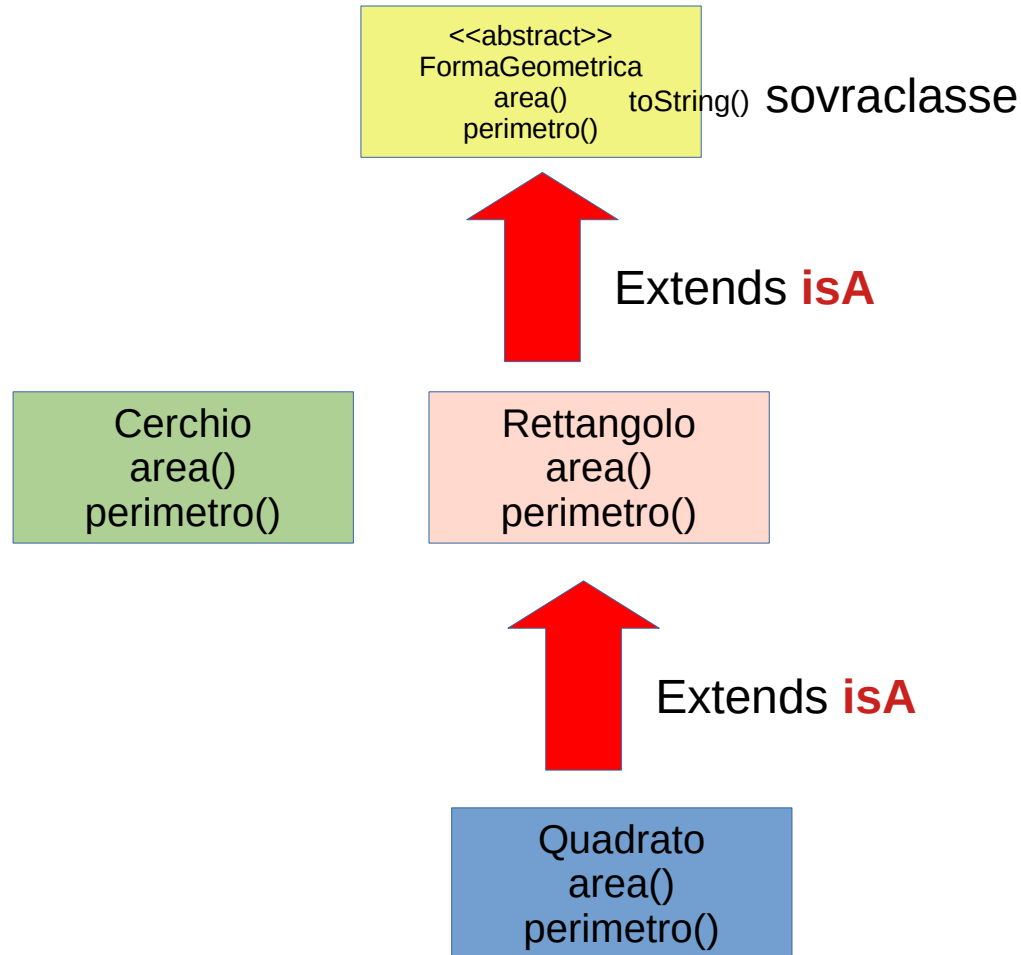
OOP → Ereditarietà



OOP - Polimorfismo



OOP → Ereditarietà

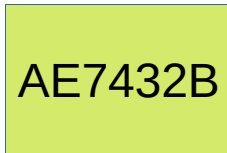


Tipi di Dato e pass by value

Int x=100;



Quadrato q1 = new Quadrato(10);



riferimento

Punta

AE7432B

Quadrato di lato 10



Tipi di Dato

- Dicotomia
 - Tipi di dato primitivi (int, long, float, double, char, byte, boolean)
variabile contiene il valore
 - Classi => Oggetti
variabile contiene il riferimento (puntatore)
alla zona della JVM dove sono memorizzati i dati

Wrapper Classes

generano **oggetti immutabile**

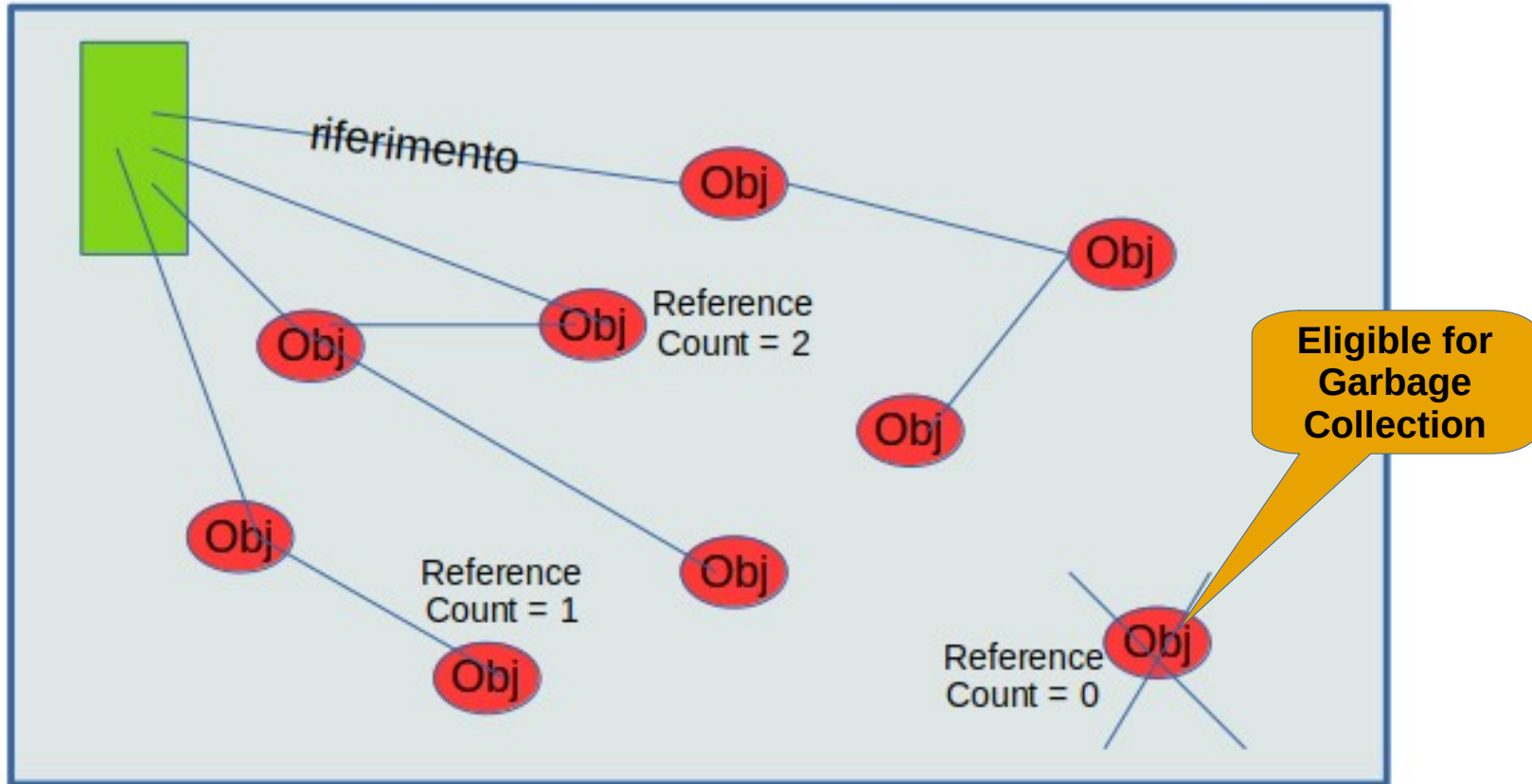
- `int ==> Integer`
- `double ==> Double`
- `boolean ==> Boolean`
- `char ==> Character`

~~`Integer x = new Integer(10);`~~

~~`int y = x.intValue();`~~

Boxing e Unboxing

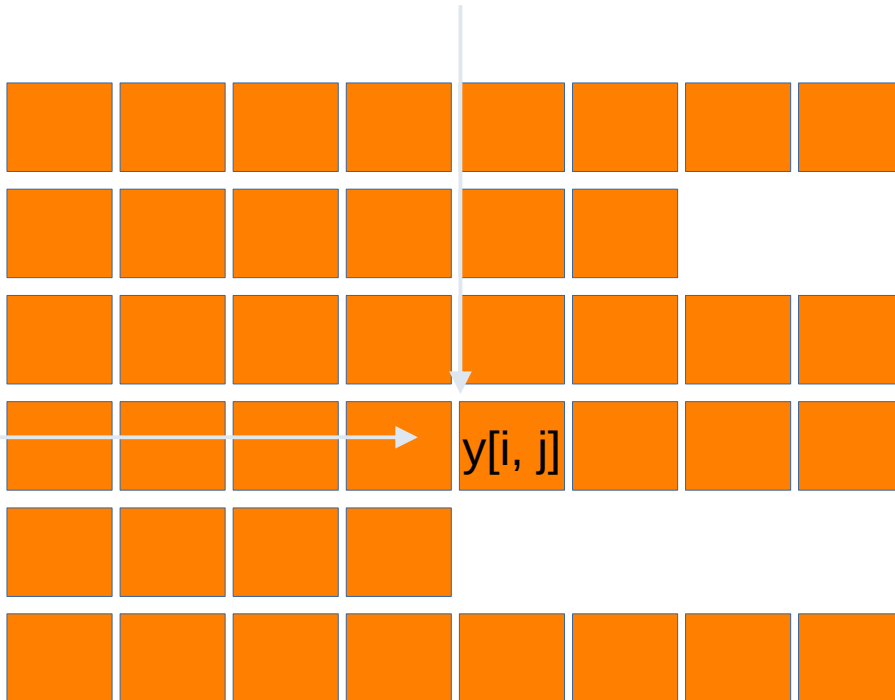
JVM – garbage collection



Array

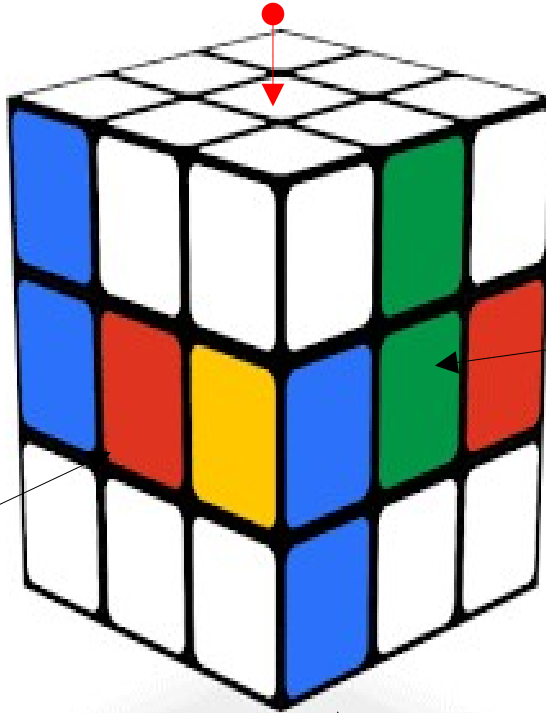


Array Monodimensionale
1 indice per scorrerlo (i)



Array Bidimensionale
2 indice per scorrerlo (i,j)

$t[i, j, k]$



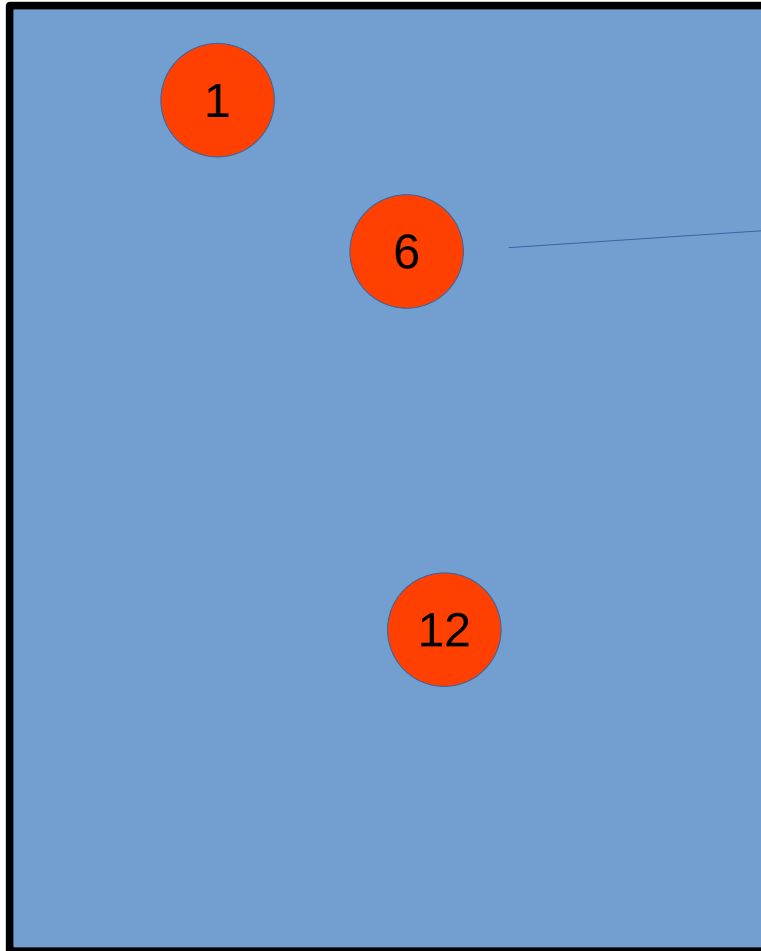
Array Tridimensionale
3 indice per scorrerlo [i,j, k]

Array

- Un array è un oggetto e deve essere creato con new.
- Qual'è lo scopo di un array?
 - Raggruppare data objects dello stesso tipo.
 - Raggruppare valori primitivi or class types.

Es: `int s[];` `Point [] pts;`

Tombola

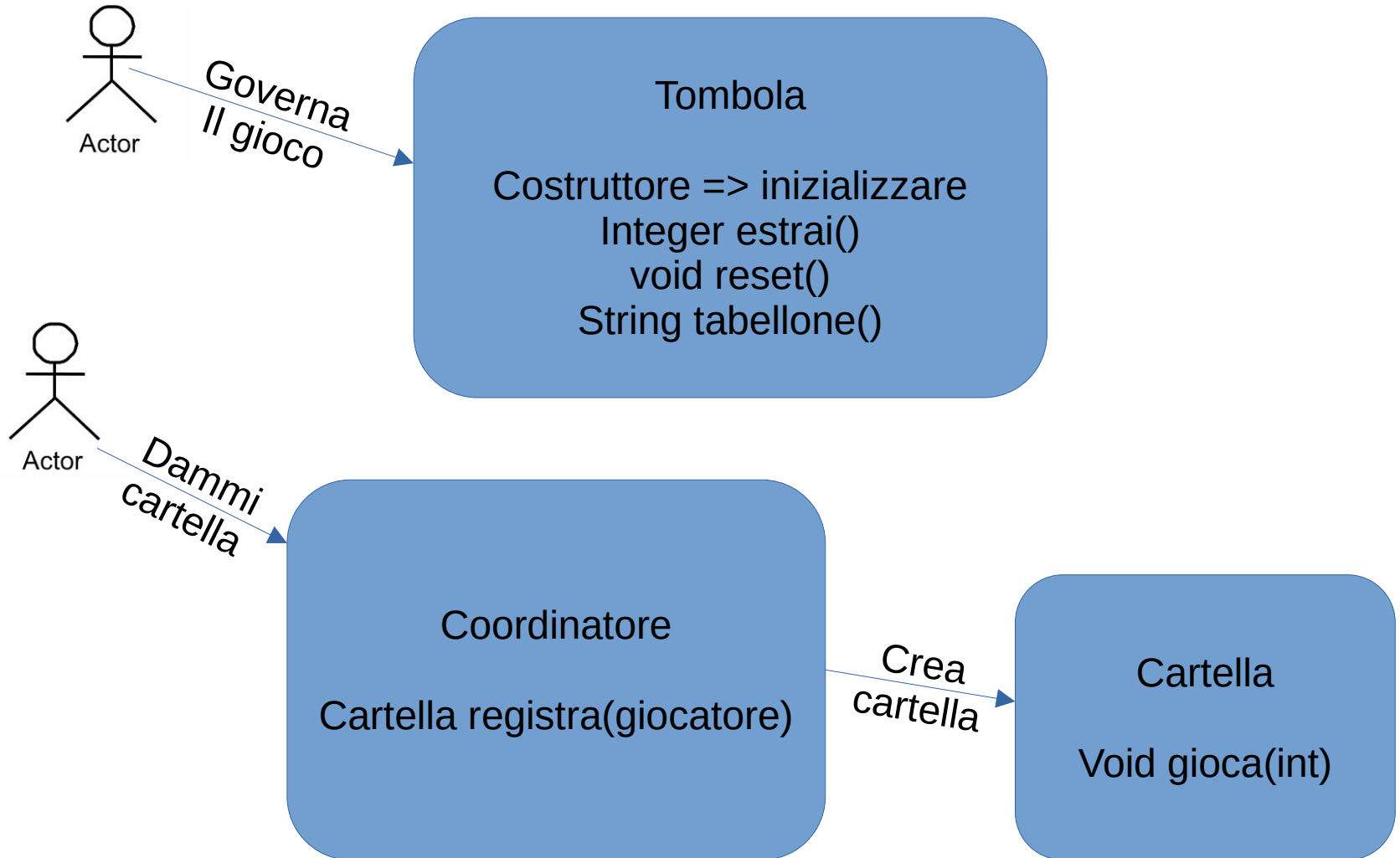


1) riempire il
sacchetto.

2) Estrarli uno
per volta
e una volta
sola

=> estraggo
un numero lo
rimuovo dal
sacchetto

Tombola



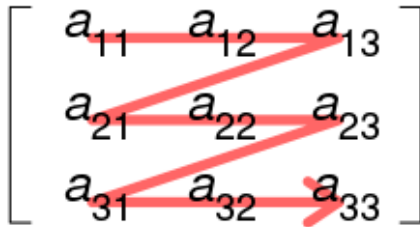
Tombola

TABELLONE

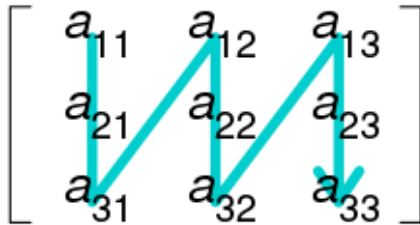
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90

Tombola

Row-major order

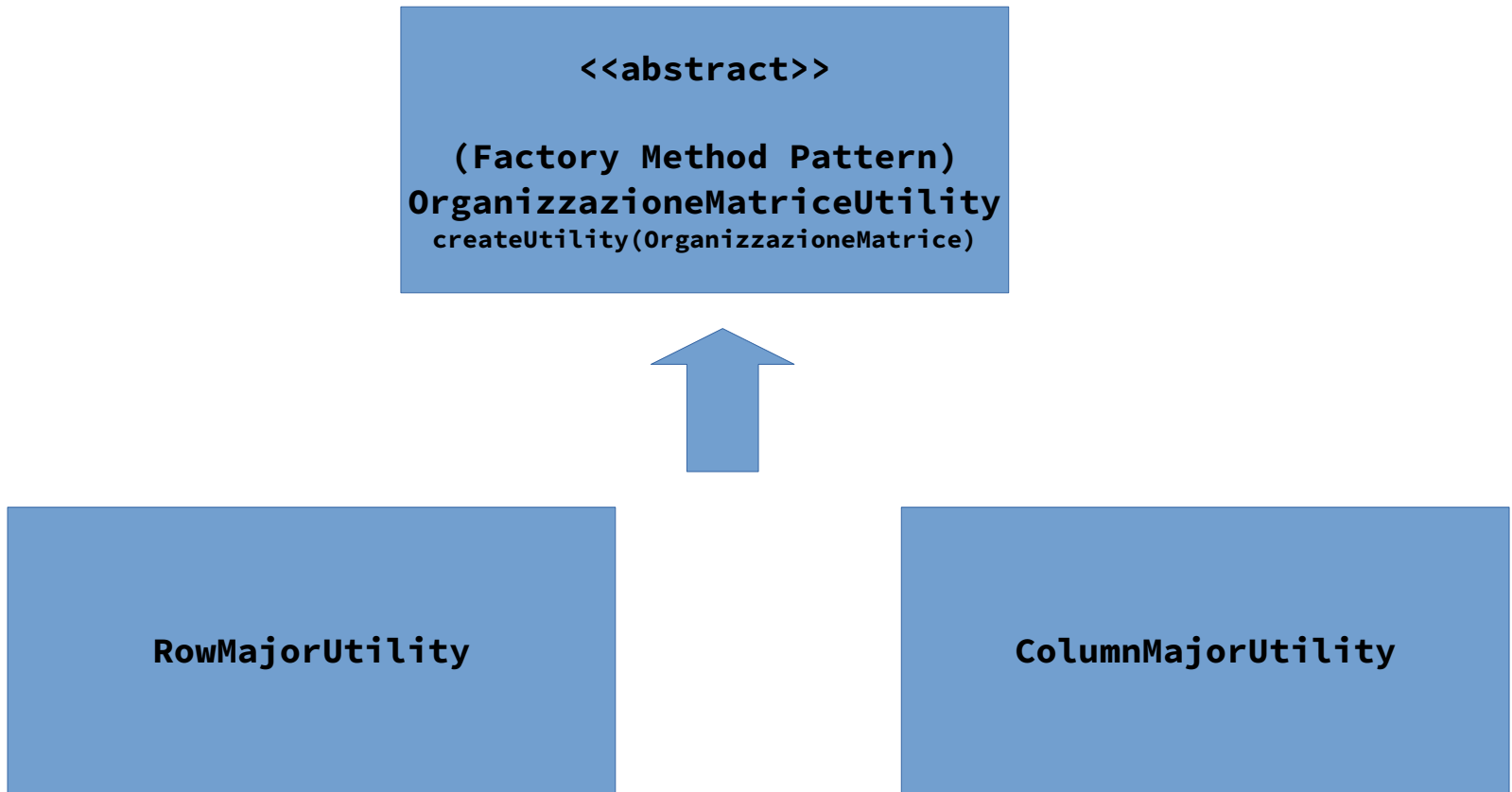


Column-major order

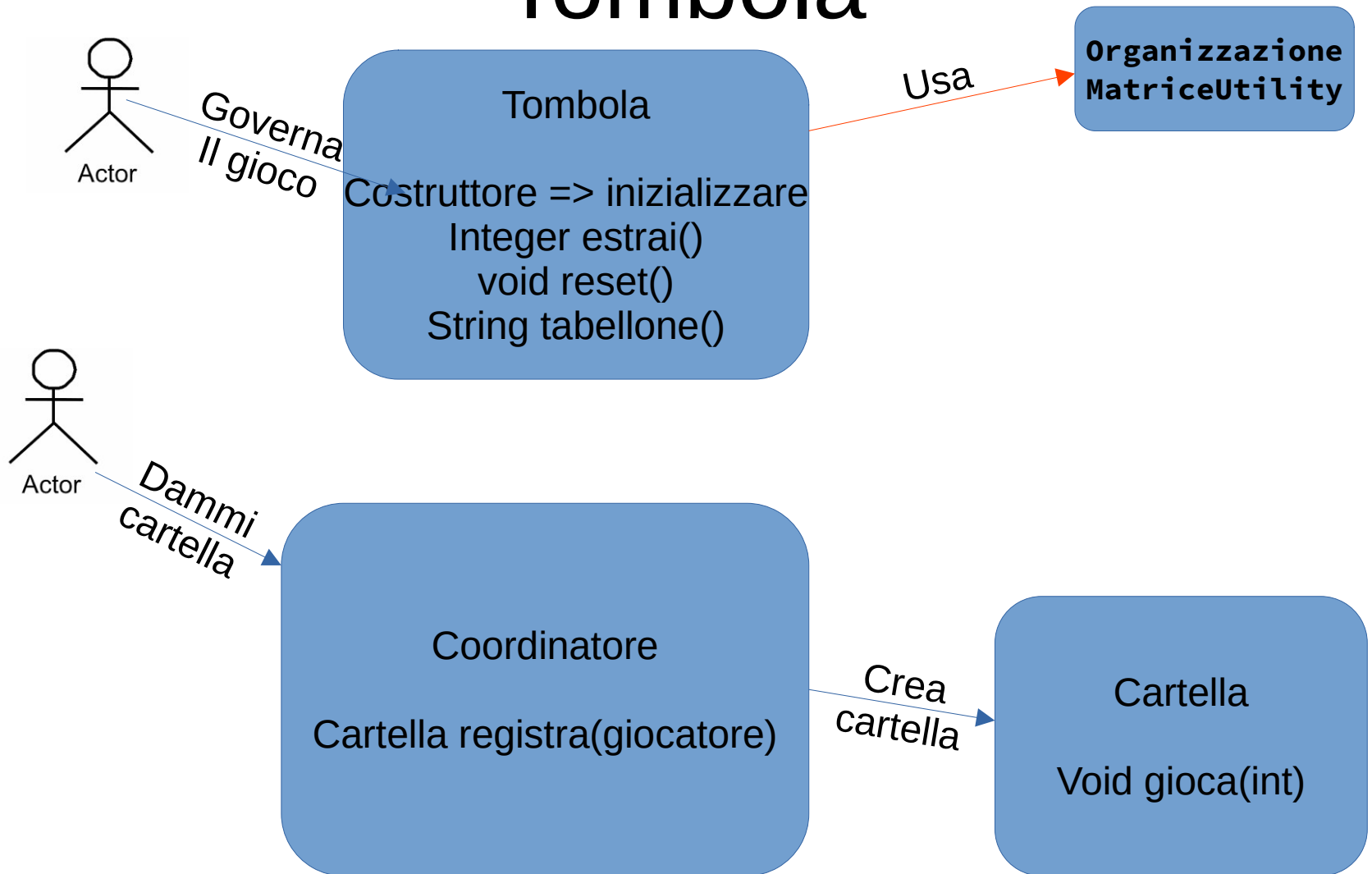


- Il tabellone ha ordine **row-major**

Tombola

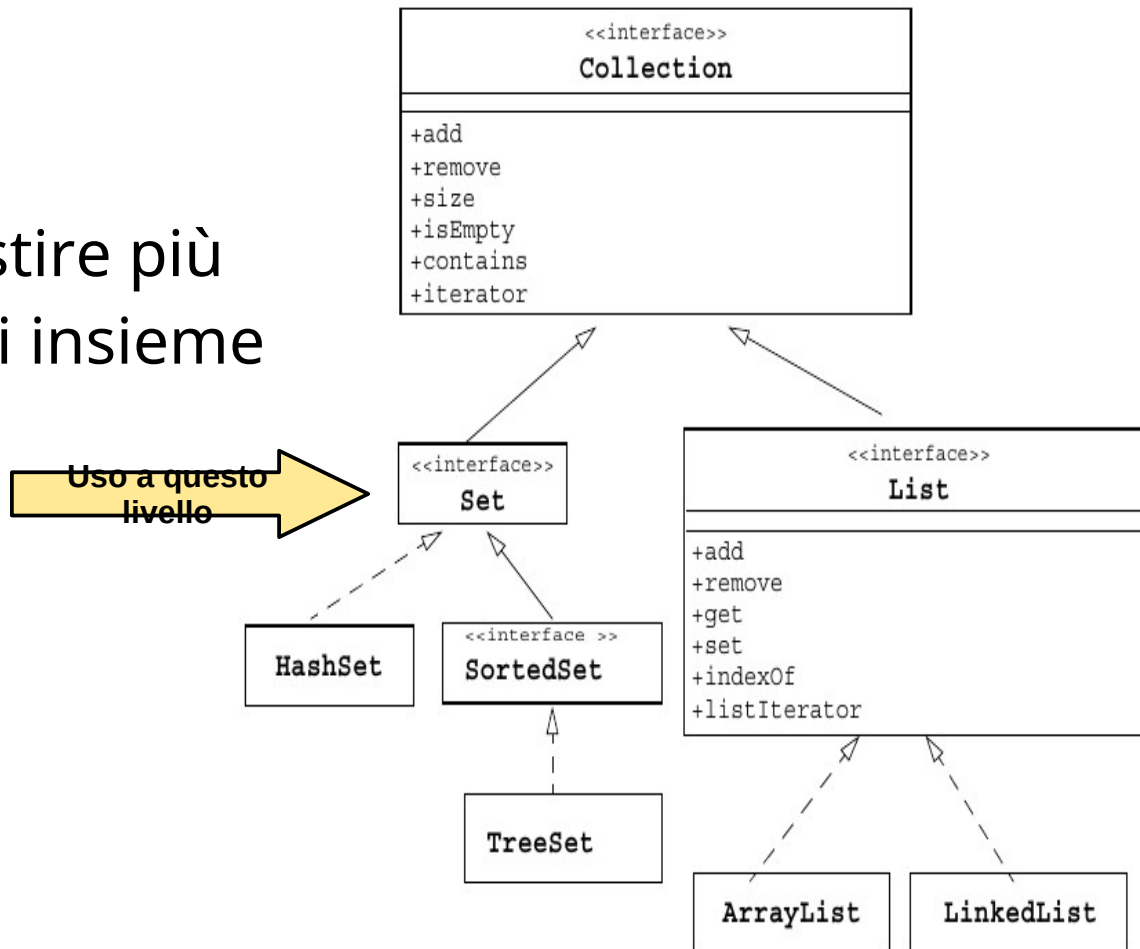


Tombola

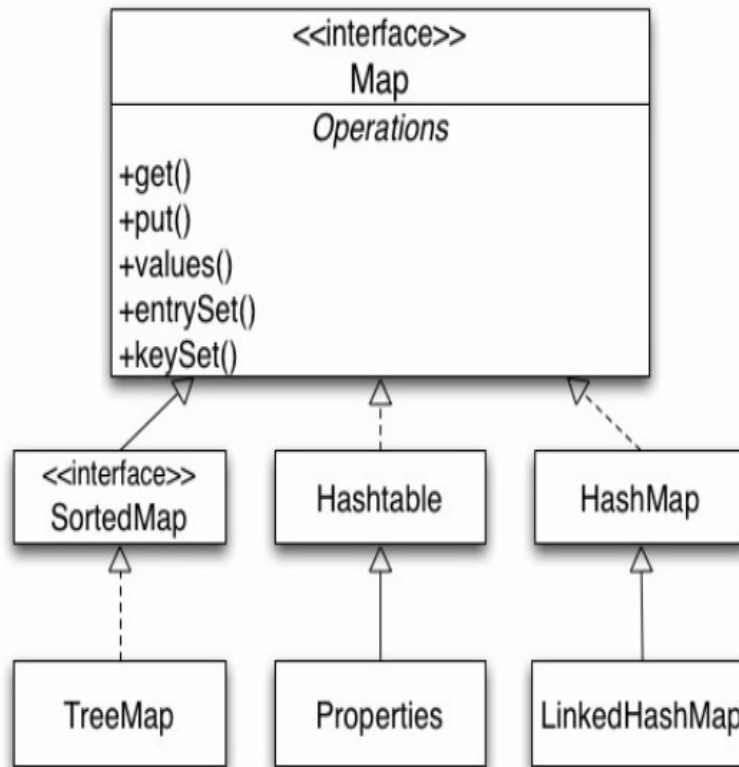


The Collections API

Per gestire più
Oggetti insieme



The Collections API



List

```
List lista = new ArrayList(); // creare la lista
```

```
Oggetto o = new Oggetto("Inserire in lista posizione "+i);  
lista.add(o); // aggiungere oggetti
```

```
for(Object o: lista) {  
    System.out.println(o); // usare la lista  
}
```

```
lista.remove(3); // rimuovere un elemento
```

```
lista.clear(); // svuotare la lista
```

```
lista.contains(o); // la lista contiene un elemento?
```

```
Object [] objects = lista.toArray(); // convertiamo la lista in un array
```

List - Generics

```
List<Oggetto> lista = new ArrayList<Oggetto>(); // creare la lista
```

```
Oggetto o = new Oggetto("Inserire in lista posizione "+i);  
lista.add(o); // aggiungere oggetti
```

```
for(Object o: lista) {  
    System.out.println(o); // usare la lista  
}
```

```
lista.remove(3); // rimuovere un elemento
```

```
lista.clear(); // svuotare la lista
```

```
lista.contains(o); // la lista contiene un elemento?
```

```
Object [] objects = lista.toArray(); // convertiamo la lista in un  
array
```

Generics

Generics

Using non-generic collections:

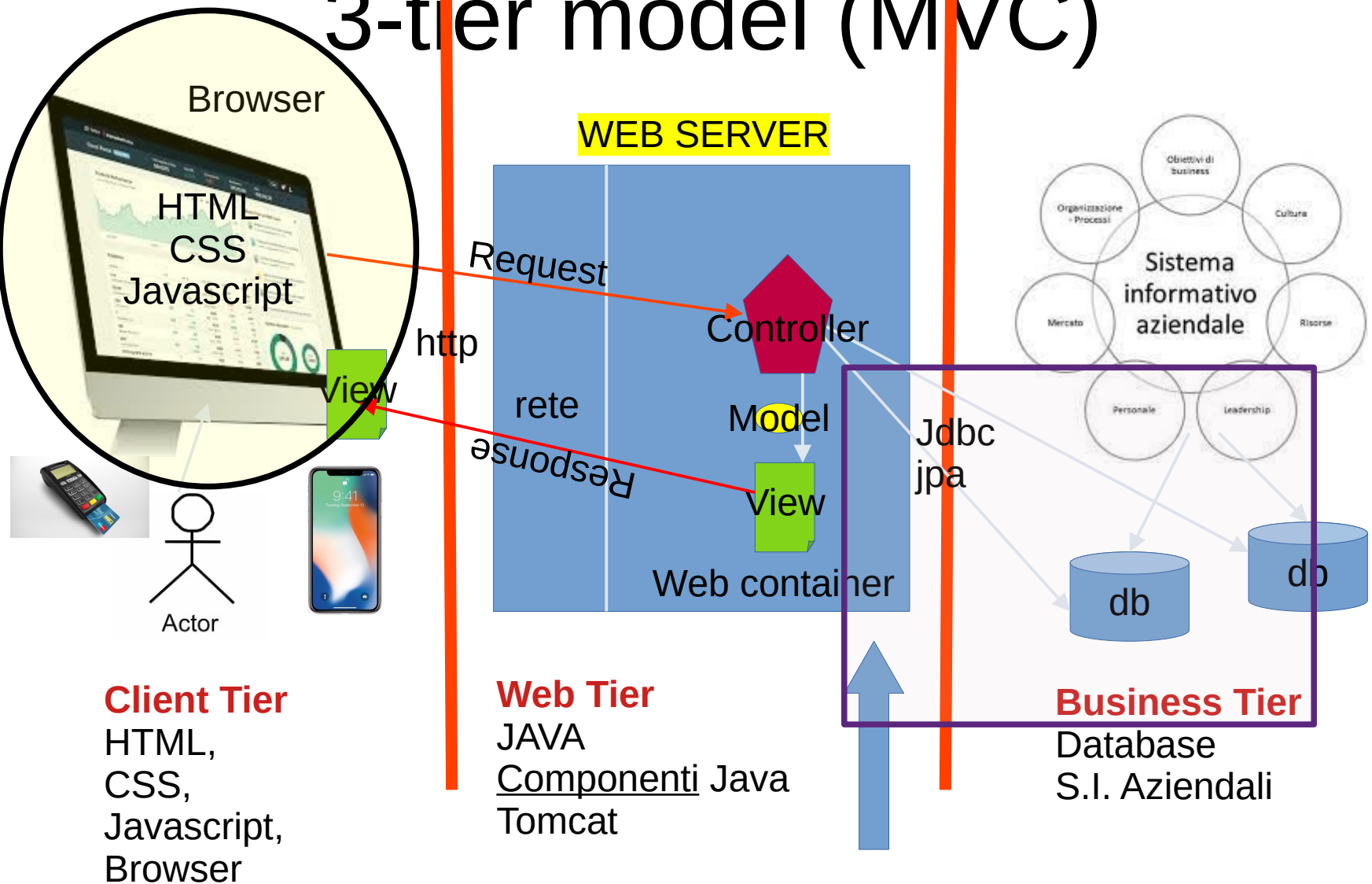
```
ArrayList list = new ArrayList();  
list.add(0, new Integer(42));  
int total = ((Integer)list.get(0)).intValue();
```

Using generic collections:

```
ArrayList<Integer> list = new ArrayList<Integer>();  
list.add(0, new Integer(42));  
int total = list.get(0).intValue();
```

Applicazioni aziendali via WEB

3-tier model (MVC)



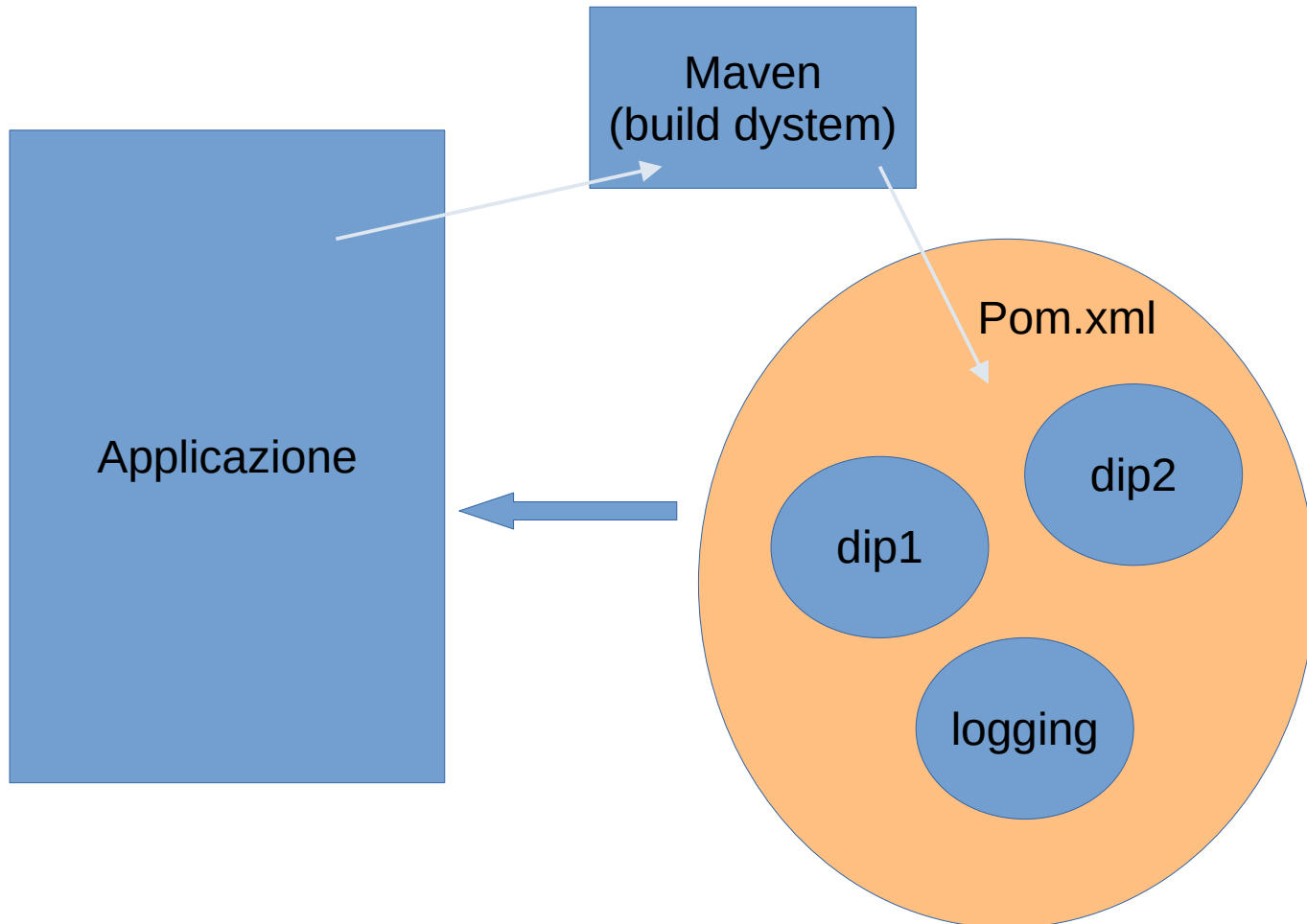
Datacenter



@Test

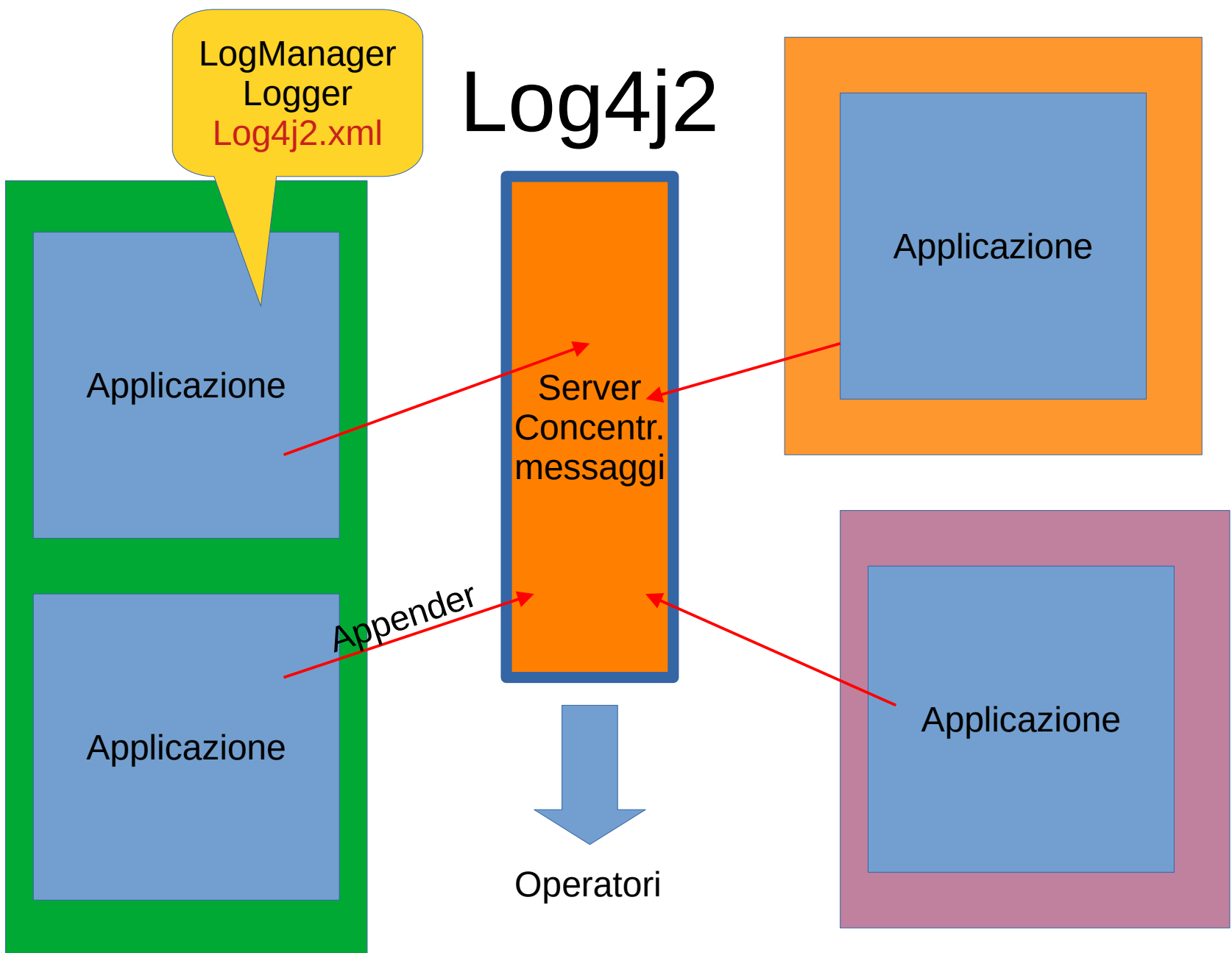
- Test come strumento di validazione
- Test → **formali e ben definiti**
- Test → **ripetibili** (anche automaticamente) da ogni persona
- => **Junit** (v.4 e v.5 jupiter) per fare i test

Maven / Gradle

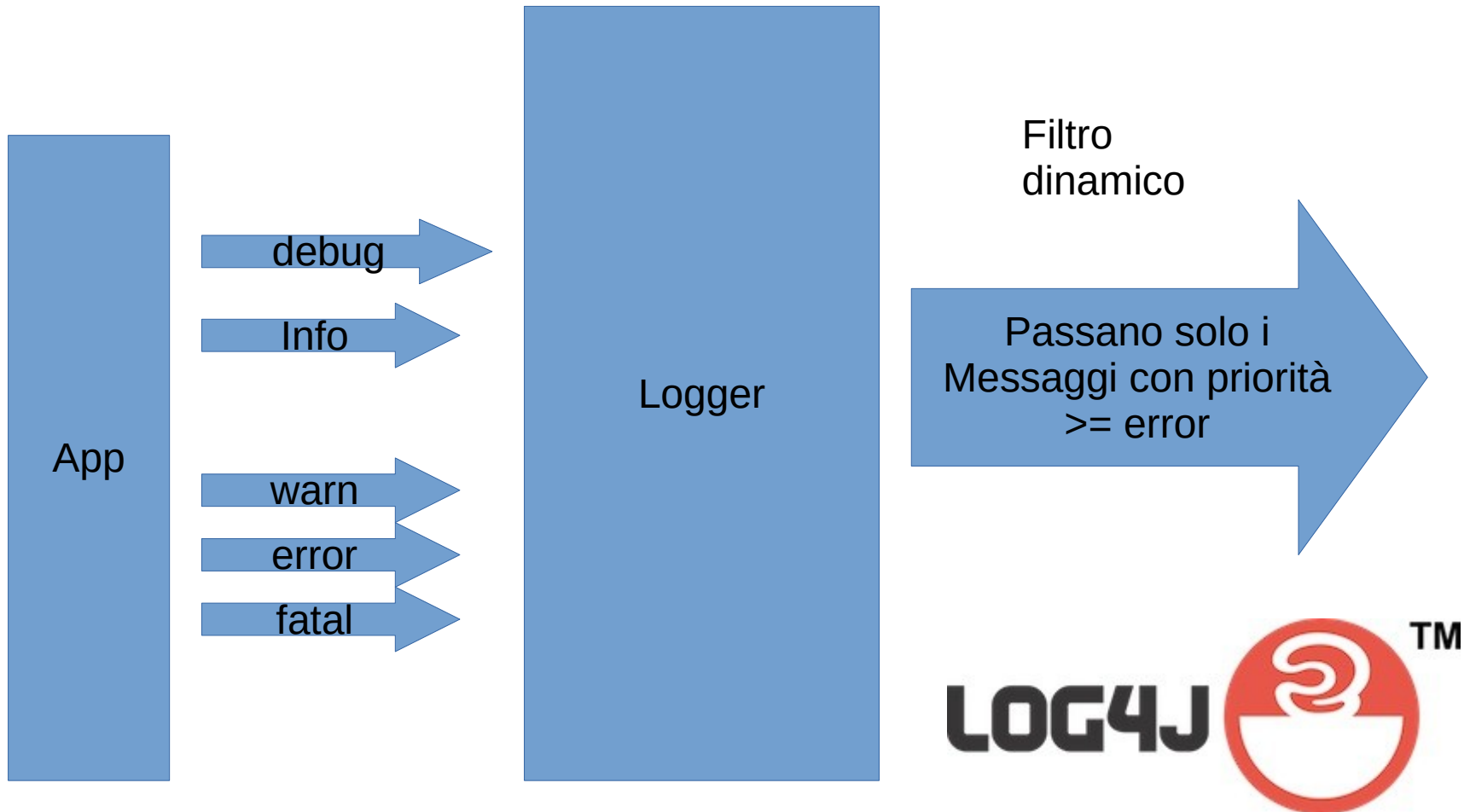


LogManager
Logger
Log4j2.xml

Log4j2



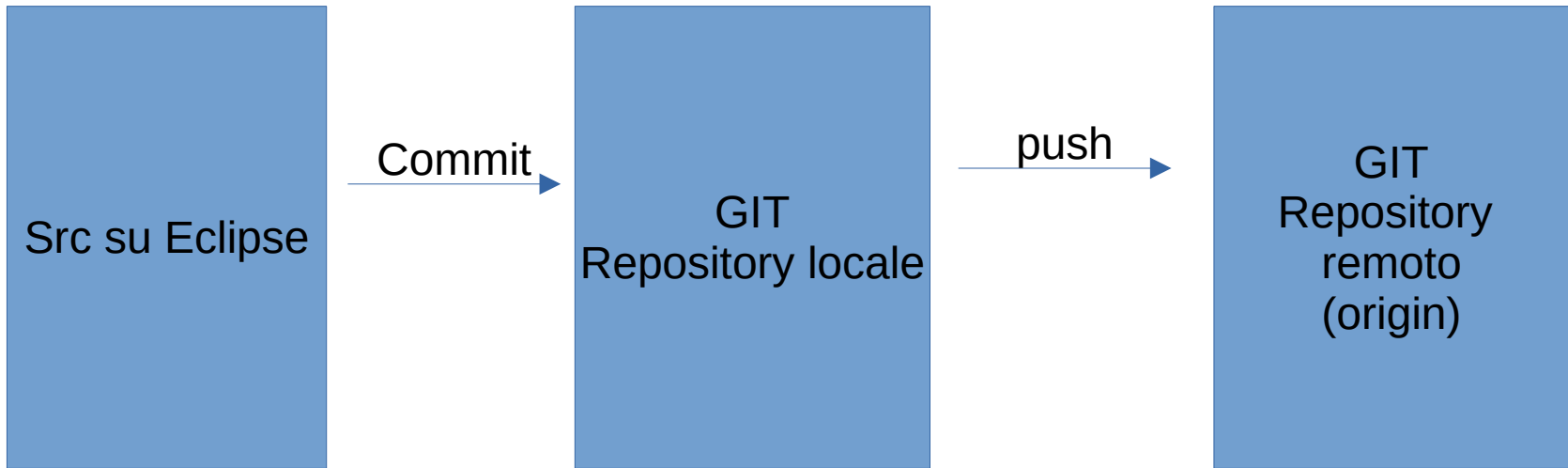
Logger (applicazione)



Esercizio

- Creare il progetto maven di nome “db-access”
 - Predisposto per il logging
 - Predisposto per il testing con Junit Jupiter 5
 - Creare Prova.java per verificare il log
 - Creare ProvaTest.java per verificare Junit

Git



GitHub

- 1) Aprire una User ID
- 2) Generare il Token
- 3) Su Eclipse aprire la prospettiva GIT
- 4) Creare un nuovo repository locale
selezionando come repository la cartella
del progetto da catalogare
- 5) Effettuare la commit del progetto Eclipse

GitHub

- 6) Modificare la voce “Remote” del repository locale configurando il repository remoto (utente=git password=token_generato)
- 7) Effettuare la push

GitHub

...or create a new repository on the command line

```
echo "# repo001" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/didattica-forever/repo001.git
git push -u origin main
```

de 43

...or push an existing repository from the command line

```
git remote add origin https://github.com/didattica-forever/repo001.git
git branch -M main
git push -u origin main
```