ASTANA IT UNIVERSITY	**Cours:** Object-Oriented Programming **Deadline:** December 25, 2023 till (23:59)

Instructor: Toleu Altynbek

ASSIGNMENT 2

Lesson plan:

Arrays
Two-dimensional arrays
Jagged arrays

An array is a container of elements

You've probably heard that computers can process enormous amounts of information. The conditional statement (if-else) and loops (for, while) certainly help with this. But you won't get far with them, because the processed data needs to be stored somehow.

For this case, in Java, as in almost all programming languages, there is such a great thing as arrays (Array). They are also called tables.

An array is a special object in which you can store more than one value.

Creating an array of elements in Java

Let's say your program needs to store 100 integers somewhere. Most likely, an array would be suitable for this. And how to create it?

If we wanted to store one integer, we'd use an int, but if we wanted to store 100 integers, we'd probably need an array of int. Here's what the code to create it will look like:

```
int[] array = new int[100];
```

Let's figure out what is written here.

As you may have guessed, to the left of the equal sign we have a variable declaration named array and type int[]. The int type is followed by square brackets, which seem to hint that in "boxes" of this type it will be possible to store not one value, but several.

Task 1

In the main method, initialize the intArray and doubleArray variables with arrays of the corresponding types, the size of which is 10.

Requirements:
• In the main(String[]) method, initialize the intArray variable according to the condition.
• In the main(String[]) method, initialize the doubleArray variable according to the condition.

Code	Explanation

`int[] a = new int[10];`
`a[2] = 4;`
`a[7] = 9;`
`a[9] = a[2] + a[5];`
	We create an array of 10 elements of type int.
	In the cell with index 2 we write the value 4.
	In the cell with index 7 we write the value 9.
	In the cell with index 9 we write the sum of the values that are stored in cells 2 (4 is stored) and 5 (0 is stored).

Task 2

In the main(String[]) method you need to fill the strings array with values. If the array index is even, assign the value "Even" (zero means the digit is even), otherwise assign it "Odd". Print the result to the console.

Requirements:
• The Solution class must have a public static variable strings of type String[], initialized with an array of 5 elements.
• The main(String[]) method should fill the array according to the condition.

Task 3

Implement the main(String[]) method, which reverses the sign of an array element if the value of that element is even.

Requirements:

• Implement the main(String[]) method according to the condition.

Array length

As you saw in the previous example, you can separately create an array variable and then assign a value to it somewhere in the code (a reference to an array object).

And how to continue working with such an array? How to find out how many elements it has?

For this purpose, the array has a special property (variable) - length. And you can find out the length of the array using the following expression:

```
array.length;
```

Code	Explanation

`int[] array;`
`if (a < 10)`
`  array = new int[10];`
`else`
`  array = new int[20];`
`for (int i = 0; i < array.length; i++)`
`{`
 	
`System.out.println(array[i]);`
`}`
	Create an array variable of type int[]
	If variable a is less than 10,
	then create an array of 10 elements.
	Otherwise
	create an array of 20 elements
	Loop through all array elements: from 0 to array.length - 1

Task 4

Implement the main(String[]) method, in which you need to copy the contents of the firstArray and secondArray arrays into one resultArray array.
The firstArray must be at the beginning of the new resultArray, and the secondArray must be after it.
The program must correctly handle arrays of any length.

Requirements:
• The Solution class must have a public static variable firstArray of type int[], initialized as an array with 10 values.
• The Solution class must have a public static variable secondArray of type int[], initialized as an array with 10 values.
• The Solution class must have a public static variable resultArray of type int[].

• Implement the main(String[]) method according to the condition.

Task 5

For this task you need: Enter the number N from the keyboard. Count N integers and fill the array with them. Find the maximum number among the array elements. Requirements: • In the main(String[]) method, read the number N from the keyboard, then initialize the array array of N elements in size, and fill it with numbers from the keyboard. • In the main(String[]) method, print to the console the maximum number among the array elements.

Two-dimensional arrays

And one more interesting fact about arrays. Arrays are not only linear, but also two-dimensional.

And what does this mean, you ask?

This means that array cells can be represented not only as a column (or row), but also as a rectangular table.

```
int[][] имя = new int[ширина][высота];
```

Where name is the name of the array variable, width is the width of the table (in cells), and height is the height of the table. Example:

`int[][] data = new int[2][5];` `data[1][1] = 5;`	We create a two-dimensional array: two columns and 5 rows. We write 5 in cell (1,1).

By the way, for two-dimensional arrays you can also use fast initialization:

```
// lengths of months of the year quarterly
int[][] months = { {31, 28, 31}, {30, 31, 30}, {31, 31, 30}, {31, 30, 31} };
```

Task 6

Initialize the MULTIPLICATION_TABLE array with the value new int[10][10], fill it with the multiplication table and output it to the console in the following form:
1 2 3 4 …
2 4 6 8 …
3 6 9 12 …
4 8 12 16 …
...
The numbers on the line are separated by a space.

Requirements:
• In the main method, the MULTIPLICATION_TABLE array must be filled with the multiplication table.
• The output text must be 10 lines long.
• Each line output must contain 10 numbers separated by a space.
• The numbers printed must be multiplication tables.

Jagged Arrays in Java

In Java, you, as a programmer, can not only swap the rows of a two-dimensional array, but also construct this array yourself the way you want.

Let's say you want the first row of a two-dimensional array to have a length of 10, and the second to have a length of 50. Is it possible to do this? Yes, you can.

First you need to create a “container of containers” - the first array that will store references to string arrays. This is done like this:

```
int[][] имя = new int[height][];
```

`int[][] matrix = new int[2][];`
`matrix[0] = new int[10];`
`matrix[1] = new int[50];`
	Two-dimensional array
	
	Row zero is an array of 10 elements
	The first line is an array of 50 elements.

By the way, how can we find out the length of the “container of containers” in our example? This is also an array object, which means it has a length.

The correct answer is **matrix.length**.
What about string arrays? **matrix[0].length**

Task 7

Create a triangular array where the value of each element is the sum of its indices. For example:
array[7][3] = 7 + 3 = 10,
array[3][0] = 3 + 0 = 3.

Display the array as follows:
0
12
2 3 4
3 4 5 6
4 5 6 7 8
5 6 7 8 9 10
...
The numbers on the line are separated by a space.

You can define a triangular array as follows:
int[][] array = new int[10][];
array[0] = new int[1];
array[1] = new int[2];
array[2] = new int[3];
...

Requirements:
• In the main(String[]) method, fill the two-dimensional array result with numbers according to the condition.
• The output text must be 10 lines long.
• The numbers displayed must match the condition.

Working with a two-dimensional array

Let's say you want to display a two-dimensional array. How to do this?

Our code will look something like this:

`int[][] matrix = new int[3][];`
`matrix[0] = new int[]{1, 2, 3, 4, 5, 6};`
`matrix[1] = new int[]{1, 2, 3};`
	Creating an Array
	Filling an Array with Values
	
	Outer loop through array rows - iterates through array rows.

`matrix[2] = new int[]{1};`
`for (int i = 0; i < matrix.length; i++) {`
`  for (int j = 0; j < matrix[i].length; j++)`
`  System.out.print( matrix[i][j] + " " );`
`  System.out.println();`
`}`
	Inner loop through cells - iterates through cells of one row.

Task 8

A two-dimensional array is an array of arrays, that is, in each of its cells there is a reference to a certain array. But it is much easier to present it in the form of a table that has a given number of rows (first dimension) and number of columns (second dimension). In this task we will create such an array.

To do this, implement the main(String[]) method, which:

Reads the number N from the console - the number of rows in the array (assume that this number will be greater than 0, you don't have to check it).
Reads N numbers from the console (any numbers greater than 0 may not be checked).
Initializes a two-dimensional multiArray:
number of lines N;
lines - arrays, the size of which corresponds to the numbers entered in step 2 (in the order of entry).
Example:

The number 5 has been entered.
The numbers entered are 1, 7, 5, 9, 3.
We get the following array:
[ ]
[ ][ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ][ ][ ][ ][ ]
[ ][ ][ ]

Requirements:
• The Solution class must have an uninitialized public static variable multiArray of type int[][].
• The main(String[]) method should read numbers from the keyboard according to the condition.
• In the main(String[]) method, create a two-dimensional array according to the condition.