

OGGETTI COMPOSTI

- Una classe può contenere *referimenti a altre classi* (o anche a se stessa):

```
public class Orologio {
    Counter ore, minuti;
}
```

- L'oggetto Orologio ("contenitore") può usare gli oggetti Counter ore e minuti...
- ..ma non può accedere ai loro **campi privati**

Oggetti Composti - 1

OGGETTI COMPOSTI - COSTRUZIONE

- In fase di costruzione, il costruttore dell'oggetto "contenitore" deve costruire esplicitamente, con new, gli oggetti "interni"

```
public class Orologio {
    Counter ore, minuti;

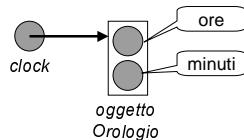
    public Orologio() {
        ore = new Counter(0);
        minuti = new Counter(0);
    }
}
```

Oggetti Composti - 2

OGGETTI COMPOSTI - COSTRUZIONE

- Quindi:
 - prima si costruisce l'oggetto contenitore (fase 1)
 - ...

```
Orologio clock = new Orologio();
```

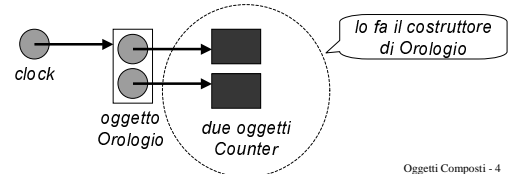


Oggetti Composti - 3

OGGETTI COMPOSTI - COSTRUZIONE

- Quindi:
 - prima si costruisce l'oggetto contenitore (fase 1)...
 - poi esso costruisce gli oggetti interni (fase 2)

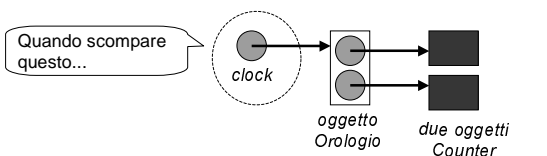
```
ore = new Counter(0);
minuti = new Counter(0);
```



Oggetti Composti - 4

OGGETTI COMPOSTI - DISTRUZIONE

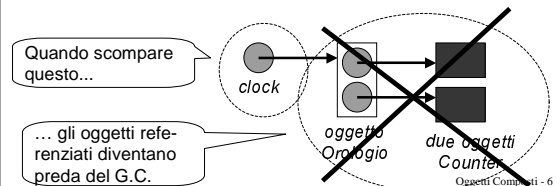
- In fase di distruzione:
 - quando il riferimento all'oggetto contenitore viene distrutto, l'oggetto passa al garbage collector, che lo distruggerà quando vorrà
 - la stessa fine fanno gli oggetti referenziati da esso



Oggetti Composti - 5

OGGETTI COMPOSTI - DISTRUZIONE

- In fase di distruzione:
 - quando il riferimento all'oggetto contenitore viene distrutto, l'oggetto passa al garbage collector, che lo distruggerà quando vorrà
 - la stessa fine fanno gli oggetti referenziati da esso



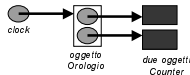
Oggetti Composti - 6

OGGETTI COMPOSTI - USO

- Solo i metodi dell'oggetto "contenitore" possono accedere agli oggetti "interni"

```
public class Orologio {
    Counter ore, minuti;

    public void tic() {
        minuti.inc();
        if (minuti.getValue() == 60) {
            minuti.reset(); ore.inc();
            if (ore.getValue() == 24) ore.reset();
        }
    }
}
```



Oggetti Composti - 7

OGGETTI COMPOSTI - DISTRUZIONE

- In fase di distruzione:

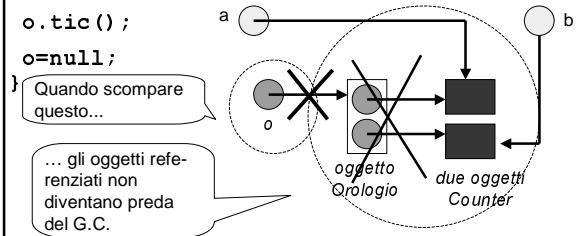
– quando il riferimento all'oggetto contenitore viene distrutto, l'oggetto passa al garbage collector, che lo distruggerà quando vorrà. La stessa fine fanno gli oggetti referenziati da esso (se non referenziati da altri oggetti!!! Ma quando sono referenziati da altri oggetti?)

Oggetti Composti - 8

```
public class Orologio
{
    Counter ore, minuti;
    public Orologio()
    {
        ore = new Counter(0);
        minuti = new Counter(0);
    }
    public Orologio(Counter x, Counter y)
    {
        ore = x;
        minuti = y;
    }
    public void tic()
    {
        ..
    }
}
```

Oggetti Composti - 9

```
public class Programma {
    public static void main(String args[]) {
        Counter a = new Counter(0);
        Counter b = new Counter(b);
        Orologio o = new Orologio(a, b);
        o.tic();
        o=null;
    }
}
```



Oggetti Composti - 10

VANTAGGI

Gli oggetti composti consentono:

- di *aggregare componenti complessi* a partire da componenti più semplici già disponibili
- di mantenere la "unitarietà" del componente, assicurarne la protezione e l'incapsulamento

Oggetti Composti - 11