Array and Methods

Arash Habibi Lashkari
Ph.D. Candidate of UTM University
Kuala Lumpur, Malaysia

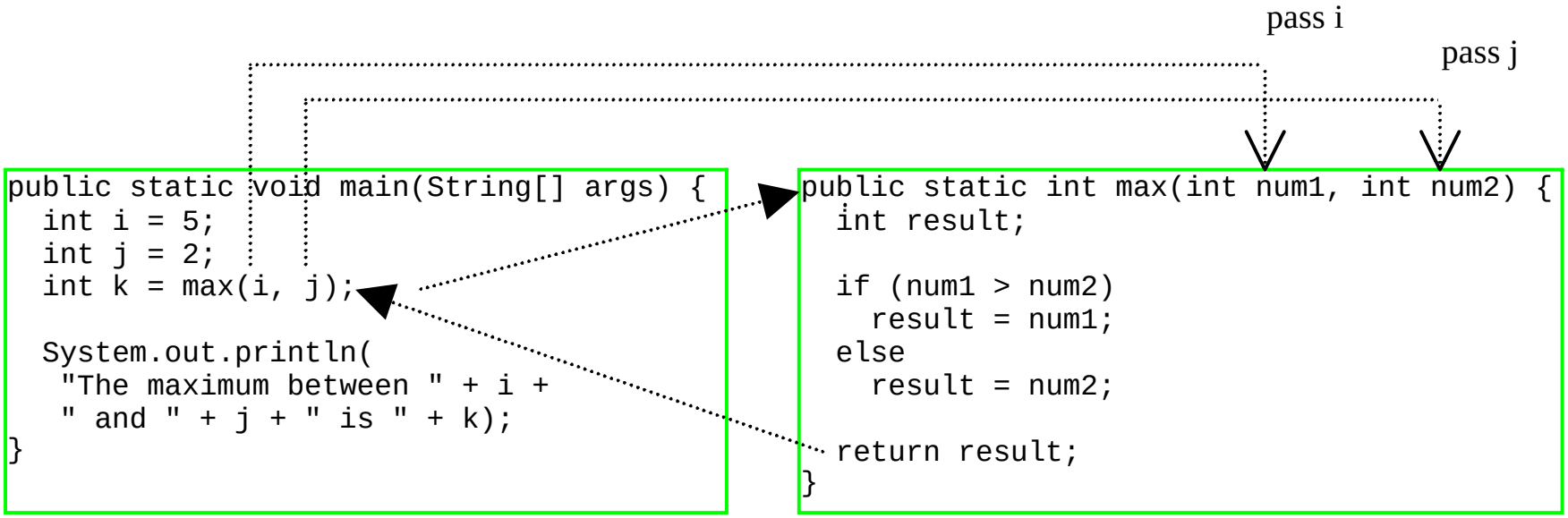
Introducing Methods

- *parameter profile* refers to the type, order, and number of the parameters of a method.
- *method signature* is the combination of the method name and the parameter profiles.
- The parameters defined in the method header are known as *formal parameters*.
- When a method is invoked, its formal parameters are replaced by variables or data, which are referred to as *actual parameters*.

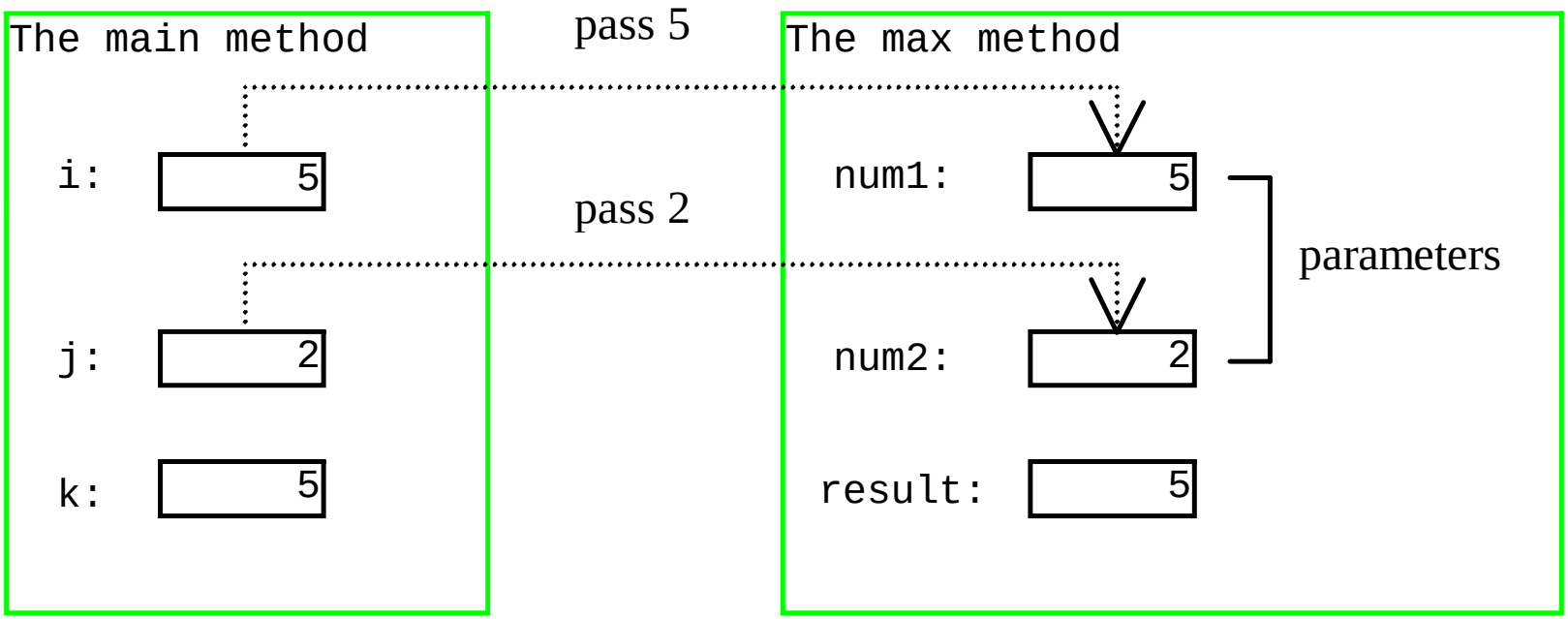
Declaring Methods

```
public static int max(int num1, int num2) {
  if (num1 > num2)
    return num1;
  else
    return num2;
}
```

Calling Methods



Calling Methods

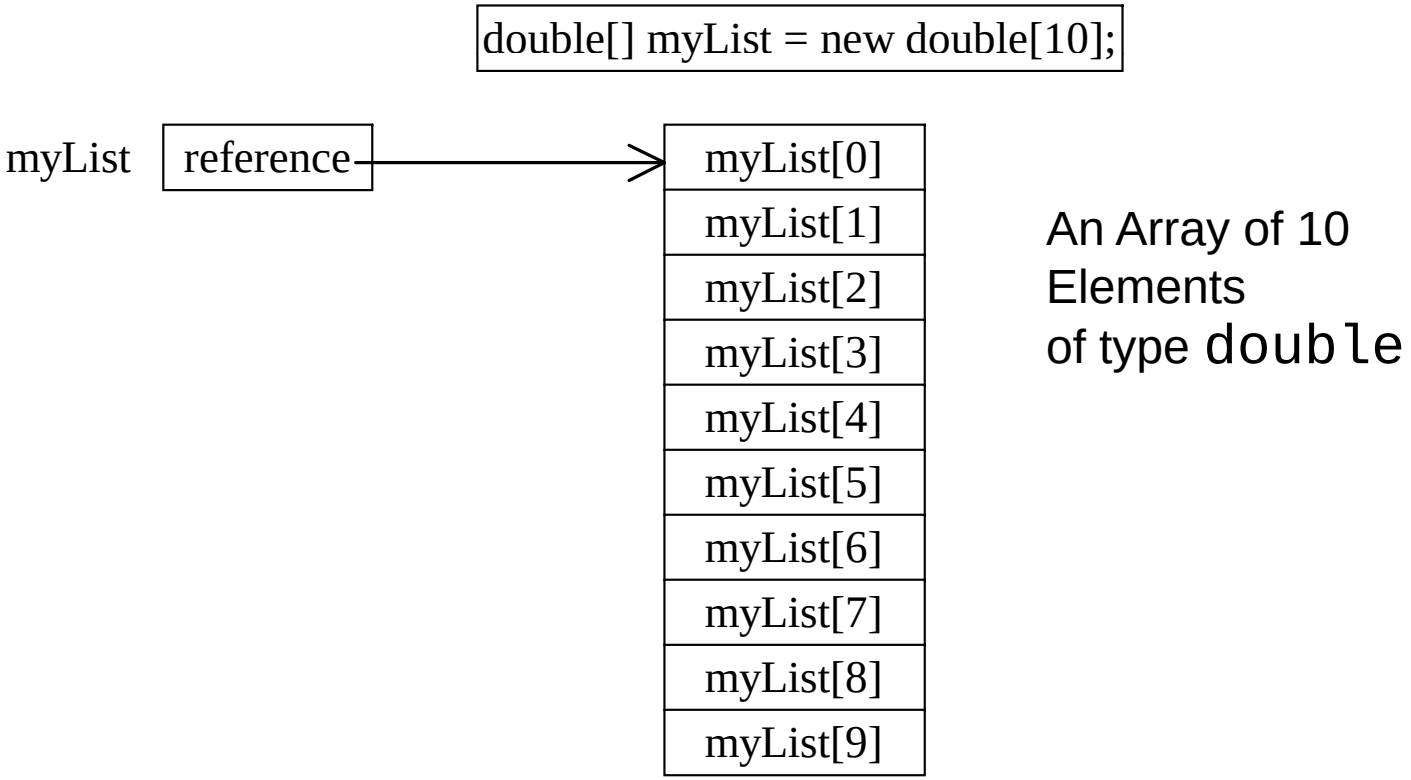


Passing Parameters

```
public static void nPrintln(String message, int n) {
  for (int i = 0; i < n; i++)
    System.out.println(message);
}
```

Arrays

Array is a data structure that represents a collection of the same types of data.



Arrays

Declaring Array Variables

- datatype[] arrayname;

  Example:

  double[] myList;

- datatype arrayname[];

  Example:

  double myList[];

Arrays

Creating Arrays:

arrayName = new datatype[arraySize];

Example:

myList = new double[10];

myList[0] references the first element in the array.

myList[9] references the last element in the array.

Arrays

- **Declaring and Creating in One Step**
- datatype[] arrayname = new datatype[arraySize];

  double[] myList = new double[10];

- datatype arrayname[] = new datatype[arraySize];

  double myList[] = new double[10];

Arrays

- Once an array is created, its size is fixed. It cannot be changed. You can find its size using

arrayVariable.length

For example,

myList.length returns 10

Arrays

Initializing Arrays

- Using a loop:

```
for (int i = 0; i < myList.length; i++)
  myList[i] = i;
```

- Declaring, creating, initializing in one step:

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

This shorthand syntax must be in one statement.

Arrays

Declaring, creating, initializing Using the Shorthand Notation

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

This shorthand notation is equivalent to the following statements:

```
double[] myList = new double[4];
myList[0] = 1.9;
myList[1] = 2.9;
myList[2] = 3.4;
myList[3] = 3.5;
```

Arrays

CAUTION

Using the shorthand notation, you have to declare, create, and initialize the array all in one statement. Splitting it would cause a syntax error. For example, the following is wrong:

```
double[] myList;

myList = {1.9, 2.9, 3.4, 3.5};
```

Arrays

Java treats arrays as objects. When the actual parameter is an array, it is passed to the method by reference:

```
int a[] = { 1, 2, 3, 4, 5};
...
printArray ( a );
...
public void printArray (int b[]) {
...
}
```

Arrays

- **Example:**
- Fill an array with 10 integer values. Pass the array to a method that will add up the values and return the sum. (In this example, no original data in the array was altered. The array information was simply used to find the sum.)

Arrays

```
public class FindSum
{
        public static void main (String [ ] args)
        {
              Scanner scanner = new Scanner(System.in);
              int [ ]  number = new int [ 10];                          // instantiate the array
              int i;
              int sum=0;
              for ( i = 0; i < 10; i++ ){                               // fill the array
                System.out.println("Enter number: " );
                  number[ i ] =scanner.nextInt();}
             int sum = find_sum(number);                                // invoke the method
             System.out.println("The sum is" +sum + ".");
        }
       public static int  find_sum(int [ ] value)                       //method definition to find sum
       {
            int i, total = 0;
            for(i=0; i<10; i++)
            {
               total = total + value[ i ];
            }
            return total;
        }
}
```

Questions

THANK YOU

Arash Habibi Lashkari

PHD. Candidate of UTM

Kuala Lumpur, Malaysia

Feb, 2010

THE END