ARRAY

CARATTERISTICHE GENERALI

- Struttura dati (è un oggetto)
- Permette di una sequenza ordinata di valori
  - sia dati fondamentali che oggetti
- Può essere utilizzato anche come parametro esplicito
- Sintassi:

```
NomeTipo[] riferimentoArray = new NomeTipo[lunghezza]

riferimentoArray[indice]

riferimentoArray.length; // lunghezza
```

Eccezioni

- Indice sbagliato: `ArrayIndexOutOfBoundsException`

INIZIALIZZAZIONE DI UN ARRAY

```
int[] primes = {2,3,5};
```

COPIARE UN ARRAY

- Usare `System.arraycopy()`

```
System.arraycopy(sourceArray, firstSourceArrayIndex, targetArray, firstTargetArr
```

- Usare `clone()`
  - serve fare il cast

```
double[] otherValues = (double[]) values.clone()
```

EFFETTUARE RESIZE DI UN ARRAY

```java
public static int[] resizeIntArray(int[] oldArray, int newLength) {
    if (newLength < 0 || oldArray == null)
        throw new IllegalArgumentException();

    int[] newArray = new int[newLength];

    int n = Math.min(oldArray.length, newLength);

    for (int i = 0; i < n; i++)
        newArray[i] = oldArray[i];

    return newArray;
}
```

ARRAY RIEMPITI SOLO IN PARTE

- Utilizzo un array molto grande ma lo riempio solo in parte
  - come faccio a sapere fino a che valore è stato riempito?
- Tengo traccia dell'indice fino alla quale lo ho riempito (creando variabile `int size`)

STATICI

- Se si supera la dimensione massima, lancia un'eccezione

DINAMICI

- Se si supera la dimensione massima, si effettua un resize
  - raddoppio dimensione di partenza (molto efficiente)

ARRAY DI NUMERI CASUALI

- Utilizzare il metodo `Math.random()`
  - restituisce numero `double` in `[0,1[`

```java
public static int[] randomIntArray(int length, int n) {
    int[] a = new int[length];

    for (int i = 0; i < a.length; i++)
        a[i] = (int) (n * Math.random());

    return a;
}
```

PRINT ARRAY

- È un oggetto, serve costruire la stringa manualmente

```
public static String printArray(int[] v, int vSize) {
    String s = "[";

    for (int i = 0; i < vSize; i++)
        s = s + v[i] + " ";

    s = s + "\b]";

    return s;
}
```

ELIMINAZIONE DI ELEMENTI IN UN ARRAY

ARRAY NON ORDINATO

- Copio l'ultimo elemento dell'array nella posizione dell'elemento da eliminare
- Ridimensionare l'array (o ridurre indice con array riempiti solo in parte)

```
public static void remove(int[] v, int vSize, int index) {
    v[index] = v[vSize - 1];
}
```

ARRAY ORDINATO

- Spostare tutti gli elementi dell'array successivi all'elemento da rimuovere alla posizione inferic
  - dall'indice basso all'alto
- Ridimensionare l'array (o ridurre indice con array riempiti solo in parte)

```
public static void removeSorted(int[] v, int vSize, int index) {
    for (int i = index; i < vSize - 1; i++)
        v[i] = v[i + 1];
}
```

INSERIRE ELEMENTO IN ARRAY

ARRAY ORDINATO

- Ridimensionare array (se è pieno)
  - dall'indice alto al basso

- Spostare tutti gli elementi successivi alla posizione di inserimento

```
public static int[] insert(int[] v, int vSize, int index, int val) {
    if (vSize == v.length)
        v = resize(v, 2 * v.length);

    for (int i = vSize; i > index; i--)
        v[i] = v[i - 1];

    v[index] = val;

    return v;
}
```

TROVARE VALORE IN ARRAY

VALORE MINIMO

- Inizializzare valore candidato con il primo elemento
- Confrontare il candidato con gli elementi rimanenti
- Aggiornare valore candidato se viene trovato valore minore

```
public static int findMin(int[] v, int vSize) {
    int min = v[0];

    for (int i = 1; i < vSize; i++)
        if (v[i] < min) min = v[i];

    return min;
}
```

VALORE MASSIMO

- Inizializzare valore candidato con il primo elemento
- Confrontare il candidato con gli elementi rimanenti
- Aggiornare valore candidato se viene trovato valore maggiore

```java
public static int findMax(int[] v, int vSize) {
    int max = v[0];

    for (int i = 1; i < vSize; i++)
        if (v[i] > max) max = v[i];

    return max;
}
```

ARRAY BIDIMENSIONALI

- Chiamati anche **matrici**
- Ogni elemento è identificato da una coppia di indici

```java
int[][] matrix = new int[][];
```

ARGOMENTI SULLA RIGA COMANDI

- Si possono passare dei parametri quando si esegue il programma
- Verranno contenuti in `String[] args`

```
java MyApp arg1 arg2 arg3
```

ARRAY PARALLELI

- Creare diversi array per contenere dati **fortemente correlati**
  - indici sono tra loro correlati
  - rappresentano proprietà diverse di uno stesso concetto

> **Attenzione**
>
> Non conviene fare array paralleli ma piuttosto meglio array di oggetti!