

Array: reference predefiniti

```
int[] A;
```

reference di un array di interi (inizializzato a null)

```
int[] A = new int[10];
```

```
int n = 10;
```

```
int[] A = new int[n];
```

Creare e inizializzare un array:

```
int[] A = {1, 5, 7, 1, 2, 4};
```

==

```
int[] A = new int[] {1, 5, 7, 1, 2, 4};
```

bidimensionali

```
int[][] M = new int[3][2];
```

```
int[][] M = new int[3][2] { {1, 2}, {3, 4}, {-1, 5} };
```

selezione:

- Come C++

- se l'indice dell'array esce dai limiti della dimensione dell'array, viene sollevata un'eccezione a run time

1) preso in input n (int positivo), memorizzare in una matrice un triangolo rettangolo di asterischi di lato n e stamparlo a video

```
import java.util.Scanner;
```

```
public class Triangolo {
```

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

```
        Scanner sc = new Scanner (System.in);
```

```
        System.out.println(" lunghezza lato? ");
```

```
int n = SC.nextInt();
```

```
char [][] matrix = new char [n+1][n+1];
```

```
for (int i = 0; i < n+1; i++) {
```

```
    for (int j = 0; j < n+1; j++) {
```

```
        if (j < i) {
```

```
            matrix[i][j] = "X";
```

```
        else {
```

```
            matrix[i][j] = " ";
```

```
        }
```

```
    }
```

```
for (int i = 0; i < n+1; i++) {
```

```
    for (int j = 0; j < n+1; j++) {
```

```
        System.out.print(matrix[i][j]);
```

```
        System.out.println();
```

```
    }
```

```
}
```

```
System.out.println(matrix); // non stampa il puntatore alla prima cella ma la reference
```

```
}
```

- ogni array ha un attributo length che rappresenta la lunghezza dell'array

```
int[] A = {1, -2, 3, 0}
```

```
System.out.println(A.length); // 4
```

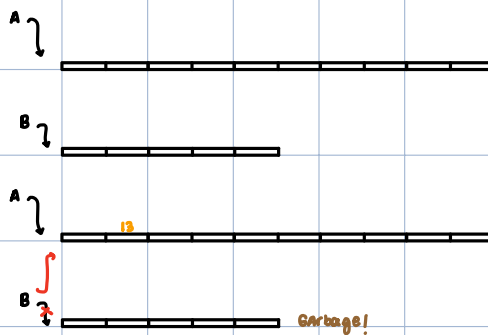
- array sono oggetti, non sono aree contigue di memoria

```
int[] A = new int [10];
```

```
int[] B = new int [5];
```

```
B = A;
```

```
System.out.println ( B[4]); //13
```



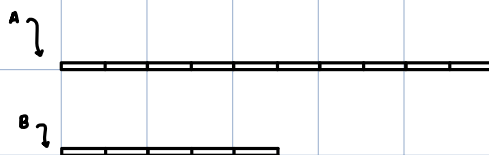
anche gli operatori `==` e `!=` si applicano alle reference

```
A == B
```

false //puntano a
due reference
diverse

```
A != B
```

true



2) Si legga una sequenza lunga max 100 int positivi, terminata da un negativo, lo si memorizzi in un array e si stampi il n max

```
import java.util. Scanner;
```

```
public class Massimo {
```

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

```
        Scanner sc = new Scanner (System.in);
```

```
        int[] array = new array [100];
```

```
        int input;
```

```
        int len = 0;
```

```
        do {
```

```
            input = sc.nextInt();
```

```
            if (input < 0) {
```

```
                break; }
```

```
            array [len] = input;
```

```
            len ++;
```

```
        } while (len < 100);
```

```

sc.close();

array[i] = input; }

int max = a[0];

for (int i = 1; i < len; i++) {

    if (a[i] > max) {

        max = a[i]; }

}

for (int i = 0; i < len; i++) {

    System.out.print (array[i] + " ");

}

System.out.println ();

System.out.println (max);

```

```

}

```

passaggio di parametri:

```

public class Main() {

    public static void f (int[] v) {

        v[0] = 7;

        return; }

    public static void main (String[] args) {

        int [] A = { 1, 2, 3 };

        f(A);

        System.out.println (A[0]);

        return; }

```

```

}

```

n.b. Tipi primitivi: passati per valore

Tipi reference: passati per riferimento

Deallocazione memoria dinamica:

garbage collector: un oggetto viene automaticamente eliminato quando l'oggetto viene deferenziato

3) Tris:

```
import java.util.Scanner;
```

```
public class Tris {
```

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

```
        Scanner sc = new Scanner (System.in);
```

```
        Tris game = new Tris ();
```

```
        boolean player = false ;
```

```
        int row, column;
```

```
        do {
```

```
            if (player) {
```

```
                System.out.println("Giocatore 1: ")
```

```
                row = sc.nextInt();
```

```
                column = sc.nextInt();
```

```
                if (! game.move Player 1 (row, column)) {
```

```
                    System.err.println("Mossa Errata.");
```

```
                    continue; }
```

```
            }
```

```
        else {
```

```
            System.out.println("Giocatore 2 : ")
```

```
            row = sc.nextInt();
```

```
column = sc.nextInt();
```

```
if (! game.movePlayer2(row, column)) {
```

```
    System.err.println("Mossa Errata.");
```

```
    continue; }
```

```
}
```

```
player = ! player;
```

```
game.print();
```

```
} while (! game.someoneWon() && ! game.isFull());
```

```
if (game.getWinner() == -1) {
```

```
    System.out.println("Pareggio!"); }
```

```
else {
```

```
    System.out.println("Vincitore: Giocatore " + game.getWinner());
```

```
}
```

```
}
```