

## MEMORIA

```
class Fibonacci {
```

```
    int ultimi, penultimo;
```

Rappresenta MEMORIA/STADIO/STORIA  
dell'oggetto

```
    public void next() {
```

```
        int x; // VOLATILE, non è la storia dell'oggetto
```

```
    }
```

lista ContType

legg stringe

increment ContType

richiedi Sale (CONTINUARE)

while Sale == continue

Vincitore il ContType

```
class Contatore {
```

```
    int c; // valore del conteggio
    public Contatore() {
        c=0;
    }
```

```
    public void incrementa()
```

```
    public int getConteggio()
```

```
}
```

```
class Contatore {
    int c; // valore del conteggio
    public Contatore(){
        c=0;
    }
    public void incrementa() {
    }
    public int getConteggio() {
    }
}
```

Dopo aver svolto l'esercizio "conta le stringhe"

Realizzare un programma "conta i numeri", in cui l'operatore inserisce una serie di numeri e al termine il programma deve indicare quanti numeri sono stati inseriti.

Cosa hanno in comune e cosa di diverso questi due esercizi?

Buon lavoro.

Ora è il momento di fare un po' di calcoli:

il programma questa volta deve sommare il valore di tutti i numeri inseriti.

Lo sviluppiamo seguendo lo schema precedente: prima per 'contare' abbiamo realizzato un 'contatore', questa volta per 'sommare' abbiamo bisogno di un 'sommatore'. Quali proprietà deve avere? Come realizzarlo?

COMUNE

HANNO LA  
STESSA CONTATORE

HANNO LA STESSA  
STRUTTURA CICLICA

DIVERSO

GESTIONE { STRINGA  
NUMERO

CHIEDERE IL NUMERO

SOMMARE IL NUMERO CON LA SOMMA  
PRECEDENTE

CHIEDERE SE SI VUOL CONTINUARE

WHILE SI VUOL CONTINUARE

```
class Sommatore {  
    risultato; // somma dei  
               valori precedenti
```

```
    somma
```

```
}
```