

# Il ciclo `for` generalizzato

- Il ciclo `for` generalizzato scandisce tutti gli elementi di un array (si usa anche per altri tipi di raccolta, che si vedranno in seguito):

```
double[] lista = . . .;
double somma = 0;
for (double e : lista) // si legge "per ogni e in lista"
{
    somma = somma + e;
}
```

# Il ciclo `for` generalizzato

- Per scandire tutti gli elementi di un array non è obbligatorio utilizzare il ciclo `for` generalizzato: lo stesso ciclo può essere realizzato con un `for` normale e una variabile indice esplicita.

```
double[] lista = . . .;
double somma = 0;
for (int i = 0; i < lista.length; i++)
{
    double e = lista[i];
    somma = somma + e;
}
```

# Sintassi: Il ciclo `for` generalizzato

```
for (<TipoElArray> <Identificatore> : <IdeArray>)  
    <Comando>
```

## Esempio:

```
for (double e : lista)  
    somma = somma + e;
```

## Semantica:

Eseguire un ciclo avente un'iterazione per ogni elemento appartenente a un array. All'inizio di ciascuna iterazione viene assegnato alla variabile l'elemento successivo dell'array (a partire dal primo), poi viene eseguito il comando.

# Il ciclo for generalizzato

- ATTENZIONE: Il ciclo for generalizzato serve solo per la scansione e non per la modifica: es. non permette di azzerare gli elementi di un array

```
double[] lista = . . .;
for (double e : lista) //
{
    e = 0;    // NO
}
```

```
double[] lista = . . .;
for (int i = 0; i < lista.length; i++)
{
    double e = lista[i];

    e = 0;    // il valore di lista[i] non cambia
}
```

# Il ciclo for generalizzato

- ATTENZIONE: Il ciclo for generalizzato serve solo per la scansione e non per la modifica

```
double[] lista = . . .;
for (double e : lista) //
{
    e = 0; // NO
}
```

```
double[] lista = . . .;
for (int i = 0; i < lista.length; i++)
{
    lista[i] = 0; // SI
}
```