

# Classe ArrayList

- La classe ArrayList consente la memorizzazione di una raccolta di oggetti, esattamente come un array, ma con due significativi vantaggi:
  - La dimensione di un vettore può aumentare o diminuire, in base alle necessità
  - La classe ArrayList fornisce metodi per svolgere le operazioni più comuni come la rimozione e l'inserimento di un elemento

Dichiarazione Es:

```
ArrayList<String> nomi= new ArrayList<String>();
```

Il tipo ArrayList<String> specifica un vettore di stringhe

# Classe ArrayList

- I tipi primitivi non possono essere utilizzati come tipo parametro
- La dimensione iniziale di un vettore dopo la sua costruzione è 0.
- Bisogna usare il **metodo add** per aggiungere elementi nel vettore
- **Es:** `nomi.add("Marco");` aggiunge Marco al vettore nomi che ora ha dimensione 1
- Per ispezionare il valore di un elemento si usa il **metodo get**
- **Es:** `nomi.get(0);` restituisce l'elemento avente indice 0
- Per assegnare un nuovo valore a un elemento di un vettore si usa il **metodo set**
- **Es:** `nomi.set(0,"Giovanni");` sostituisce l'elemento avente indice 0 con la stringa "Giovanni"

# Classe ArrayList

- Si può anche inserire un elemento in una data posizione intermedia all'interno di un vettore:
  - **Es:** `nomi.add(0,"Laura")` sposta il primo elemento al secondo elemento e mette a primo elemento la stringa "Laura"
- Al contrario il **metodo remove** elimina l'elemento che si trova in una determinata posizione, sposta di una posizione all'indietro tutti gli elementi che si trovano dopo l'elemento rimosso
  - **Es:** `nomi.remove(0)`
- Il **metodo size** restituisce la dimensione del vettore
  - **Es:** `nomi.size()`