



06 Masyvai

Jaroslav Grablevski

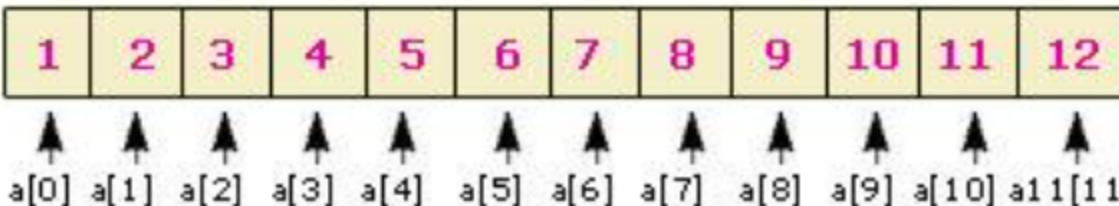
Masyvai

- Masyvai skirti saugoti dideliam kiekiui vienodo tipo reikšmių;
- Masyvo elementai skaičiuojami nuo 0 (žymi pirmą elementą);
- Masyvo dydis nurodomas jį sukuriant ir negali kisti. Jį galima sužinoti per savybę „length“.

One Dimensional array

Initialization `int a[] = new int [12];`

Value



Index

`System.out.print(a[5]);`

Output: 6



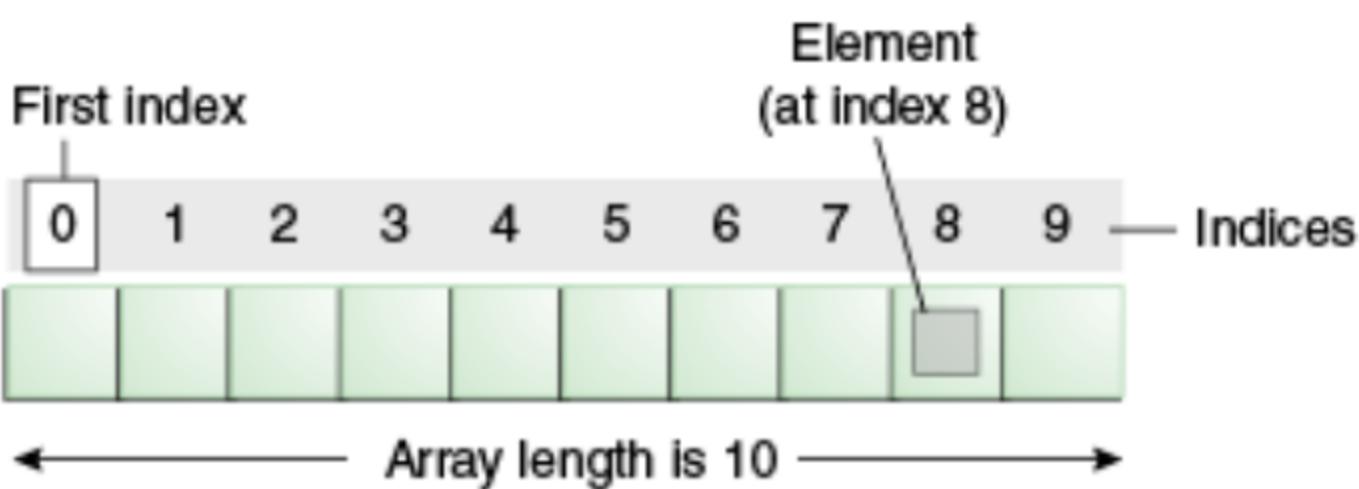
Deklaravimas

```
int[] marks;  
String[] words;  
double[] numbers;  
  
int primes[]; //legalu, bet nerekomenduojama
```



Inicjalizavimas

```
//dataType[] arrayRefVar = new dataType[arraySize];  
  
int[] arr = new int[10]; //Sukurs tuščią masyvą 10čiai skaičių saugoti  
  
//dataType[] arrayRefVar = {value0, value1, ..., valuek};  
  
int[] primes = {1,2,3,5,6,11,13}; //Sukurs 7 elementų masyvą ir užpildys ji  
//nurodytomis reikšmėmis būtent tokią tvarka
```



Reikšmės pagal nutylėjimą

Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (or any object)	null
boolean	false



Reikšmių priskyrimas

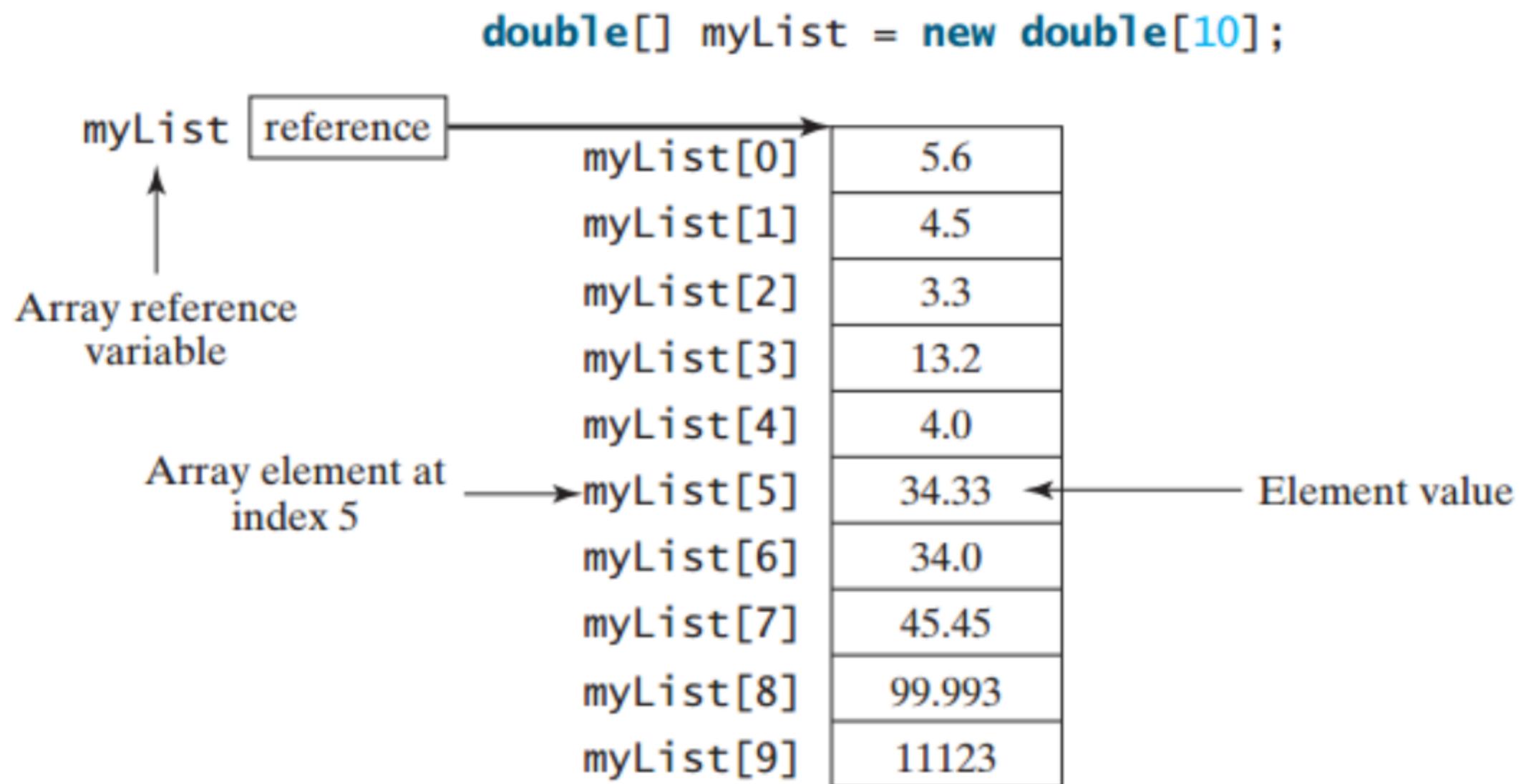
```
// Declare & allocate a 5-element array
int[] marks = new int[5];

// Assign values to the elements
marks[0] = 95;
marks[1] = 85;
marks[2] = 77;
marks[3] = 69;
marks[4] = 66;

System.out.println(marks[0]); // 95
System.out.println(marks[3] + marks[4]); // 135
```



Reikšmių priskyrimas



Masyvo elementų spausdinimas

```
double[] myList = { 1.9, 2.9, 3.4, 3.5 };

// Print all the array elements
for (int i = 0; i < myList.length; i++) {
    System.out.println(myList[i] + " ");
}
```



Masyvo elementų suma

```
double[] myList = { 1.9, 2.9, 3.4, 3.5 };

// Summing all elements
double total = 0;
for (int i = 0; i < myList.length; i++) {
    total = total + myList[i];
}
System.out.println("Total is " + total); // 11.7
```



Didžiausios reikšmės paieška

```
int[] arr = {3, 2, 4, 6, 5, 1};

//finding largest element
int max = arr[0];

for (int i = 0; i < arr.length; i++) {
    if(arr[i] > max) {
        max = arr[i];
    }
}
System.out.println("Max is " + max); //6
```



Didžiausios reikšmės paieška metode

```
public static void main(String[] args) {  
    int[] arr = { 3, 2, 4, 6, 5, 1 };  
    int max = max(arr);  
    System.out.println("Max is " + max);  
}  
  
public static int max(int[] arr) {  
    int max = arr[0];  
    for (int i = 0; i < arr.length; i++) {  
        if (arr[i] > max) {  
            max = arr[i];  
        }  
    }  
    return max;  
}
```



Elementų, tenkinančių sąlygą, kiekis

```
int[] arr = { 3, 2, 4, 6, 5, 1 };

int count = 0;
for (int i = 0; i < arr.length; i++) {
    if (arr[i] > 4) {
        count++;
    }
}

System.out.println("Count is " + count); // 2
```



Elemento indekso paieška

```
public static void main(String a[]) {  
    int[] a1 = { 3, 2, 4, 6, 5, 1 };  
    int key = 5;  
    System.out.println(key + " is found at index: " + linearSearch(a1, key));  
}  
  
public static int linearSearch(int[] arr, int key) {  
    for (int i = 0; i < arr.length; i++) {  
        if (arr[i] == key) {  
            return i;  
        }  
    }  
    return -1;  
}
```



for-each loop

```
for (int i=0; i < array.length; i++) {  
    System.out.println("Element: " + array[i]);  
}  
  
//enhanced for-loop  
  
for (String element : array) {  
    System.out.println("Element: " + element);  
}
```



for-each loop(2)

```
char[] characters = { 'a', 'b', 'c', 'a', 'b', 'c', 'a' };

int counter = 0;
for (char ch : characters) {
    if (ch == 'a') {
        counter++;
    }
}

System.out.println(counter); // it outputs "3"
```



for-each loop (3)

```
//We can do the same thing with the for-loop
char[] characters = { 'a', 'b', 'c', 'a', 'b', 'c', 'a' };

int counter = 0;
for (int i = 0; i < characters.length; i++) {
    if (characters[i] == 'a') {
        counter++;
    }
}

System.out.println(counter); // it outputs "3"
```

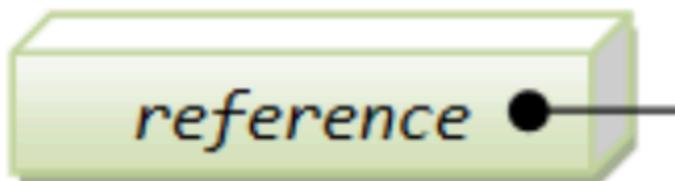


Arrays.toString

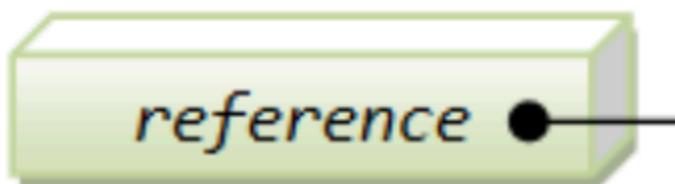
```
int[] arr = { 1, 2, 3, 4, 5 };  
  
System.out.println(arr); //prints [I@15db9742  
  
System.out.println(Arrays.toString(arr));  
//prints [1, 2, 3, 4, 5]
```



aPrimitiveArrayVar



anObjectArrayVar



// Primitive array (values stored directly in heap)

```
int[] aPrimitiveArrayVar = new int[4];
aPrimitiveArrayVar[0] = 10;
aPrimitiveArrayVar[1] = 2;
aPrimitiveArrayVar[2] = 8;
aPrimitiveArrayVar[3] = 57;
```

// Object array (stores references)

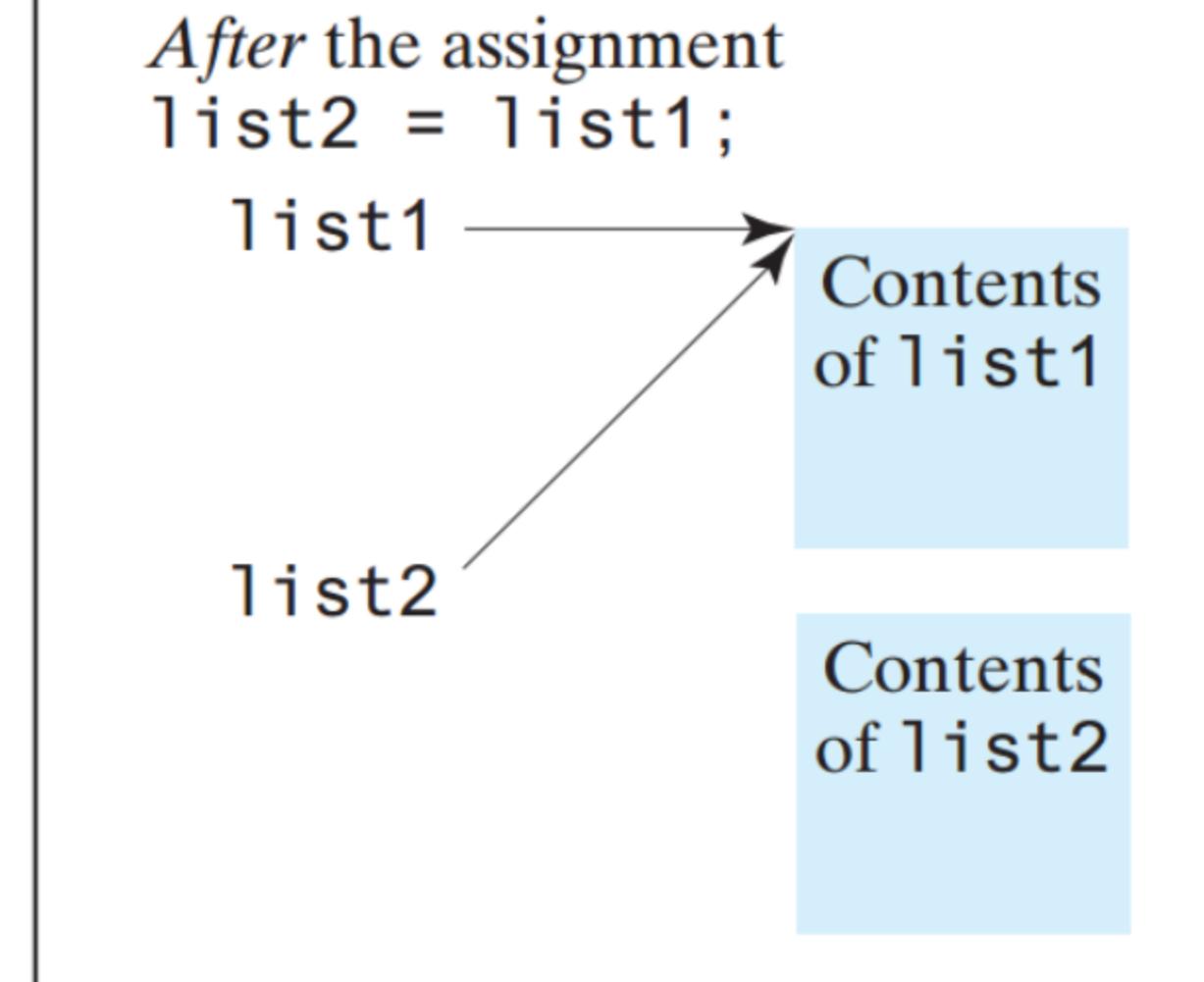
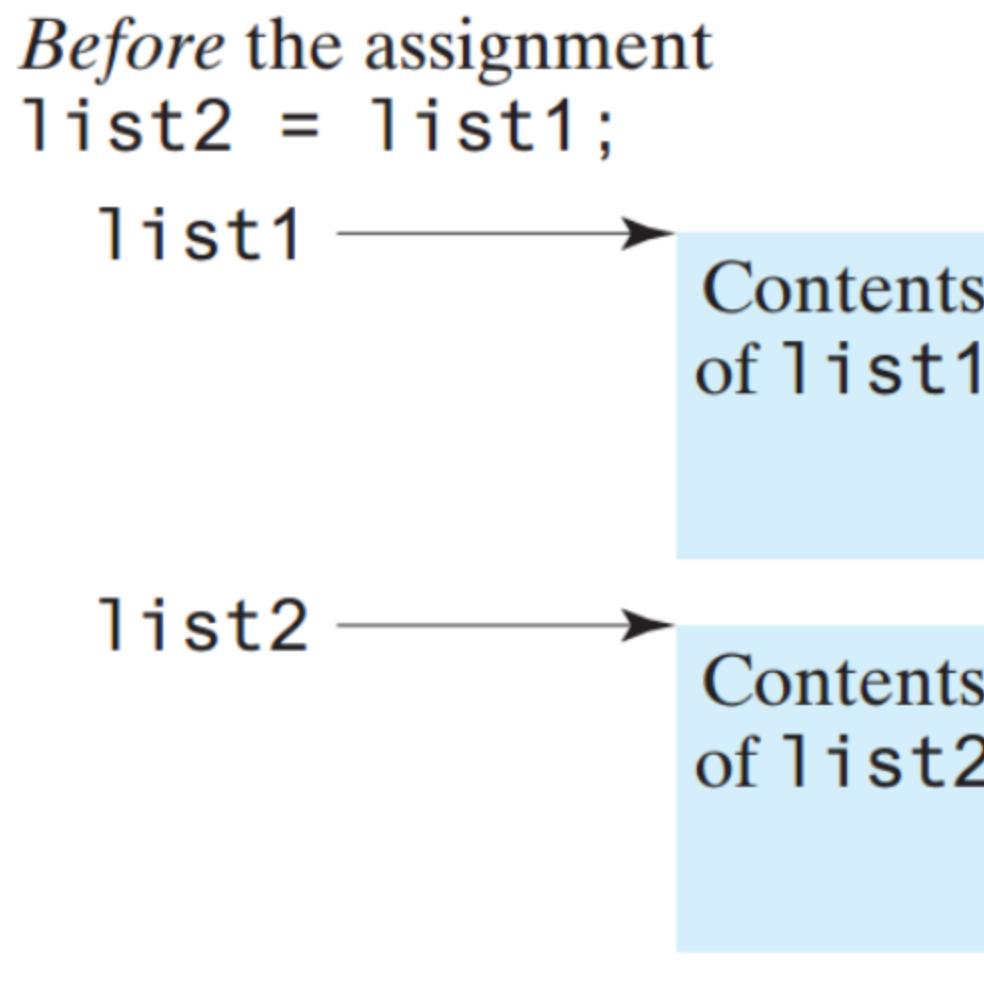
```
String[] anObjectArrayVar = new String[4];
anObjectArrayVar[0] = "Hello";
anObjectArrayVar[1] = "World";
anObjectArrayVar[2] = null; // No object assigned
anObjectArrayVar[3] = new String("Java");
```

// Kitaip tariant, primityvai negali būti null,
// nes jie nėra objektai



Kopijavimas

```
list2 = list1;
```



System.arraycopy

```
int[] arr = { 1, 2, 3, 4, 5 };

int[] copied = new int[10];
// arraycopy(sourceArray, srcPos, targetArray, tarPos, length)
System.arraycopy(arr, 0, copied, 1, 5);

System.out.println(Arrays.toString(arr));
System.out.println(Arrays.toString(copied));
```

```
[1, 2, 3, 4, 5]
[0, 1, 2, 3, 4, 5, 0, 0, 0, 0]
```



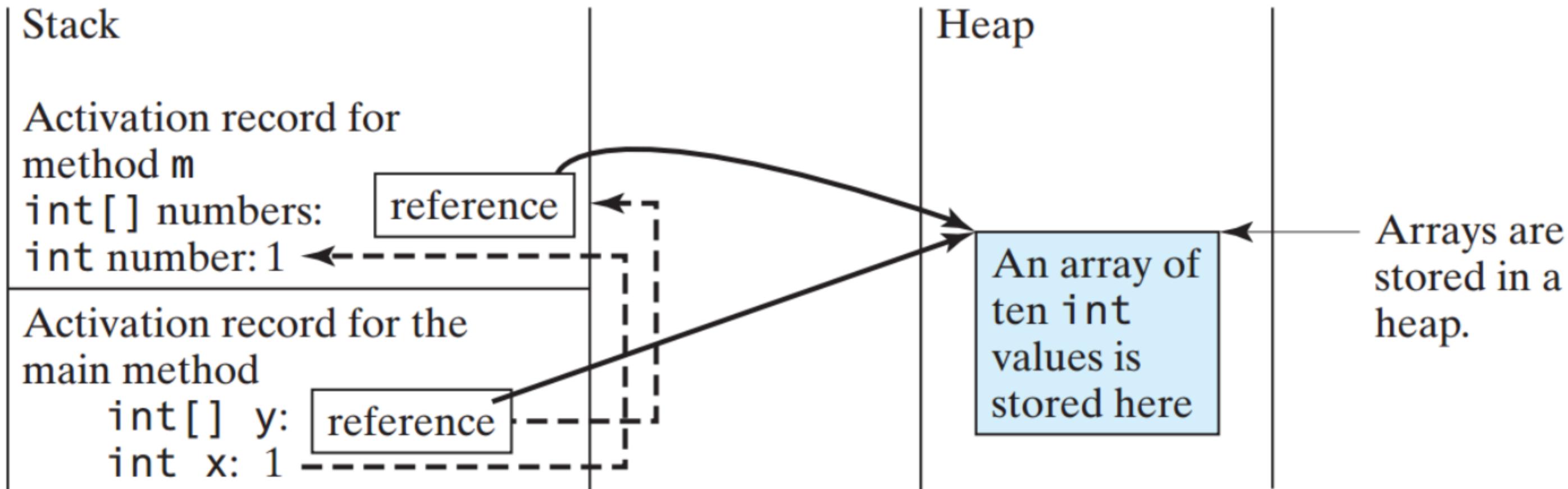
Passing arrays to methods

```
public static void main(String[] args) {  
    int x = 1; // x represents an int value  
    int[] y = new int[10]; // y represents an array of int values  
    m(x, y); // Invoke m with arguments x and y  
    System.out.println("x is " + x);  
    System.out.println("y[0] is " + y[0]);  
}  
  
public static void m(int number, int[] numbers) {  
    number = 1000; // Assign a new value to number  
    numbers[0] = 5555; // Assign a new value to numbers[0]  
}
```

x is 1
y[0] is 5555



Passing arrays to methods



varargs

```
public static void main(String[] args) {
    printMax(34, 3, 3, 52, 5);
    printMax(new int[] { 1, 2, 3 });
}

public static void printMax(int... numbers) {
    if (numbers.length == 0) {
        System.out.println("No argument passed");
        return;
    }
    int max = numbers[0];
    for (int i = 1; i < numbers.length; i++)
        if (numbers[i] > max)
            max = numbers[i];
    System.out.println("The max value is " + max);
}
```



Daugiamatis masyvas

```
//declare  
int[][] multiArr;  
  
//instantiate  
int[][] arr=new int[3][3];//3 row and 3 column  
  
//initialize  
arr[0][0]=1;  
arr[0][1]=2;  
arr[0][2]=3;  
//...  
arr[2][0]=7;  
arr[2][1]=8;  
arr[2][2]=9;  
  
//declaring and initializing 2D array  
int arr2[][]={{1,2,3},{2,4,5},{4,4,5}};
```



Daugiamatis masyvas

[0][1][2][3][4]				
[0]	0	0	0	0
[1]	0	0	0	0
[2]	0	0	0	0
[3]	0	0	0	0
[4]	0	0	0	0

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

[0][1][2][3][4]				
[0]	0	0	0	0
[1]	0	0	0	0
[2]	0	7	0	0
[3]	0	0	0	0
[4]	0	0	0	0

```
matrix[2][1] = 7;
```

[0][1][2]		
[0]	1	2
[1]	4	5
[2]	7	8
[3]	10	11
	12	

```
int[][] array = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9},  
    {10, 11, 12}  
};
```

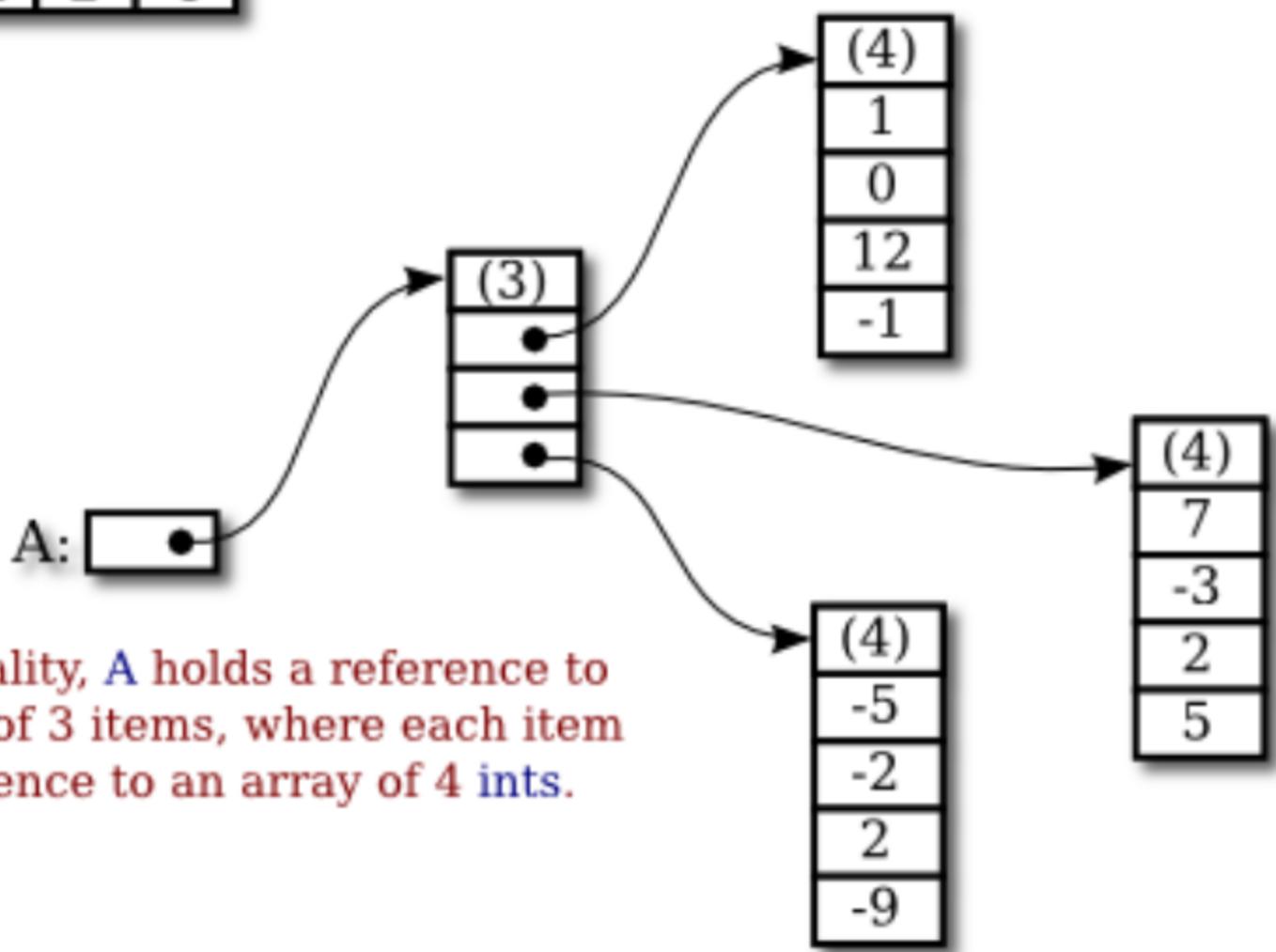


Daugiamatis masyvas

A:

1	0	12	-1
7	-3	2	5
-5	-2	2	-9

If you create an array `A = new int[3][4]`,
you should think of it as a "matrix" with
3 rows and 4 columns.

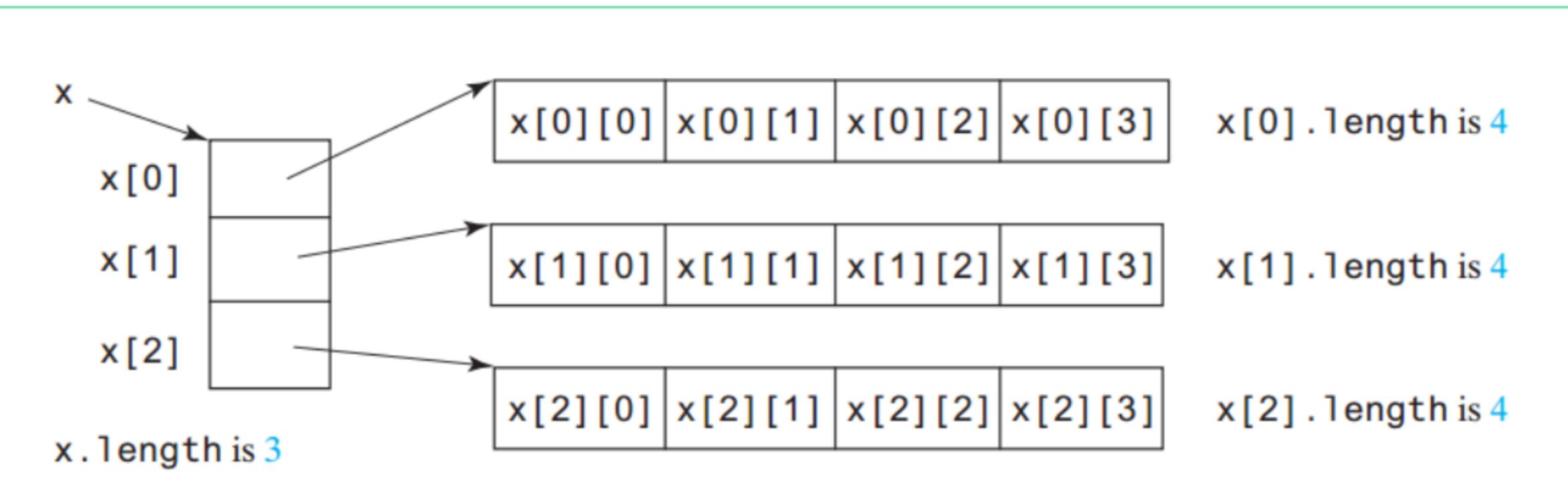


But in reality, `A` holds a reference to
an array of 3 items, where each item
is a reference to an array of 4 ints.



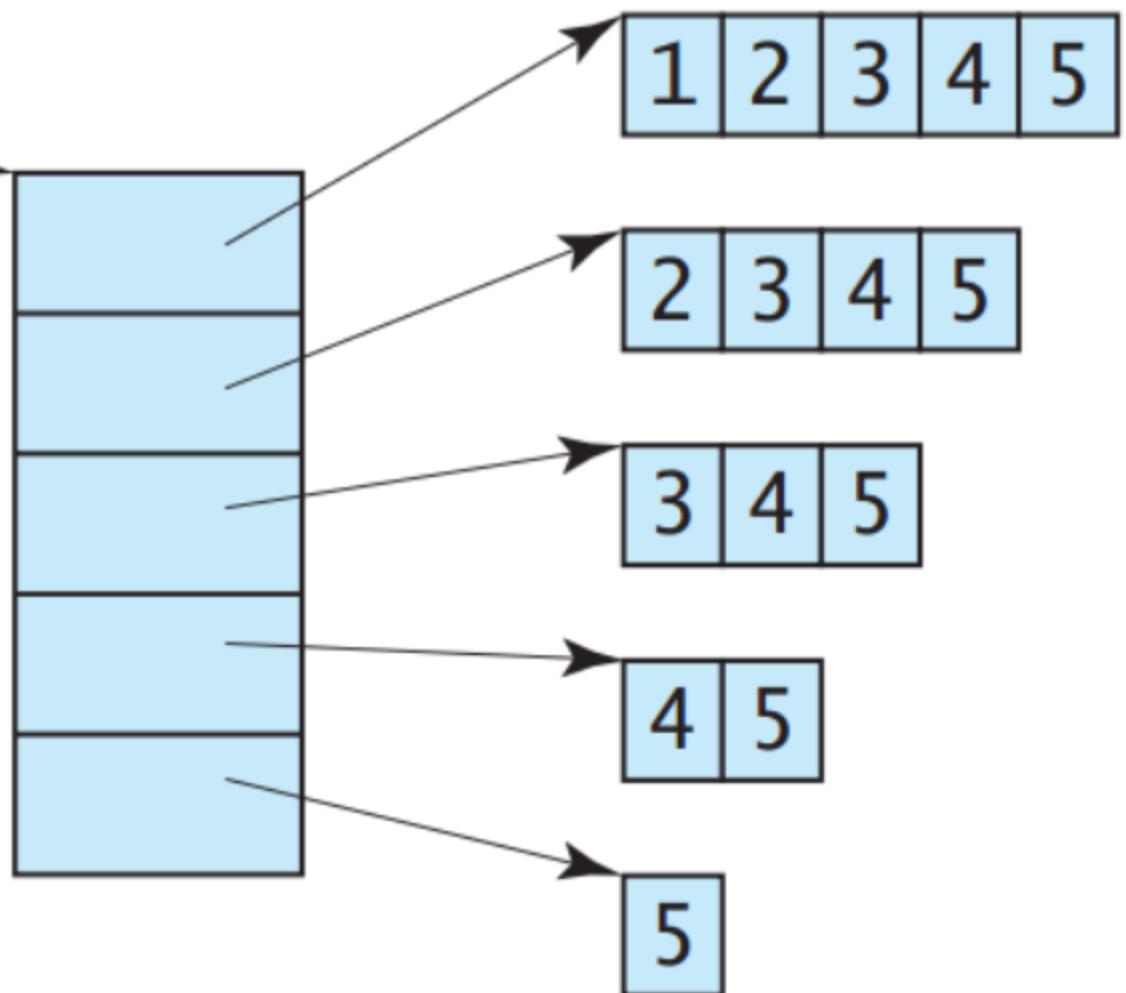
Daugiamatis masyvas

```
int[][][] x = new int[3][4];
```



Ragged array

```
int[][] triangleArray = {  
    {1, 2, 3, 4, 5},  
    {2, 3, 4, 5},  
    {3, 4, 5},  
    {4, 5},  
    {5}  
};
```



Dvimačio masyvo spausdinimas

```
int arr[][] = { { 1, 2, 3 }, { 2, 4, 5 }, { 4, 4, 5 } };  
  
//printing 2D array  
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        System.out.print(arr[i][j] + " ");  
    }  
    System.out.println();  
}
```

1	2	3
2	4	5
4	4	5



Dvimačio masyvo spausdinimas (2)

```
int arr[][] = { { 1, 2, 3 }, { 2, 4, 5 }, { 4, 4, 5 } };  
  
System.out.println(arr);  
//prints [I@15db9742  
  
System.out.println(Arrays.toString(arr));  
//prints [[I@6d06d69c, [I@7852e922, [I@4e25154f]  
  
System.out.println(Arrays.deepToString(arr));  
//prints [[1, 2, 3], [2, 4, 5], [4, 4, 5]]
```

