Presentation 03
Arrays

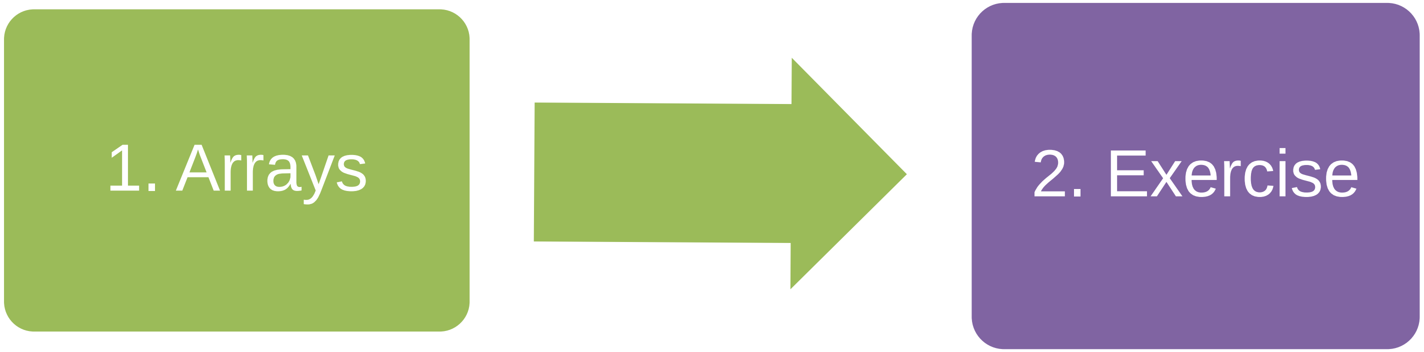
Christian Rodríguez Bustos

Edited by Juan Mendivelso

Object Oriented Programming

UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE BOGOTÁ

Agenda



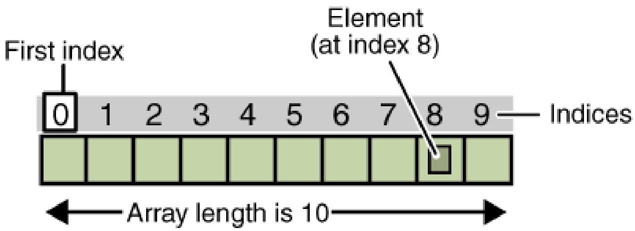
1. Arrays

2. Exercise

1. Arrays

Arrays are containers

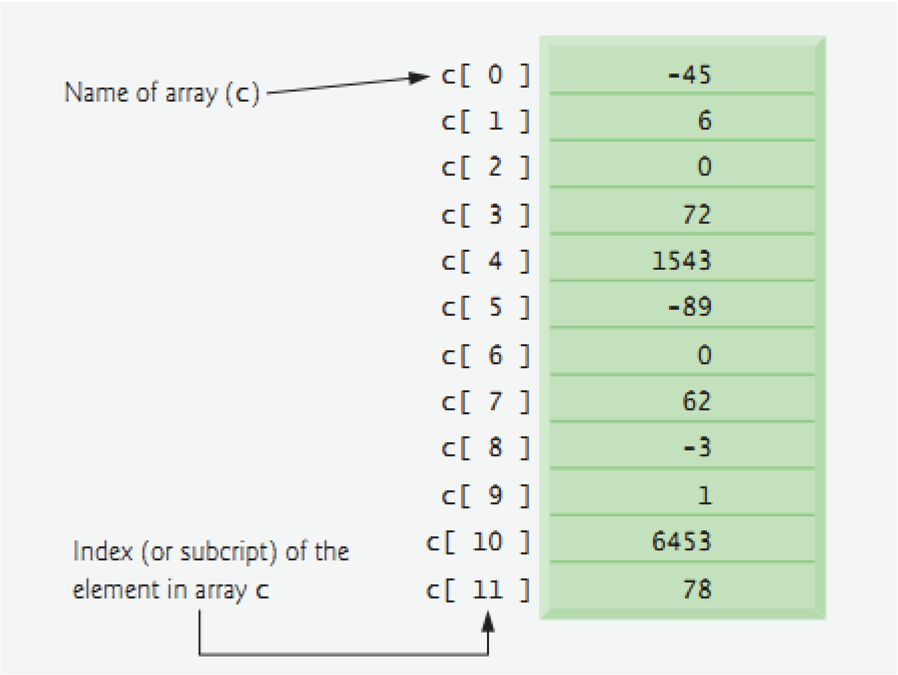
Container object that holds a fixed number of values of the **same type**



```
int c[] = new int[ 12 ];
```

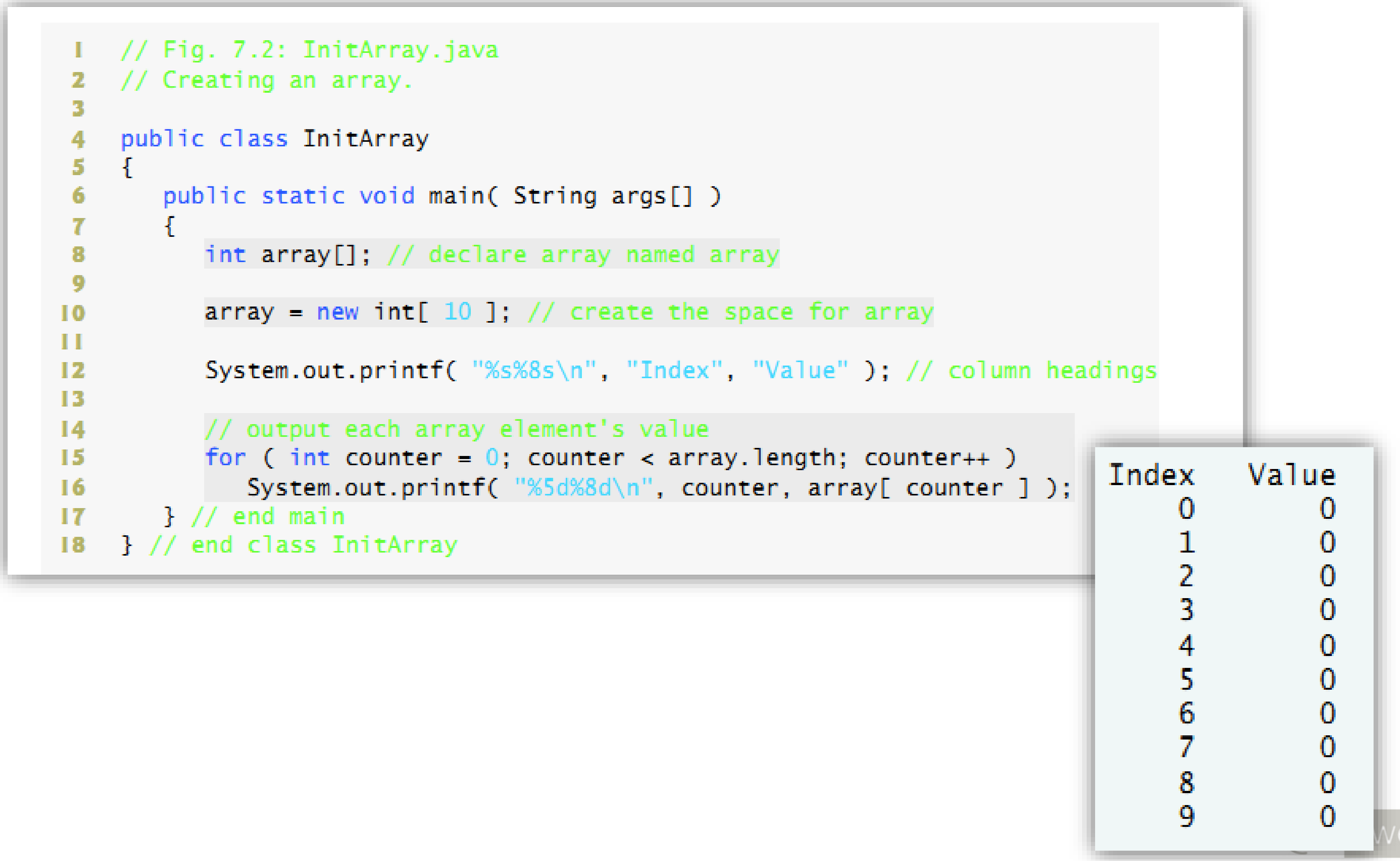
or

```
int c[];
c = new int[ 12 ];
```



c[0]	-45
c[1]	6
c[2]	0
c[3]	72
c[4]	1543
c[5]	-89
c[6]	0
c[7]	62
c[8]	-3
c[9]	1
c[10]	6453
c[11]	78

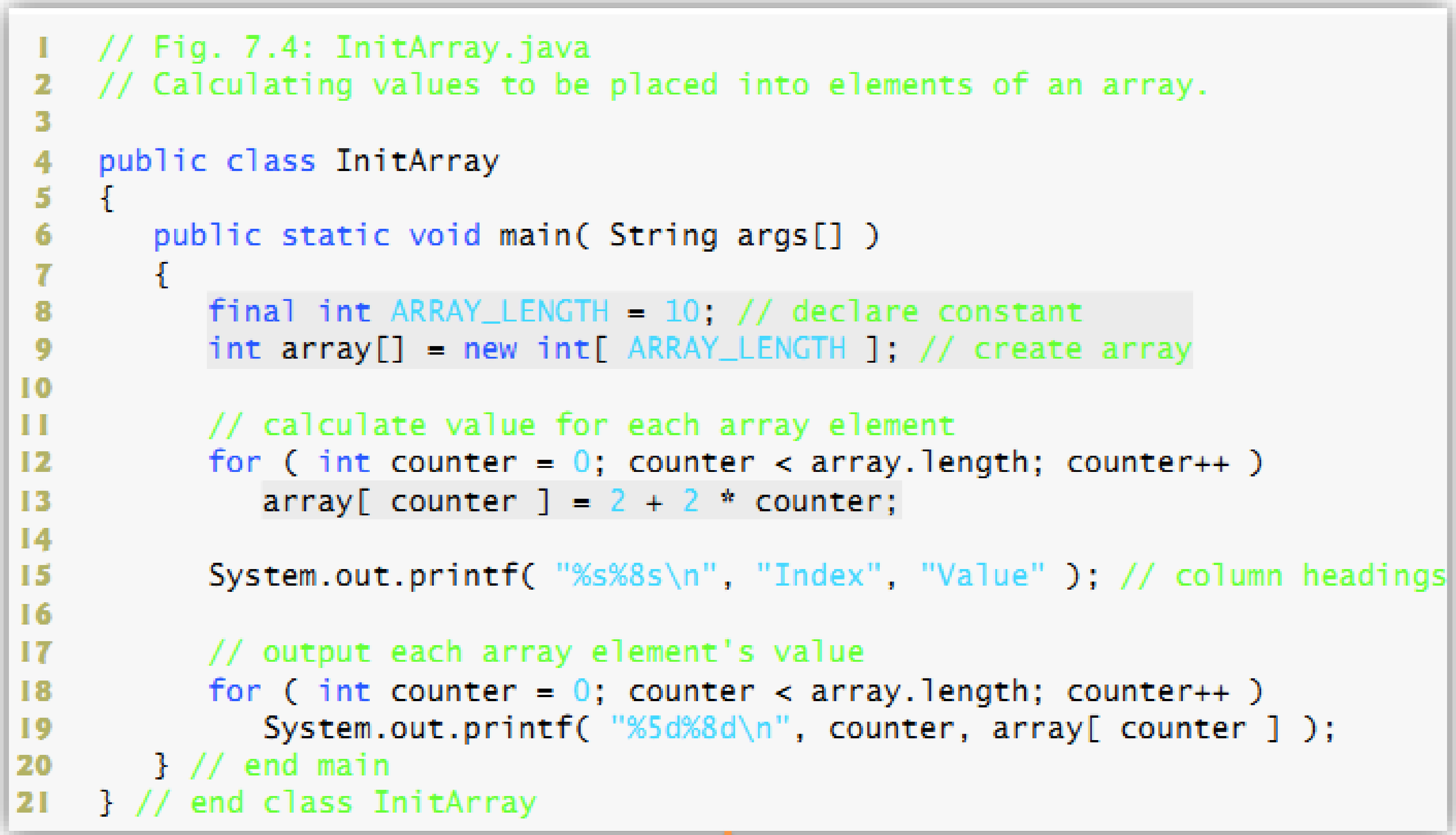
Arrays use example 1



```java
// Fig. 7.2: InitArray.java
// Creating an array.

public class InitArray
{
   public static void main( String args[] )
   {
      int array[]; // declare array named array

      array = new int[ 10 ]; // create the space for array

      System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings

      // output each array element's value
      for ( int counter = 0; counter < array.length; counter++ )
         System.out.printf( "%5d%8d\n", counter, array[ counter ] );
   } // end main
} // end class InitArray
```

```
Index   Value
    0       0
    1       0
    2       0
    3       0
    4       0
    5       0
    6       0
    7       0
    8       0
    9       0
```

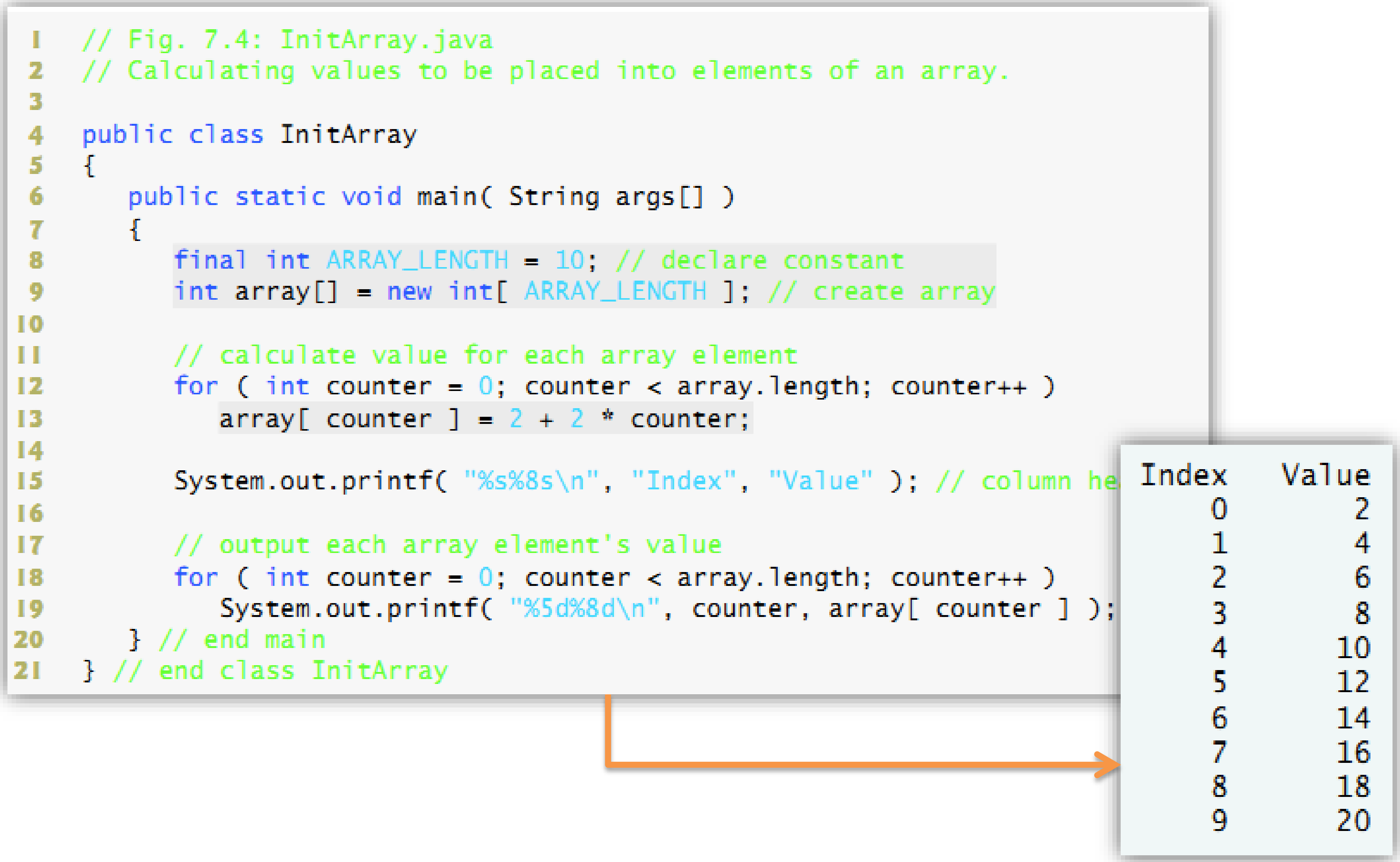
Arrays use example 2



```java
// Fig. 7.4: InitArray.java
// Calculating values to be placed into elements of an array.

public class InitArray
{
   public static void main( String args[] )
   {
      final int ARRAY_LENGTH = 10; // declare constant
      int array[] = new int[ ARRAY_LENGTH ]; // create array

      // calculate value for each array element
      for ( int counter = 0; counter < array.length; counter++ )
         array[ counter ] = 2 + 2 * counter;

      System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings

      // output each array element's value
      for ( int counter = 0; counter < array.length; counter++ )
         System.out.printf( "%5d%8d\n", counter, array[ counter ] );
   } // end main
} // end class InitArray
```

? ? ? ?
? ?

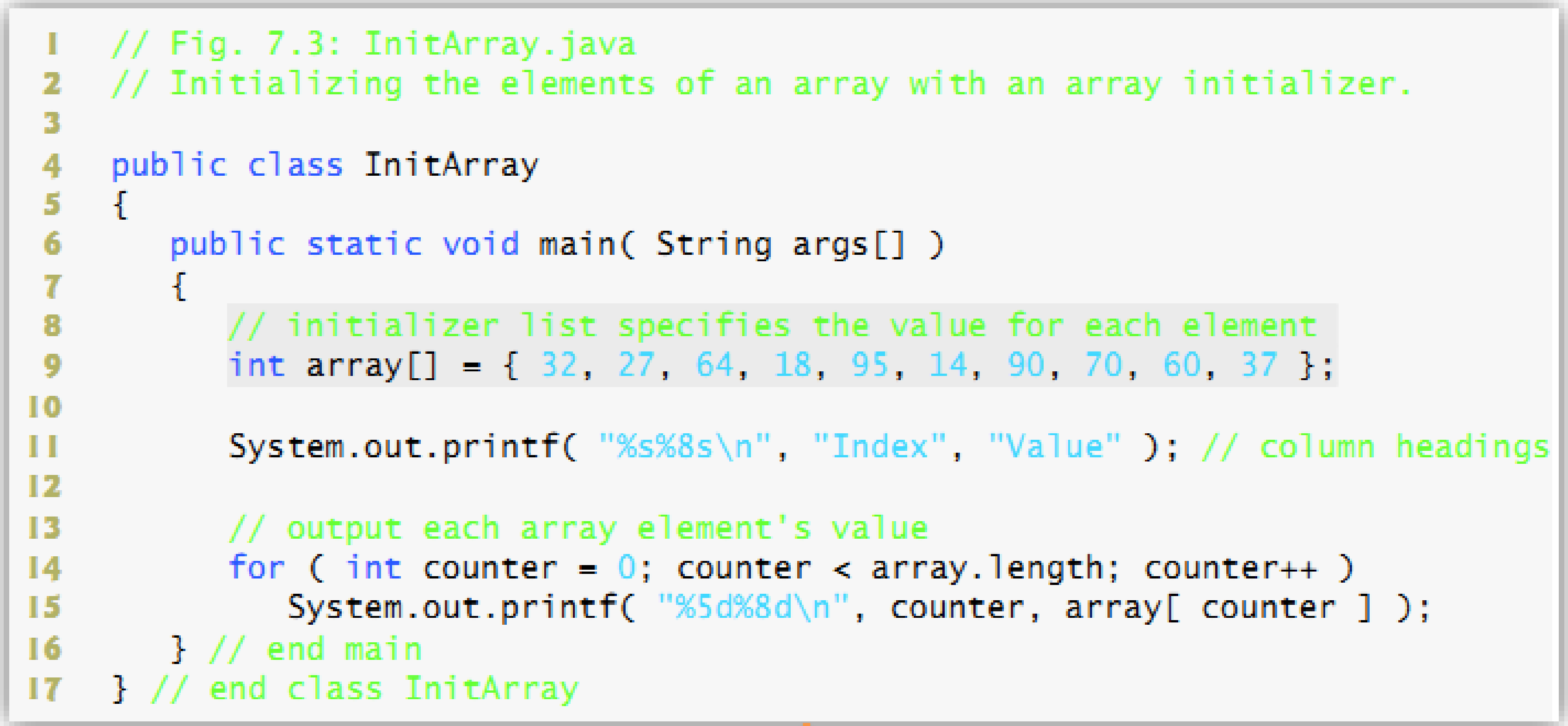
Arrays use example 2



```java
// Fig. 7.4: InitArray.java
// Calculating values to be placed into elements of an array.

public class InitArray
{
   public static void main( String args[] )
   {
      final int ARRAY_LENGTH = 10; // declare constant
      int array[] = new int[ ARRAY_LENGTH ]; // create array

      // calculate value for each array element
      for ( int counter = 0; counter < array.length; counter++ )
         array[ counter ] = 2 + 2 * counter;

      System.out.printf( "%s%8s\n", "Index", "Value" ); // column he

      // output each array element's value
      for ( int counter = 0; counter < array.length; counter++ )
         System.out.printf( "%5d%8d\n", counter, array[ counter ] );
   } // end main
} // end class InitArray
```

```
Index   Value
    0       2
    1       4
    2       6
    3       8
    4      10
    5      12
    6      14
    7      16
    8      18
    9      20
```

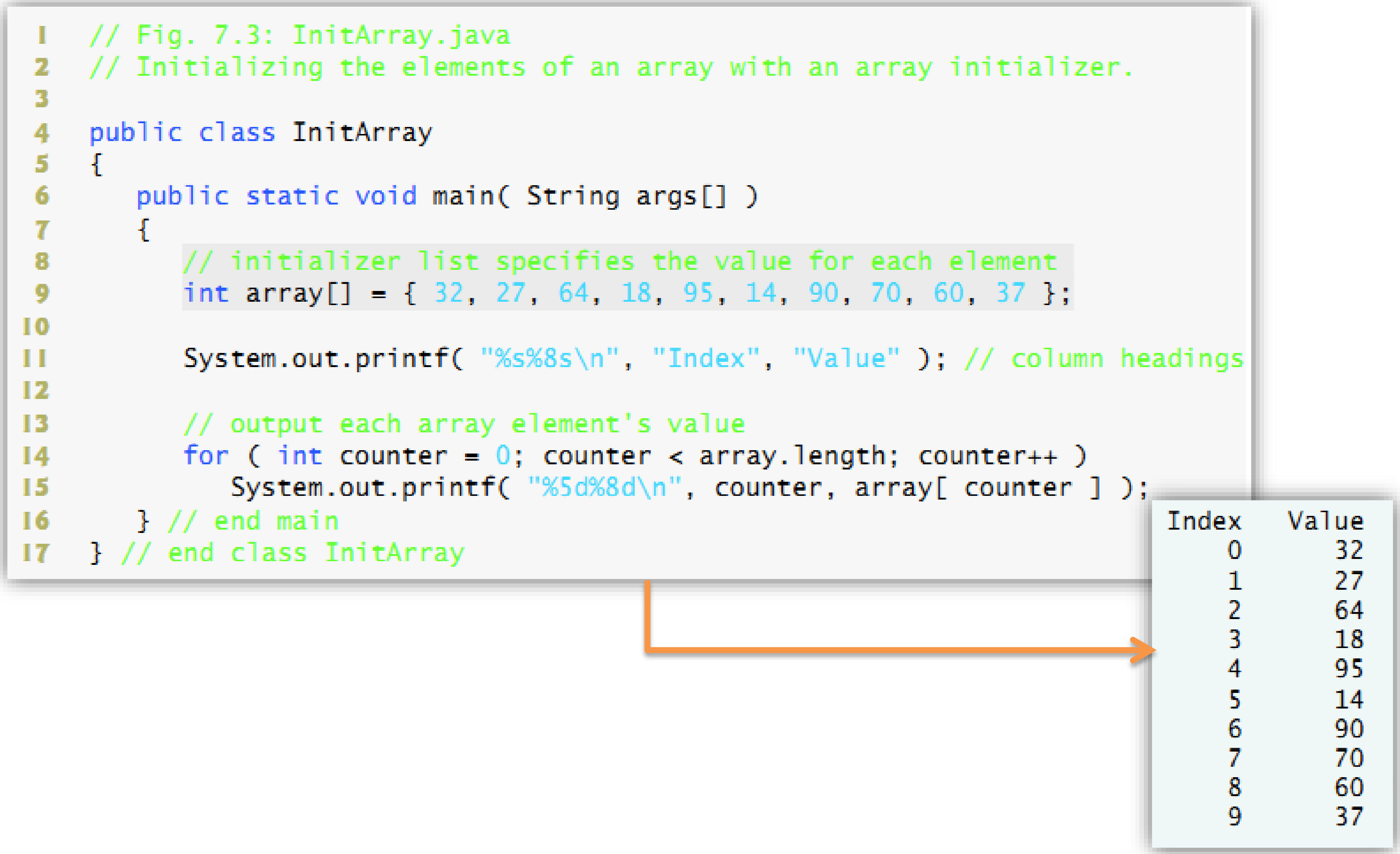
Arrays use example 3



```
// Fig. 7.3: InitArray.java
// Initializing the elements of an array with an array initializer.

public class InitArray
{
   public static void main( String args[] )
   {
      // initializer list specifies the value for each element
      int array[] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };

      System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings

      // output each array element's value
      for ( int counter = 0; counter < array.length; counter++ )
         System.out.printf( "%5d%8d\n", counter, array[ counter ] );
   } // end main
} // end class InitArray
```

? ? ? ?
? ?

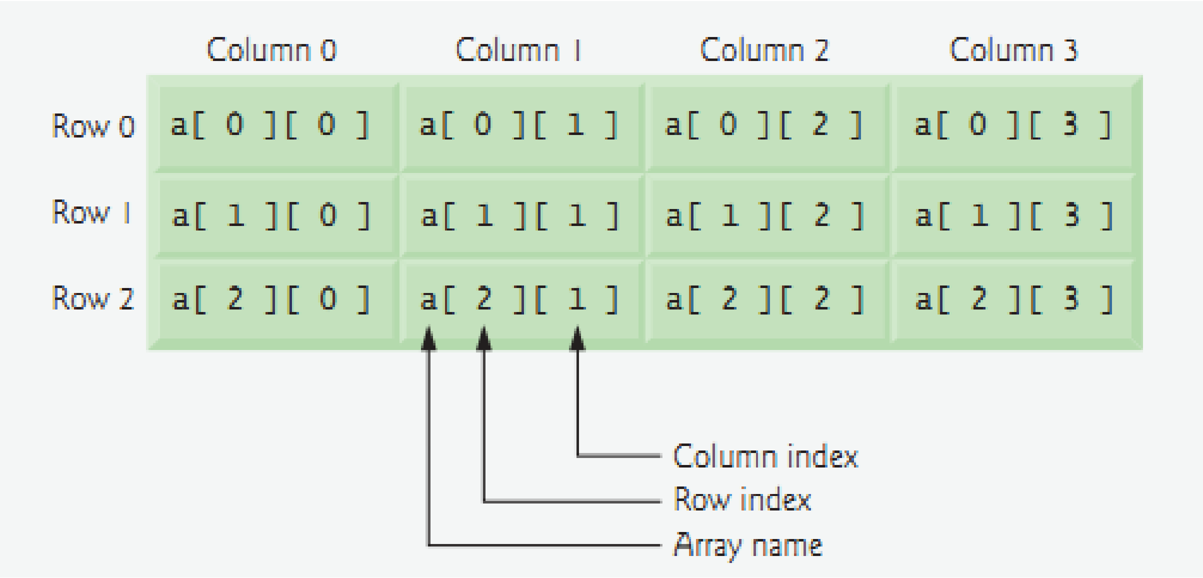
Arrays use example 3



```java
// Fig. 7.3: InitArray.java
// Initializing the elements of an array with an array initializer.

public class InitArray
{
   public static void main( String args[] )
   {
      // initializer list specifies the value for each element
      int array[] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };

      System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings

      // output each array element's value
      for ( int counter = 0; counter < array.length; counter++ )
         System.out.printf( "%5d%8d\n", counter, array[ counter ] );
   } // end main
} // end class InitArray
```

```
Index   Value
    0      32
    1      27
    2      64
    3      18
    4      95
    5      14
    6      90
    7      70
    8      60
    9      37
```

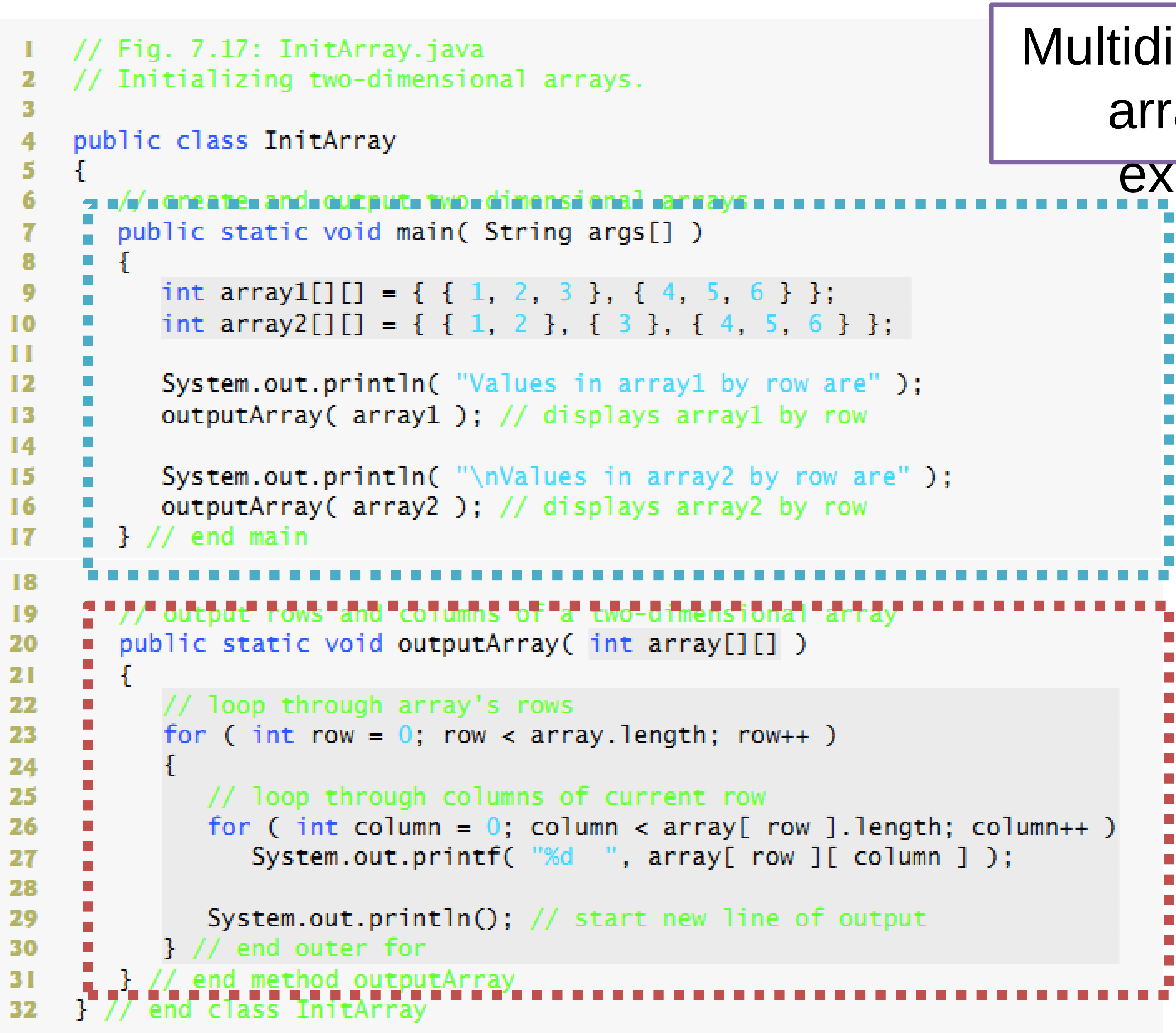
Working with multidimensional arrays



	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

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

Multidimensional array use example



```
// Fig. 7.17: InitArray.java
// Initializing two-dimensional arrays.

public class InitArray
{
   // create and output two-dimensional arrays
   public static void main( String args[] )
   {
      int array1[][] = { { 1, 2, 3 }, { 4, 5, 6 } };
      int array2[][] = { { 1, 2 }, { 3 }, { 4, 5, 6 } };

      System.out.println( "Values in array1 by row are" );
      outputArray( array1 ); // displays array1 by row

      System.out.println( "\nValues in array2 by row are" );
      outputArray( array2 ); // displays array2 by row
   } // end main

   // output rows and columns of a two-dimensional array
   public static void outputArray( int array[][] )
   {
      // loop through array's rows
      for ( int row = 0; row < array.length; row++ )
      {
         // loop through columns of current row
         for ( int column = 0; column < array[ row ].length; column++ )
            System.out.printf( "%d  ", array[ row ][ column ] );

         System.out.println(); // start new line of output
      } // end outer for
   } // end method outputArray
} // end class InitArray
```

Method 1 (function)

Method 2 (function)

Example

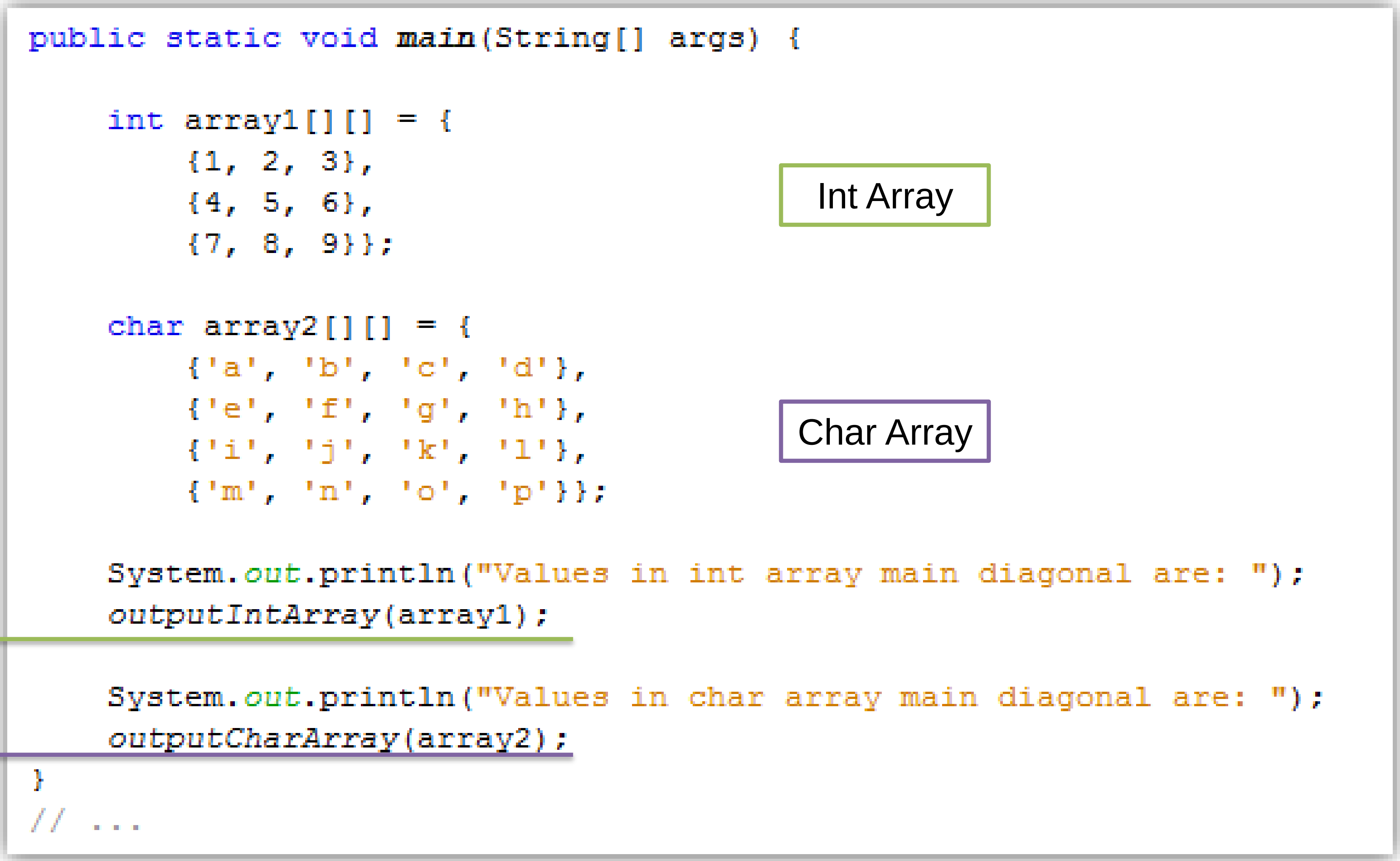
- Print the main diagonal of the next two multidimensional arrays:

1	2	3
4	5	6
7	8	9

Numbers array

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

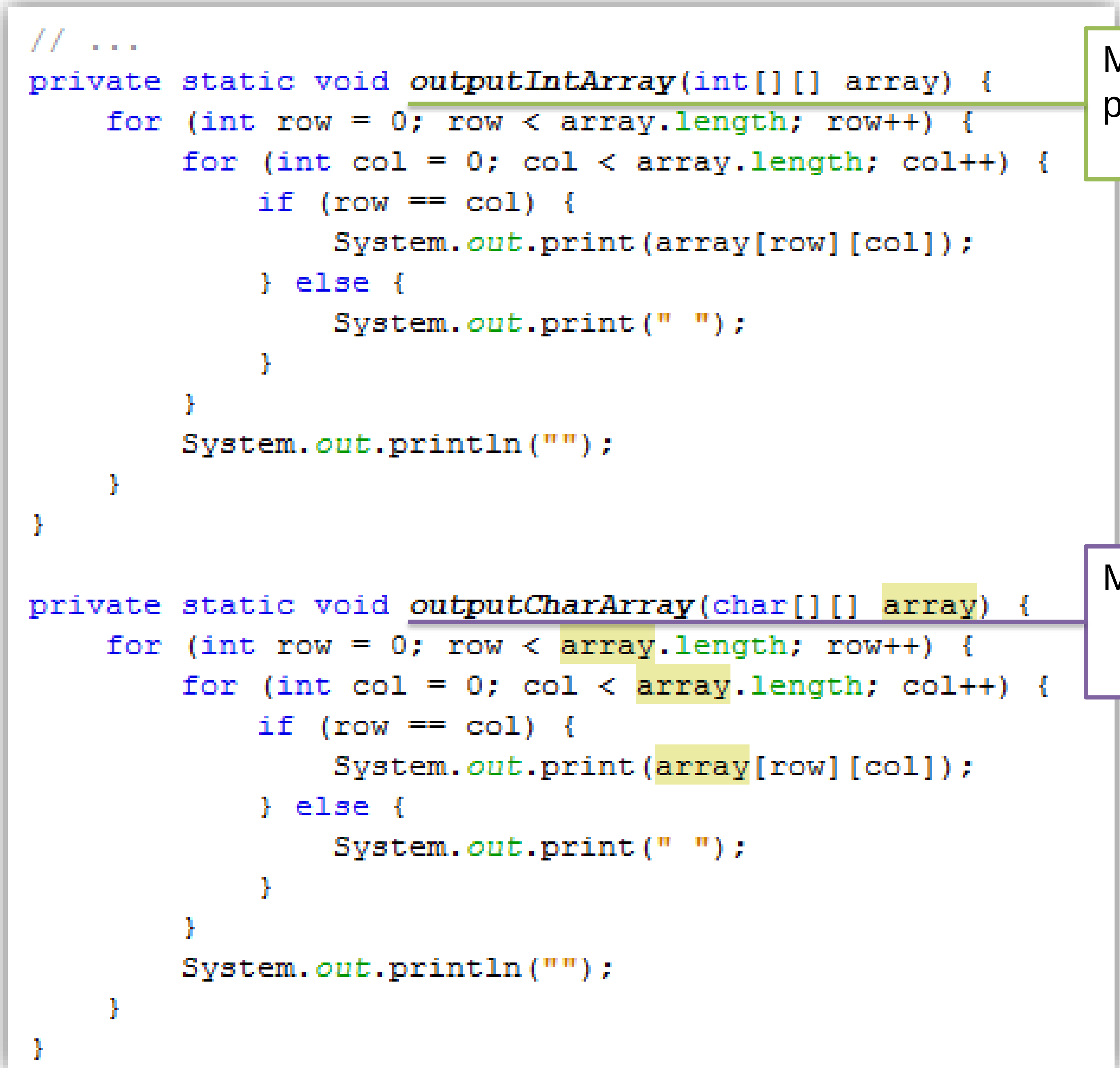
Letters array



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

    int array1[][] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}};

    char array2[][] = {
        {'a', 'b', 'c', 'd'},
        {'e', 'f', 'g', 'h'},
        {'i', 'j', 'k', 'l'},
        {'m', 'n', 'o', 'p'}};

    System.out.println("Values in int array main diagonal are: ");
    outputIntArray(array1);

    System.out.println("Values in char array main diagonal are: ");
    outputCharArray(array2);
}
// ...
```

Int Array

Char Array

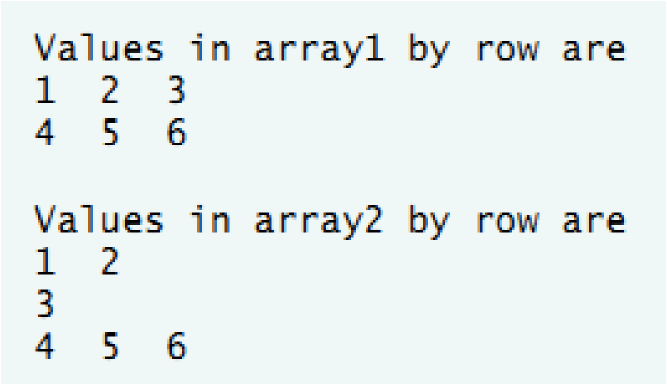


```java
// ...
private static void outputIntArray(int[][] array) {
    for (int row = 0; row < array.length; row++) {
        for (int col = 0; col < array.length; col++) {
            if (row == col) {
                System.out.print(array[row][col]);
            } else {
                System.out.print(" ");
            }
        }
        System.out.println("");
    }
}

private static void outputCharArray(char[][] array) {
    for (int row = 0; row < array.length; row++) {
        for (int col = 0; col < array.length; col++) {
            if (row == col) {
                System.out.print(array[row][col]);
            } else {
                System.out.print(" ");
            }
        }
        System.out.println("");
    }
}
```

Method for printing **int** arrays

Method for printing **char** arrays

Multidimensional array use example



```
Values in array1 by row are
1  2  3
4  5  6

Values in array2 by row are
1  2
3
4  5  6
```

2. Exercise

Exercise

1. Find the minimum value of an array.
2. Find the sum of the elements that are at an even position in an array.
3. Find the minimum value in a matrix.
4. Calculate the sum of the main diagonal and the sum of the secondary diagonal of a square matrix.
5. Print the prime numbers of odd rows in a matrix.
6. Calculate an array of ints equivalent to an input array but removing the repeated elements. For example, if the input array is (2,-3,2,8,8,2), the output array is (2,-3,8).

References

- [Barker] J. Barker, *Beginning Java Objects: From Concepts To Code*, Second Edition, Apress, 2005.

- [Deitel] H.M. Deitel and P.J. Deitel, *Java How to Program: Early Objects Version*, Prentice Hall, 2009.

- [Sierra] K. Sierra and B. Bates, *Head First Java*, 2nd Edition, O'Reilly Media, 2005.