# Programmazione 1 - Modulo C

## Array in Java

**Marco Beccuti**

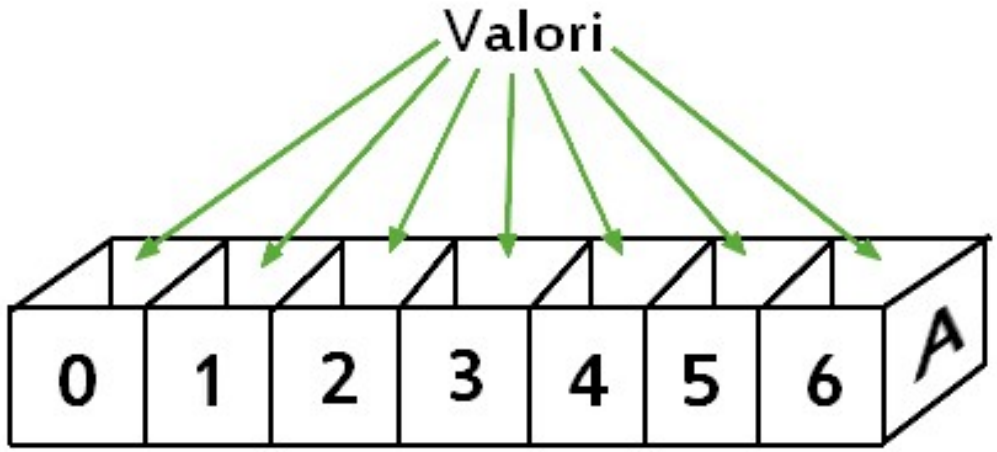
*Università degli Studi di Torino*

*Dipartimento di Informatica*

Novembre 2020

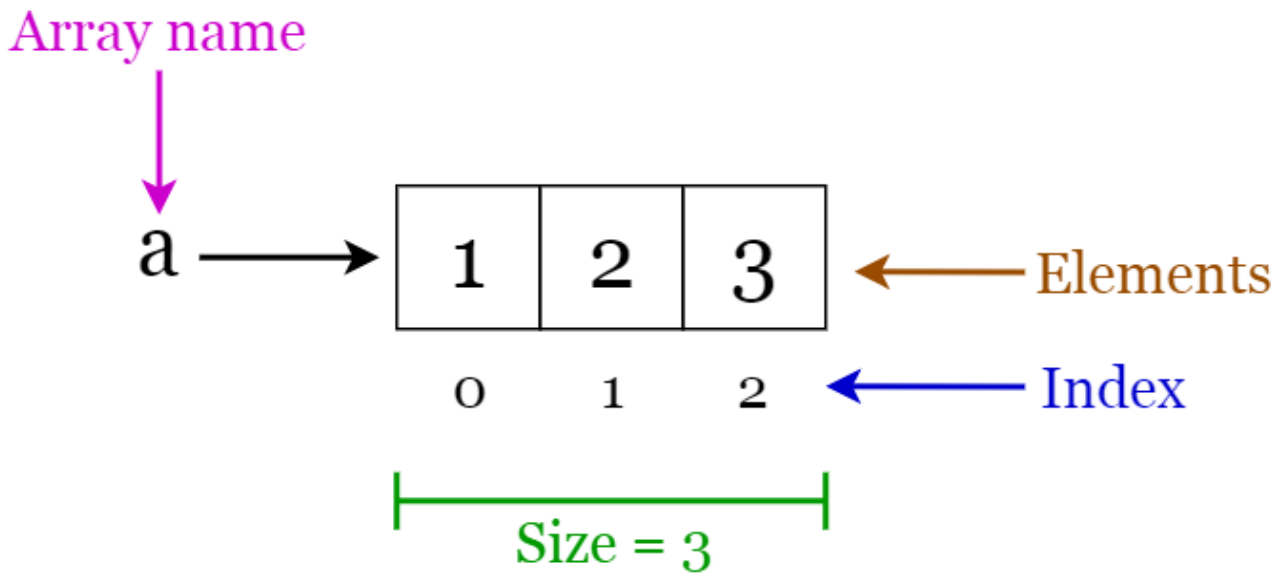
# Array

- All'interno di un programma puó essere necessario gestire collezioni di dati;
- Il modo piú semplice per organizzarli, se sono di tipo omogeneo, é attraverso il concetto di **array** o **vettore**;



# Array in Java (1)

- Oggetti speciali che incapsulano sequenze **ordinate e omogenee** di dati;
- Hanno dimensione **fissa**, definita all'atto della creazione;
- Sintassi **semplificata** per accedere ai singoli elementi;



# Dichiarazione di un array

- L'istruzione

  $$T\ [\ ]\ x;$$

  per qualsiasi tipo $T$, dichiara una variabile $x$ di tipo array di $T$.

- Rispetto a $T$ possiamo definire due gruppi di array:
  - Quelli che modellano sequenze di tipi elementari (interi, reali, caratteri, booleani);
  - Quelli che modellano sequenze di oggetti.

**Per esempio:**

$$int\ [\ ]\ x;$$

dichiara la variabile $x$ con tipo: **array di interi.**

# Creazione di un array

- Vengono creati dinamicamente con **new**:
- L'istruzione

$$T\ [\ ]\ x = new\ T[n];$$

  - **dichiara** la variabile $x$ di tipo array di $T$;
  - **crea** l'array corrispondente di lunghezza n.

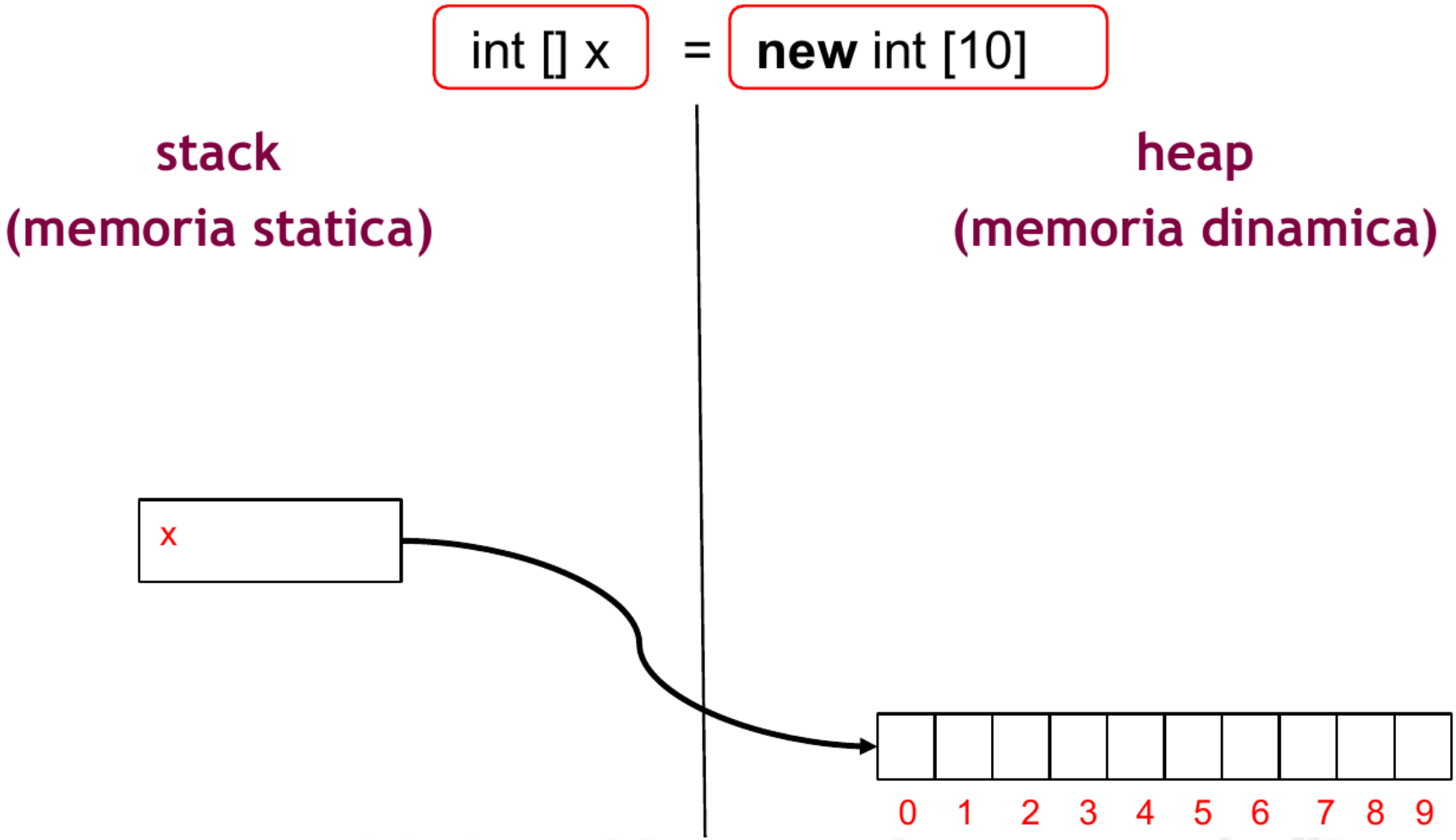
**Per esempio:**

$$int\ [\ ]\ x = new\ int\ [10];$$

dichiara la variabile $x$ con tipo: array di interi e crea il corrispondente array di 10 interi.

- La **lunghezza** dell'array é il valore dell'espressione $x.length$, assegnato al momento della creazione (non della dichiarazione)

# Creazione di un array

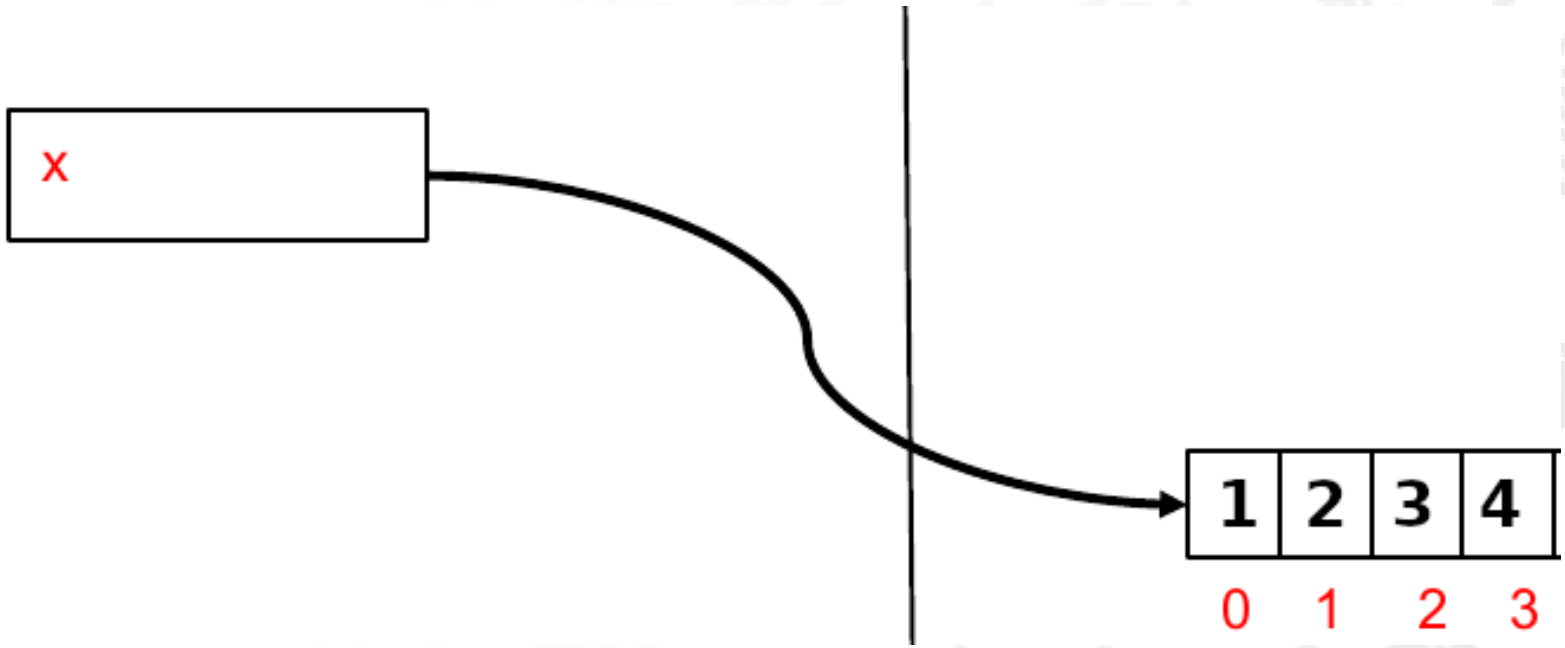



- La dichiarazione di una variabile di tipo array crea nello stack un **riferimento** ad una area dello **heap** che esiste **solo dopo l'esecuzione** della istruzione new, e che contiene gli elementi dell'array.

# Creazione di un array

- A volte si conosce a priori l'intero contenuto dell'array;
- Java offre una sintassi semplificata per la sua inizializzazione:

$$int\ [\ ]\ x = \{1,2,3,4\};$$



# Accesso agli elementi di un array

- Con l'istruzione:

$$int\ [\ ]\ x = new\ int[5]$$

la variabile $x$ si riferisce ad un array di lunghezza 5, le cui posizioni sono 5 variabili di tipo int (il tipo base dell'array) alle quali si accede con la notazione:

$$x[0],\ x[1],\ x[2],\ x[3],\ x[4]$$

**Per esempio:**

$$x[3]=10;$$

aggiorna il valore della cella 3 con il valore 10.

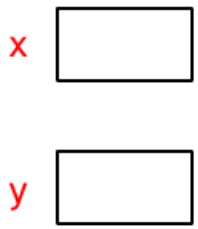
# Aliasing e uguaglianza di array

- il termine **aliasing** indica la situazione in cui una stessa posizione di memoria è associata a nomi simbolici diversi all'interno di un programma.

stack

(memoria statica)

```
int [] x;
int [] y;
```

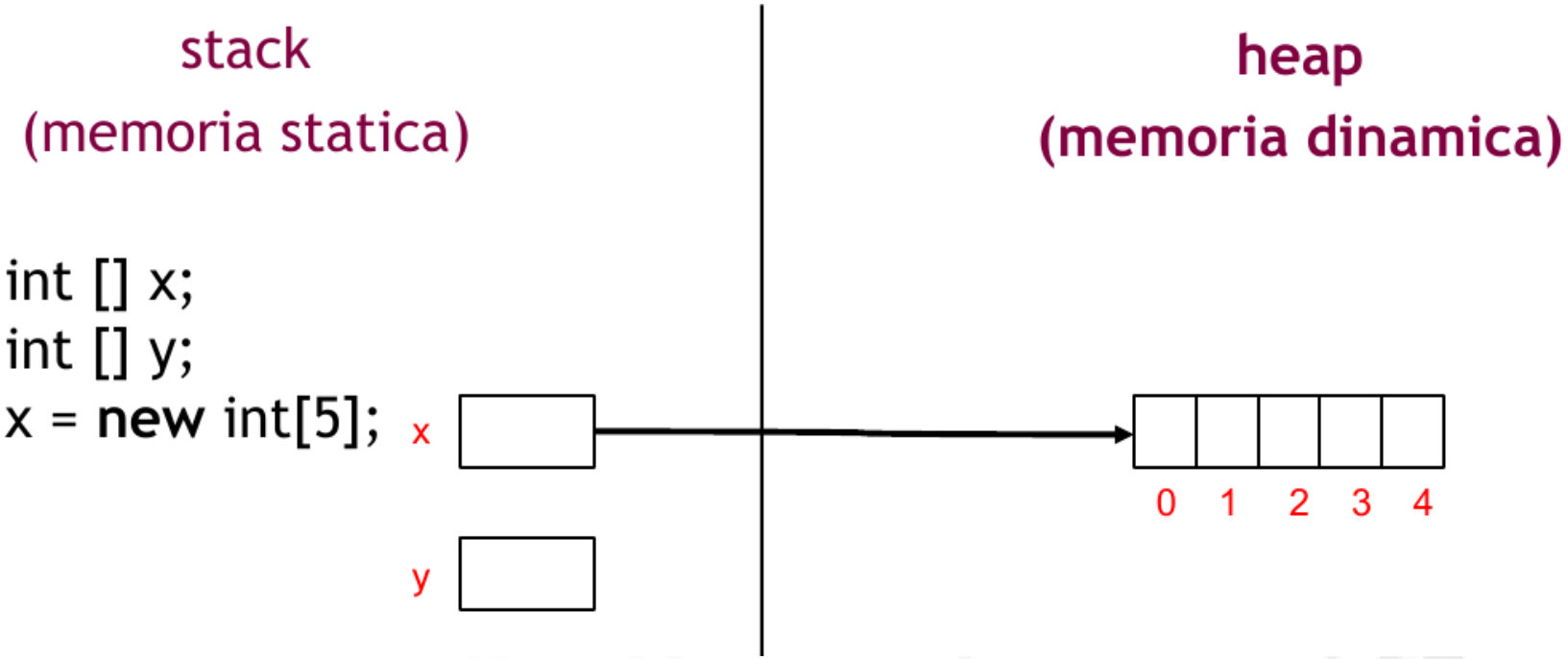



heap

(memoria dinamica)

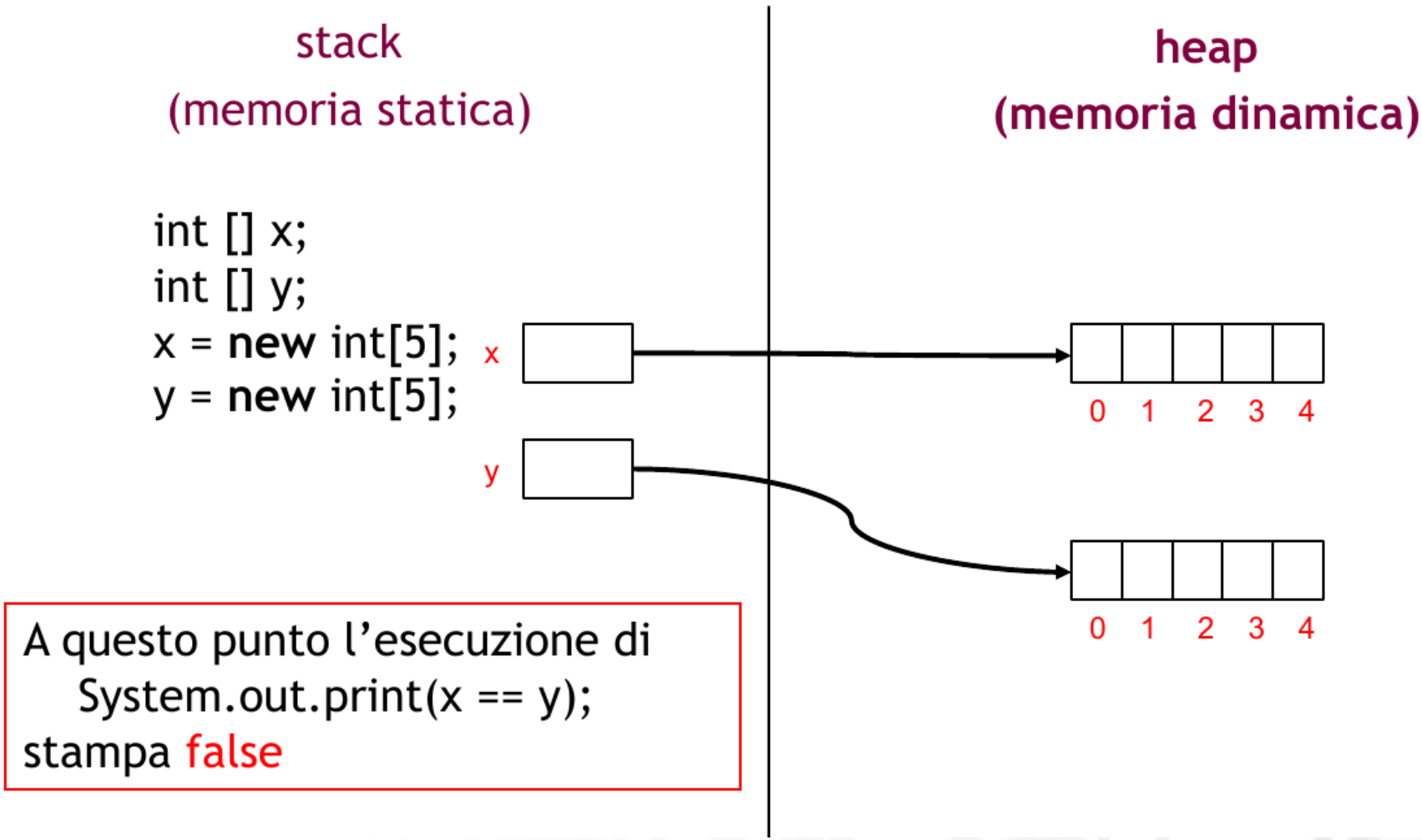
# Aliasing e uguaglianza di array

- il termine **aliasing** indica la situazione in cui una stessa posizione di memoria è associata a nomi simbolici diversi all'interno di un programma.



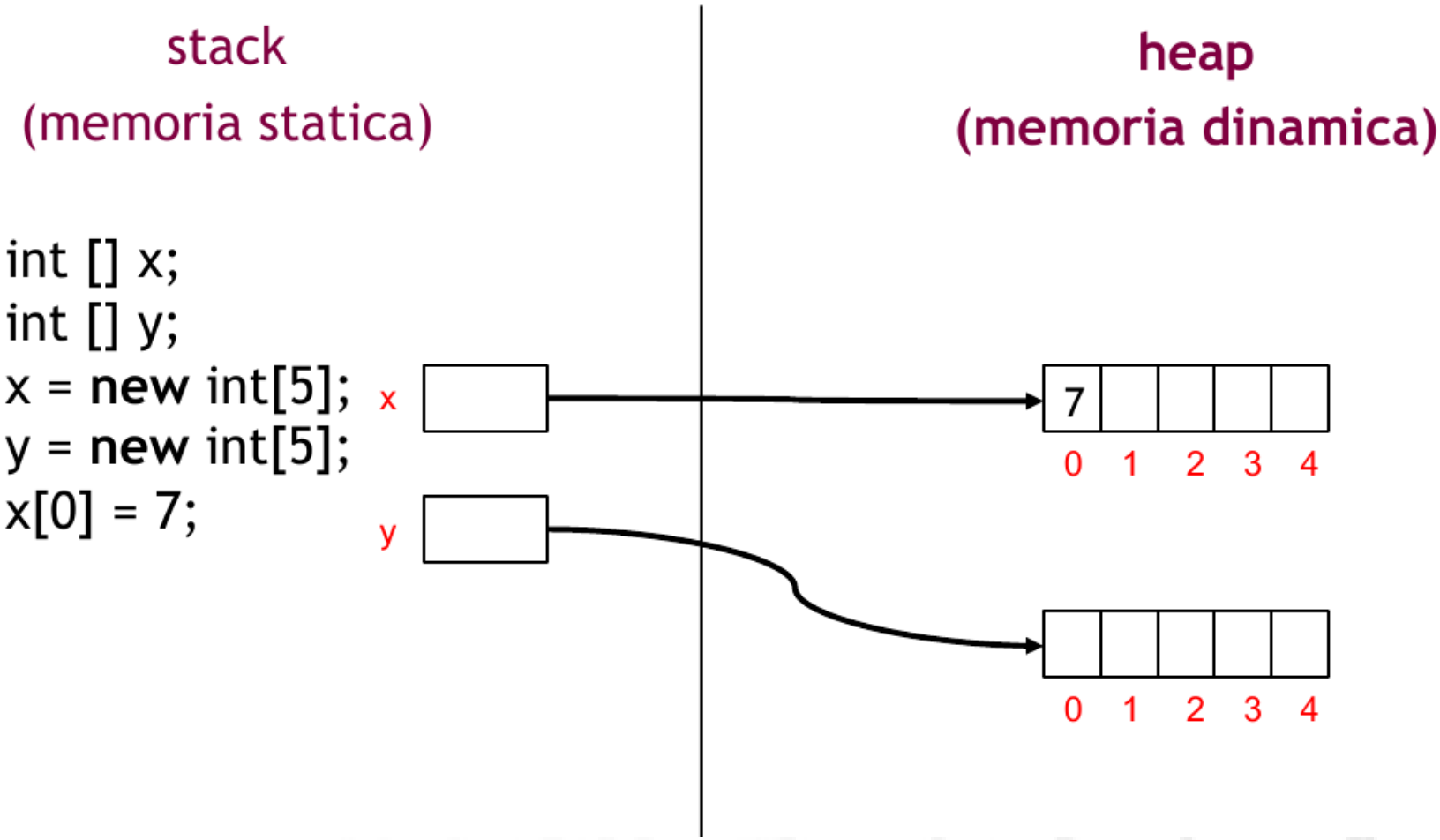
# Aliasing e uguaglianza di array

- il termine **aliasing** indica la situazione in cui una stessa posizione di memoria è associata a nomi simbolici diversi all'interno di un programma.



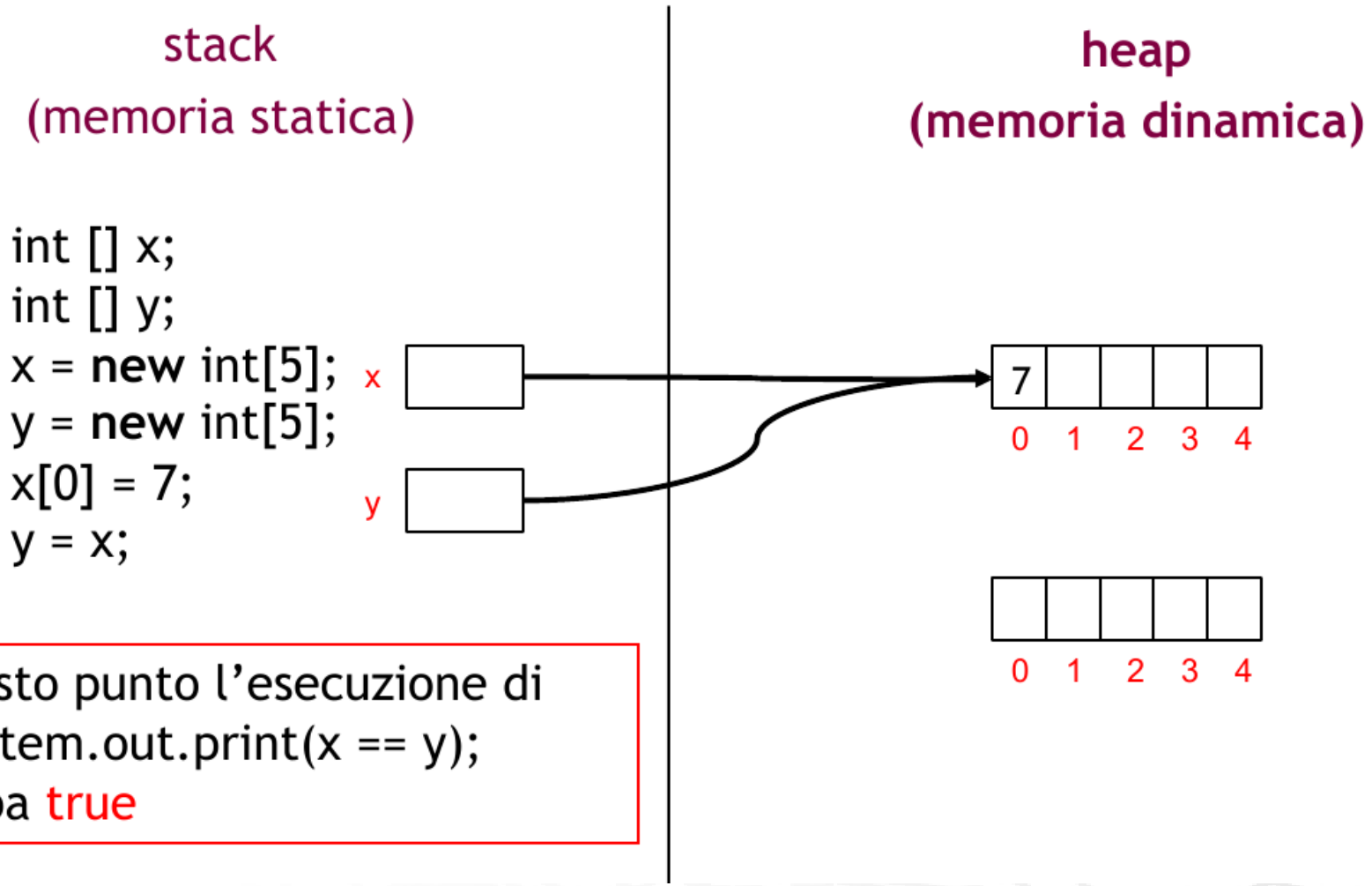
# Aliasing e uguaglianza di array

- il termine **aliasing** indica la situazione in cui una stessa posizione di memoria è associata a nomi simbolici diversi all'interno di un programma.



# Aliasing e uguaglianza di array

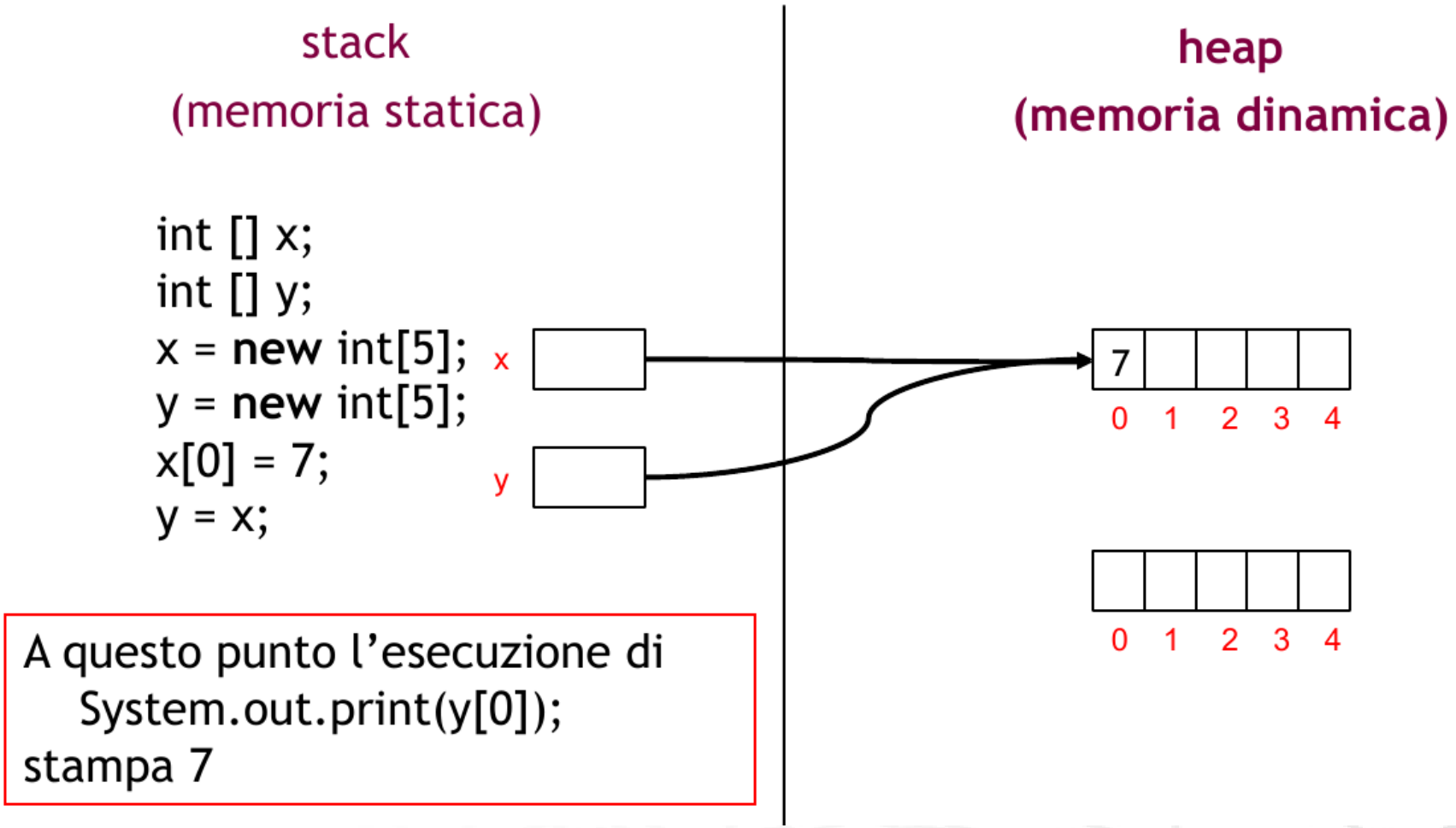
- il termine **aliasing** indica la situazione in cui una stessa posizione di memoria è associata a nomi simbolici diversi all'interno di un programma.




A questo punto l'esecuzione di
`System.out.print(x == y);`
stampa true

# Aliasing e uguaglianza di array

- il termine **aliasing** indica la situazione in cui una stessa posizione di memoria è associata a nomi simbolici diversi all'interno di un programma.



# Quando viene liberata la heap?

- Per costruzione Java non offre al programmatore la possibilitá di deallocare gli oggetti in modo esplicito;
- Il **garbage collector** é un meccanismo che tiene traccia delle locazioni di memoria utilizzate e le libera solo quando non sono effettivamente piú necessarie;
- Il garbage collector é quindi una speciale routine di sistema che scandisce il Java Heap liberando la memoria occupata dagli oggetti non piú referenziati;
- Gli array vengono distrutti automaticamente dal **garbage collector** quando non piú referenziati;

# Accedere ad una cella al di fuori delle dimensioni di un array

- In C avremmo avuto effetti imprevedibili perché non c'é alcuna verifica sugli indici: l'effetto é quello di sporcare la memoria;
- In Java genera un errore in fase di esecuzione.