

OGGETTI E CLASSI

```
class Studente {  
    String nome;  
    String cognome;  
    int età;  
    boolean italiano;  
}
```

campi
attributi

DICHIARAZIONE DELLA
CLASSE

in un file dal nome
Studente.java

dichiarazione

```
Studente v;
```

crea una variabile di tipo studente
(inizialmente è null)

espressione

```
new Studente ()
```

crea un oggetto di tipo studente
e riempie tutti i campi con il valore
di default (zero, false o null)

```
v = new Studente ()
```

assegna alla variabile v un puntatore
al nuovo studente creato

```
v.nome = "PAOLO";
```

```
v.cognome = "Rossi";
```

```
v.età = 48
```

```
v = new Studente ("Paolo", "Rossi", 48, true)
```



visto nella esercitazione di ieri da parte di alcuni
studenti ... purtroppo non funziona.

```
public static Studente creaStudente (String nome, String cognome,  
int ete, boolean italiano) {
```

campi
obiett. oggetto

```
Studente s = new Studente ();
```

```
s.nome = nome;
```

```
s.cognome = cognome;
```

```
s.ete = ete;
```

```
s.italiano = italiano;
```

```
return s;
```

```
}
```

parametri del metodo:

In un senso generico si
può chiamare costruttore.

Effettivamente è un metodo che possiamo
usare per costruire un oggetto

```
creaStudente ("Robb", "Rossi", 42, true)
```

Un "vero" costruttore ci consentirebbe di scrivere così:

```
new Studente ("Robb", "Rossi", 42, true)
```

Italiano

```
public static Studente creaStudente (String nome, String cognome, int ete,  
boolean italiano) {
```

```
Studente s = new Studente ();
```

```
s.nome = nome;
```

```
s.cognome = cognome;
```

```
s.ete = ete;
```

```
s.italiano = italiano;
```

```
return s;
```

```
}
```

"Costruttore" per gli
studenti italiani

Studente []

array di studenti

Esercizio

Scrivere un metodo che prende un array di studenti, un intero x , e restituisce un nuovo array con tutti gli studenti che hanno x anni o più.

```
public static Studente[] trovaStudentiAnziani (
    Studente[] arr,
    int x
```

potete scrivere tutto sulle stesse righe;

Se ci fosse un modo per allungare un array, non sarebbe necessario questo conteggio preliminare

```
) {
    // Siccome non sappiamo quanti sono gli studenti che hanno
    // più di  $x$  anni, prima li contiamo
    int conteggio = 0;
    for (Studente s: arr) {
        if (s.eta >= x) conteggio++;
    }
    // conteggio contiene il numero degli over  $x$ 
```

variabile che possiamo chiamare come vogliamo

```
Studente[] studentiAnziani = new Studente[conteggio];
```

// riempiamo l'array studenti Anziani

```
int conteggio2 = 0;
for (Studente s: arr) {
    if (s.eta >= x) {
        studentiAnziani[conteggio2] = s;
        conteggio2++;
    }
}
return studentiAnziani; }
```

VERSIONE COL FOR CLASSICO INVECE DI FOR-EACH

```
public static Studente[] trovaStudentiAnziani (  
    Studente[] err,  
    int x  
) {
```

// Siccome non sappiamo quanti sono gli studenti da hanno
// più di x anni, prima li contiamo

```
int conteggio = 0;
```

```
for (int i=0; i < err.length; i++) {
```

```
    if (err[i].eta >= x) conteggio++;  
}
```

Studente s = err[i];
if (s.eta >= x) ...

// conteggio contiene il numero degli over - x

```
Studente[] studentiAnziani = new Studente [conteggio];
```

// riempiamo l'array studenti Anziani

```
int conteggio2 = 0;
```

```
for (int i=0; i < err.length; i++) {
```

```
    if (err[i].eta >= x) {
```

```
        studentiAnziani [conteggio2] = err[i];
```

```
        conteggio2++;  
    }  
}
```

Indica una posizione
nell'array err

Indica la posizione nell'array
studenti Anziani dove copiare il prossimo
studente anziano.

PARAMETRI DEL METODO MAIN

```
public static void main (String[] args) {
```

```
}
```