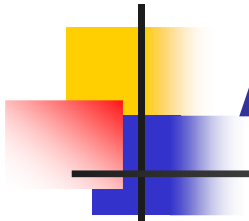
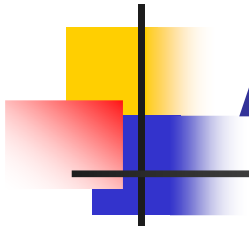


Array

- L'array in Java fornisce il concetto presente nei più comuni linguaggi di programmazione; un array in Java è però un oggetto che estende implicitamente la classe Object
- Un array e' una struttura statica, una volta creato la sua dimensione (numero di elementi che lo compongono) non puo' essere più modificata; per sequenze di lunghezza modificabile, Java fornisce la classe Vector
- L'array puo' contenere elementi che sono tipi primitivi, o oggetti (in realta' riferimenti). In generale gli array sono omogenei, cioe' ogni elemento è dello stesso tipo. Questo limite puo' essere superato con il polimorfismo.
- una variabile di tipo array ammette gli stessi modificatori degli attributi, però che si applicano alla variabile nel suo complesso e non ai singoli elementi dell'array, per i quali non è possibile specificare alcun modificatore



Array

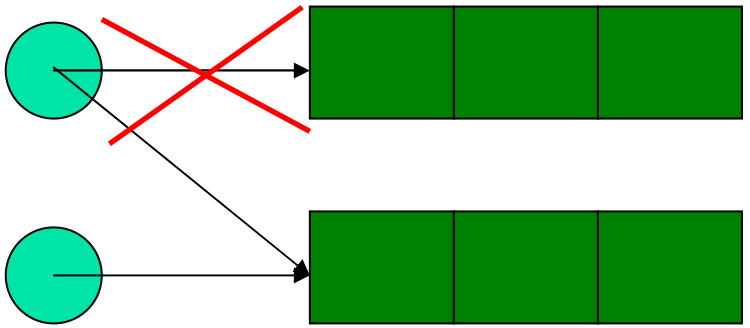
- Gli array Java sono oggetti, istanze di una classe speciale denotata da [ ]; La posizione delle [ ] è a scelta: dopo il nome, come in C, oppure di seguito al tipo:
  <elemType>[ ] <arrID> oppure
  <elemType> <arrID>[ ];
- Esempi:
  int[ ] gradi;
  int gradi[];
  float pressione[];
  boolean[] stato;
- La dimensione si specifica all'atto della creazione:
  gradi = new int[10];
  pressione = new float[100];
  stato = new boolean[15]
- E' possibile la dichiarazione e creazione/inizializzazione implicita:
  int[] x = {10, 100, 90, 50, 45}

Array

- Esempi di assegnazione fra array

int oldcount[] = new int[3];

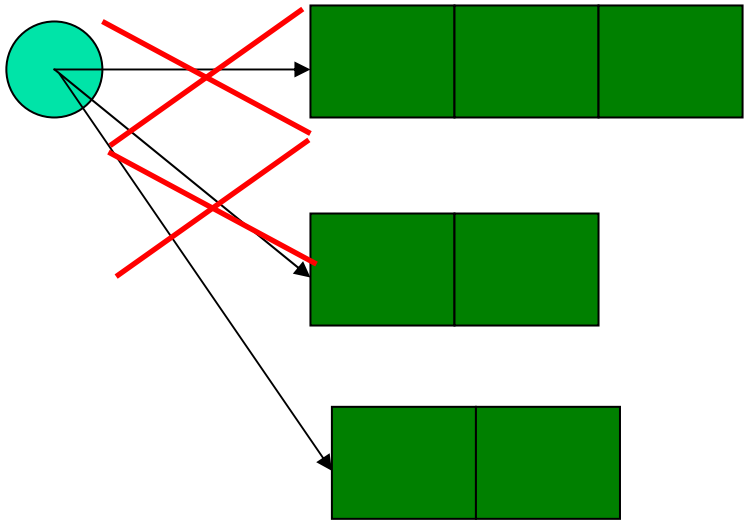
int counts[] = new int[3];

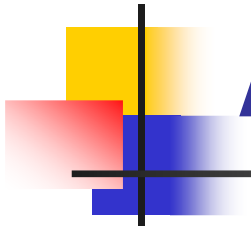


oldCounts = counts;

Array

```
float[] pressione = new float[3];
int i=3;
int j=1;
   while(i>j) {
        pressione = new float[2];
        j++
        }
```



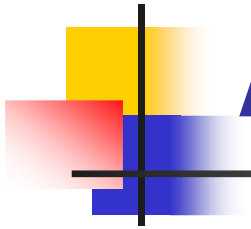


Array

- La dimensione dell'array può essere nota tramite l'attributo length (che nella classe String è invece un metodo length())

```
int[ ] gradi = new int[10];
for (int i=0; i < gradi.length; i++)  {
      gradi[i] = 0;
}
```

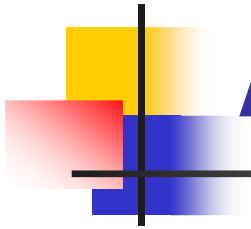
- L'escursione dell'indice dell'array è da 0 a N-1 per N elementi, come in C; length è l'N, quindi una scansione dell'array tramite ciclo può andare da 0 a (array.length)-1



Array

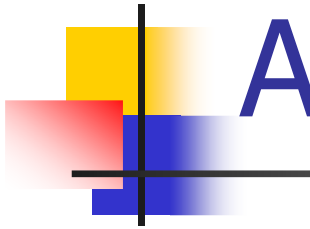
Esempio che mostra la similitudine con il C:

```
public int maggiore(int[ ]  myArray) {
      int massimo = myArray[0];
    int i;
    for (i = 1; i < myArray.length; i++)
        {
           if (massimo < myArray[i])
              massimo = myArray[i];
        }
      return massimo;
}
```



Array

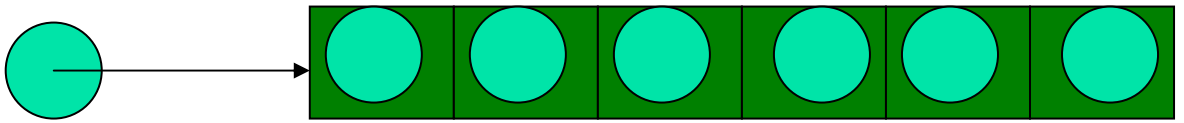
- Se un array è di oggetti, allora:
  - l'identificatore dell'array è un riferimento ad un array di oggetti
  - ogni elemento dell'array è un riferimento a un oggetto della classe specificata come tipo base dell'array
- Istanziare l'array di oggetti non assicura l'istanziazione dei vari oggetti che costituiscono gli elementi dell'array, elementi che quindi devono essere esplicitamente istanziati.

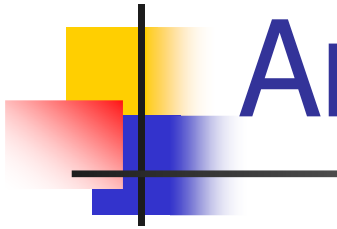


Array

- ogni elemento dell'array è una variabile, se gli elementi dell'array sono di un tipo primitivo (int, float, char, …), ad esempio *v = new int[3];*

- è un riferimento a un (futuro) oggetto, se gli elementi dell'array sono (riferimenti a) oggetti, ad esempio *w = new Counter[6];* presenta 6 oggetti Counter, inizialmente tutti null

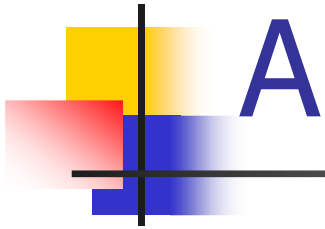




Array

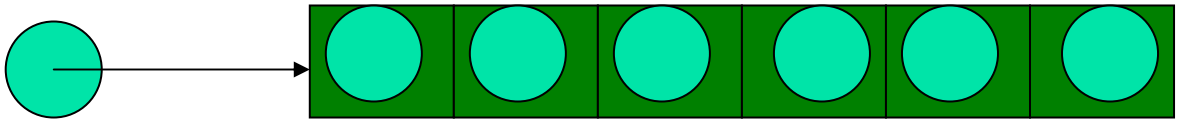
- Nel primo caso ogni elemento dell'array è una normale variabile usabile cosi com'è:

```
v = new int[3];
v[0] = 1; v[1] = 34;
```

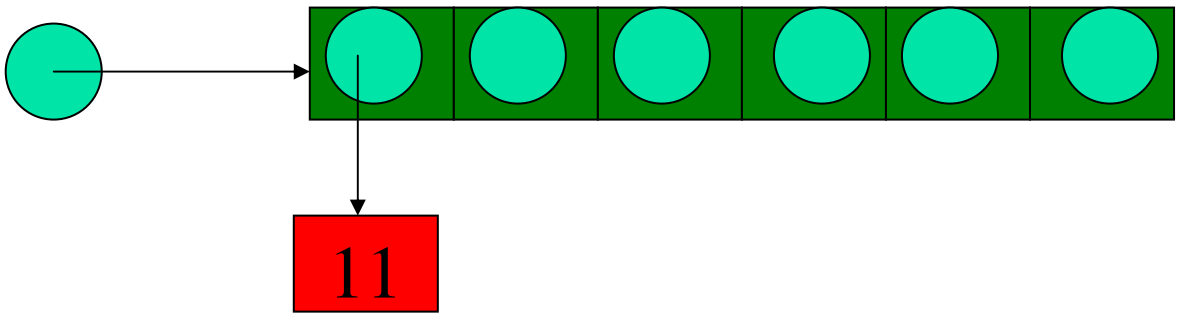


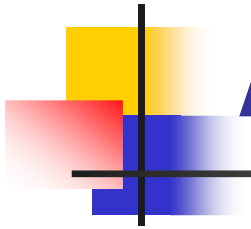
Array

- Nel secondo caso invece, ogni elemento della array è solo un riferimento: se si vuole un nuovo oggetto bisogna crearlo
- w = new Counter[6];



w[0] = new Counter(11);

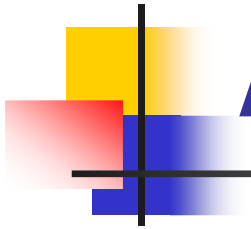




Array

- Esempio di stampa del vettore di argomenti passati dalla linea di comando

```
public class EsempioMain{
public static void main(String[] args){
 if (args.length == 0)
        System.out.println("Nessun argomento");
 else
        for (int i=0; i<args.length; i++)
              System.out.println("argomento " + i
                                    + ": " + args[i]);
        }
}
```



Array

- Dichiarazione – crea solo il riferimento – valore null

```
int myInts[];
int[] myInts;
```

- Istanziazione

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

- dichiarazione e istanziazione

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

- accesso a ciascun elemento

```
myInts[3] = 9;
x = myInts[4];
```

- inizializzazione statica

```
int[] myInts = {1,2,5,6,7,4};
```