

JAVA PROGRAMMING COURSE

EXERCISE

VARARGS IN JAVA

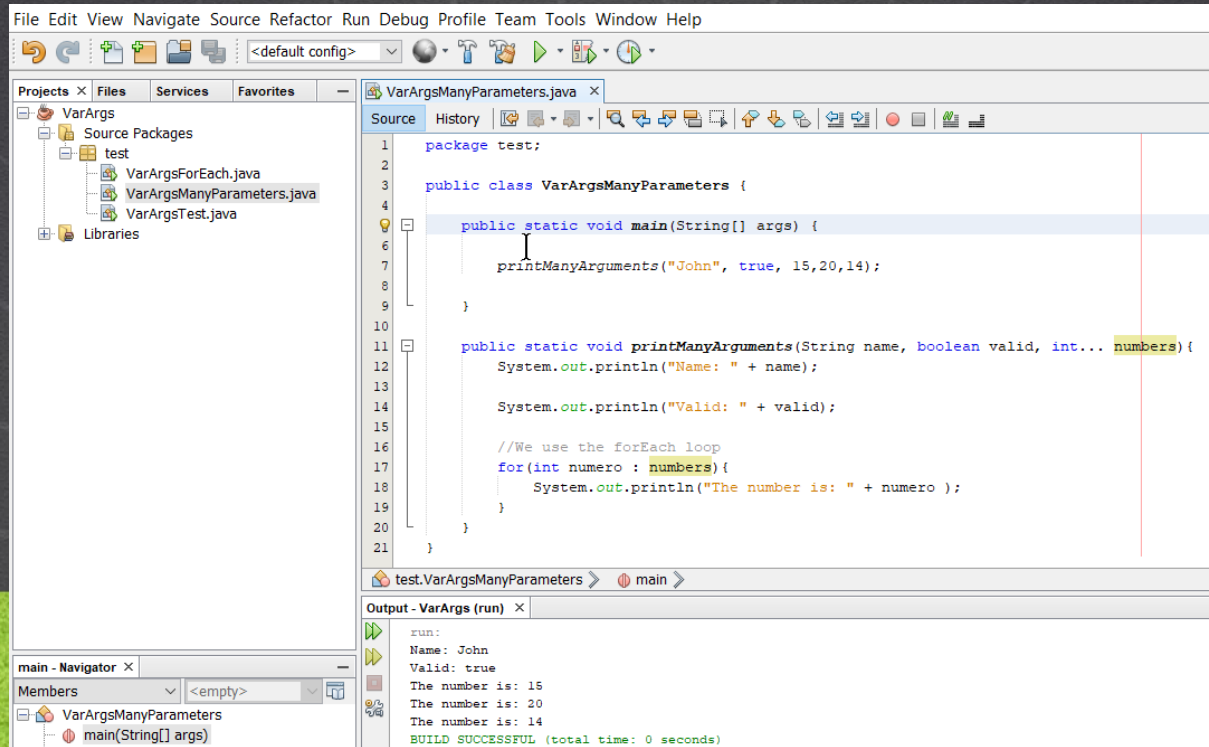


JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Create a program to implement the use of variable arguments in Java (varargs). At the end we should observe the following:



The screenshot displays an IDE window with the following components:

- Project Explorer:** Shows a project named 'VarArgs' with a 'test' package containing 'VarArgsForEach.java', 'VarArgsManyParameters.java', and 'VarArgsTest.java'.
- Source Editor:** Displays the code for 'VarArgsManyParameters.java'. The code defines a package 'test', a class 'VarArgsManyParameters', and a static method 'main' that calls 'printManyArguments' with four arguments: 'John', 'true', '15', and '20'. The 'printManyArguments' method is a static void method that takes a String 'name', a boolean 'valid', and a variable-length array of integers 'numbers'. It prints the name, valid status, and each number in the array using a for-each loop.
- Run Configuration:** Shows 'test.VarArgsManyParameters' with the 'main' method selected.
- Output Console:** Displays the output of the program run, showing the name 'John', valid status 'true', and the numbers '15', '20', and '14' (the last number is not explicitly shown in the output but is implied by the code and the 'BUILD SUCCESSFUL' message).

```
1 package test;
2
3 public class VarArgsManyParameters {
4
5     public static void main(String[] args) {
6         printManyArguments("John", true, 15, 20, 14);
7     }
8
9     public static void printManyArguments(String name, boolean valid, int... numbers) {
10         System.out.println("Name: " + name);
11         System.out.println("Valid: " + valid);
12         //We use the forEach loop
13         for(int numero : numbers){
14             System.out.println("The number is: " + numero );
15         }
16     }
17 }
```

run:
Name: John
Valid: true
The number is: 15
The number is: 20
The number is: 14
BUILD SUCCESSFUL (total time: 0 seconds)

1. CREATE A NEW PROJECT

Create a new project:

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

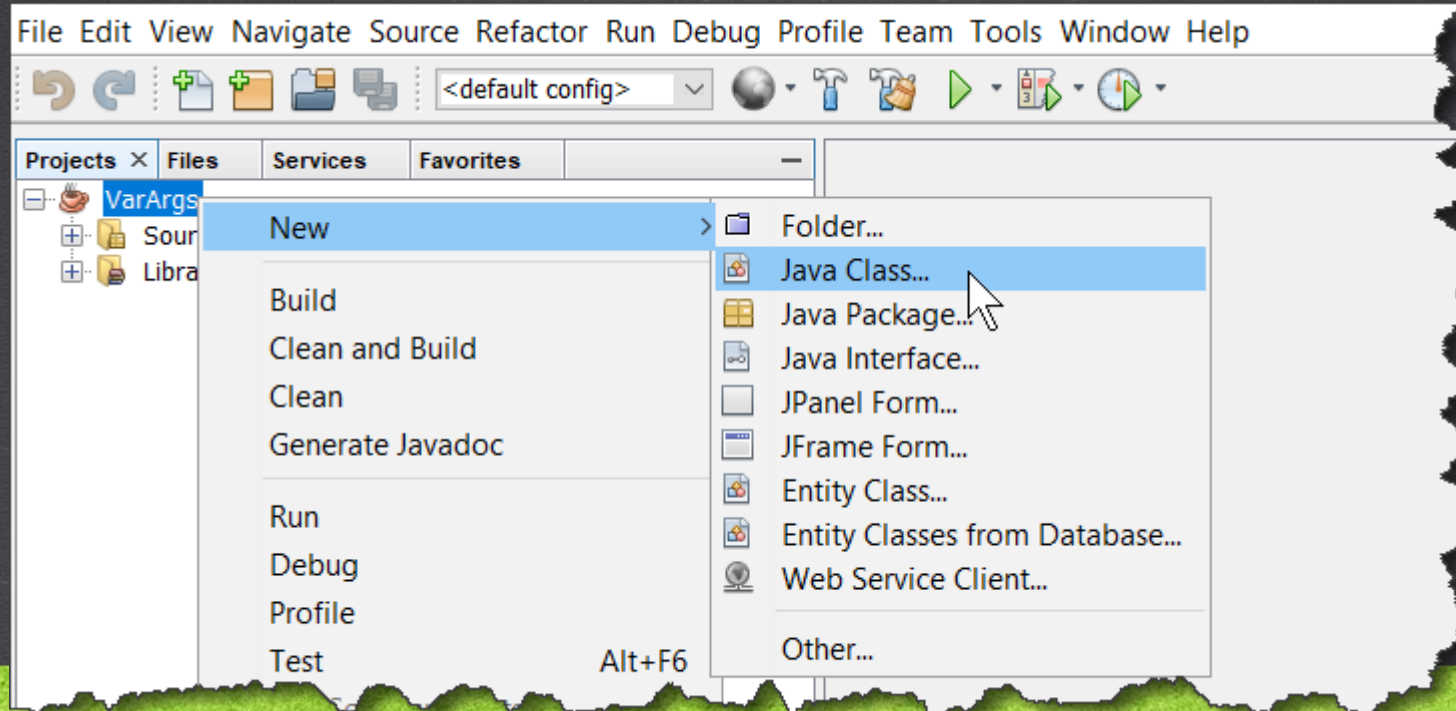
☐ Create Main Class

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

2. CREATE A NEW CLASS

Create a new Java class:



JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

2. CREATE A NEW CLASS

Create a new Java class:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

3. MODIFY THE CODE

VarArgsTest.java:

```
package test;

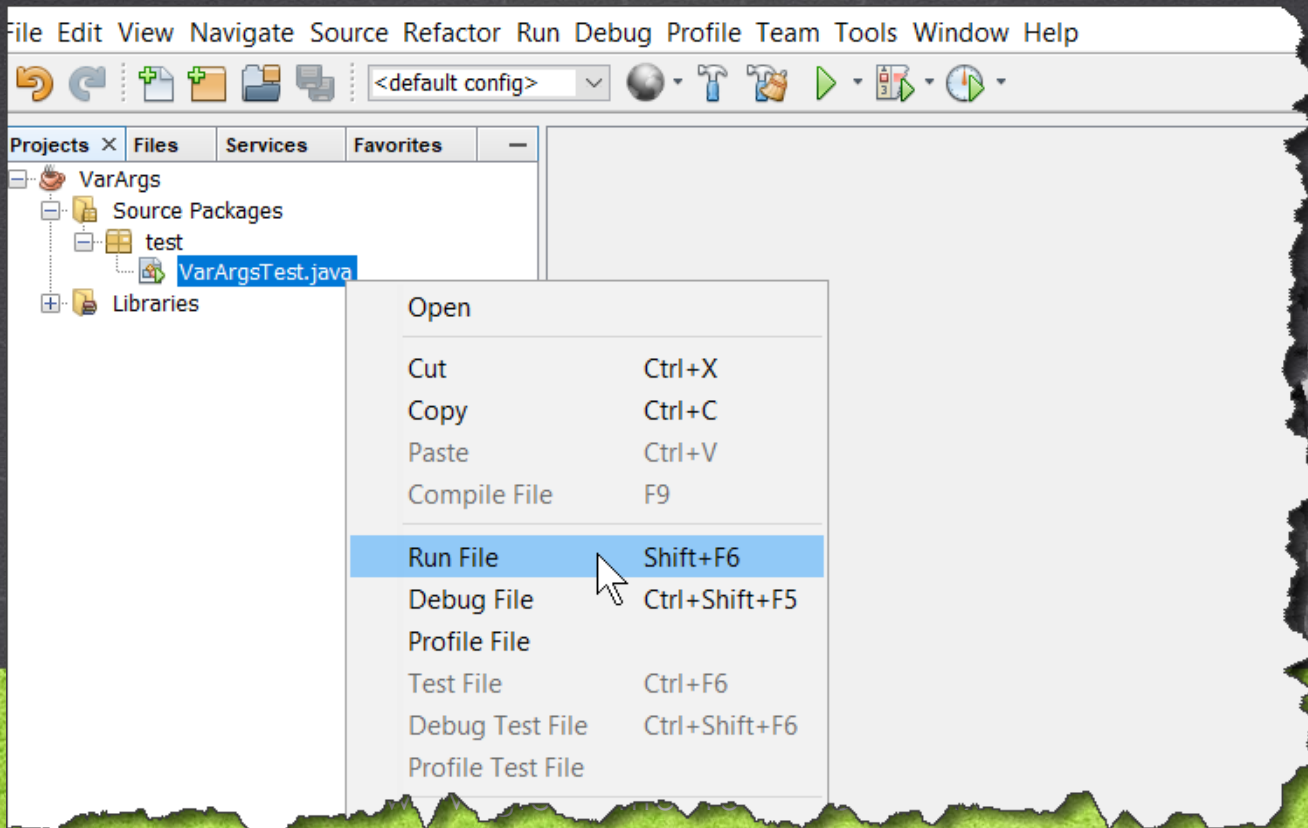
public class VarArgsTest {

    public static void main(String[] args) {
        //Print several numbers
        printNumbers(15,20,3,61,75,18,10);
    }

    public static void printNumbers(int... numbers){
        //We go through each element of the array
        int element;
        System.out.println("Number of elements:" + numbers.length);
        for(int i=0; i < numbers.length; i++){
            element = numbers[i];
            System.out.println("Element: " + element);
        }
    }
}
```

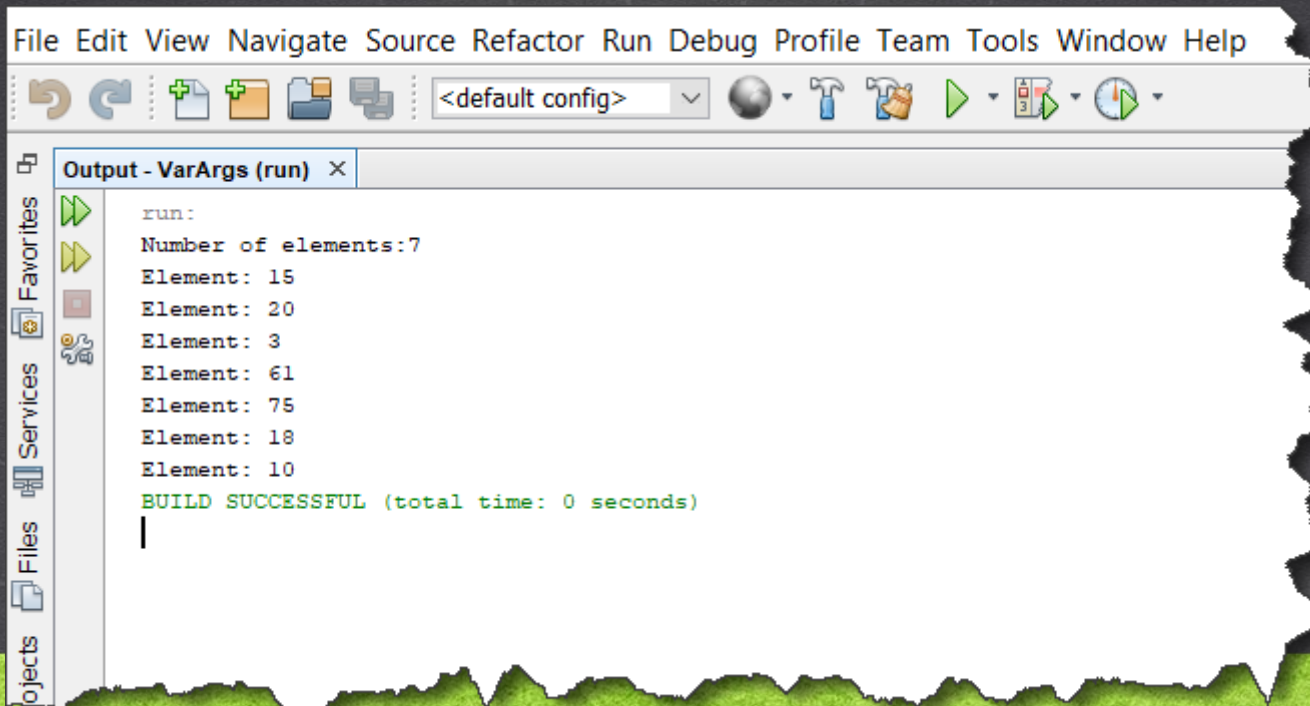

4. EXECUTE THE CLASS

Execute the class:



4. EXECUTE THE CLASS (CONT)

The result is as follows:

A screenshot of an IDE's output window. The window title is 'Output - VarArgs (run)'. The output text shows the execution of a program: 'run:', 'Number of elements:7', followed by seven 'Element:' lines with values 15, 20, 3, 61, 75, 18, and 10. It concludes with 'BUILD SUCCESSFUL (total time: 0 seconds)'. The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help) and a toolbar with icons for undo, redo, new, open, save, copy, paste, and run. A sidebar on the left shows 'Objects', 'Files', 'Services', 'Favorites', and 'Run and Debug' icons.

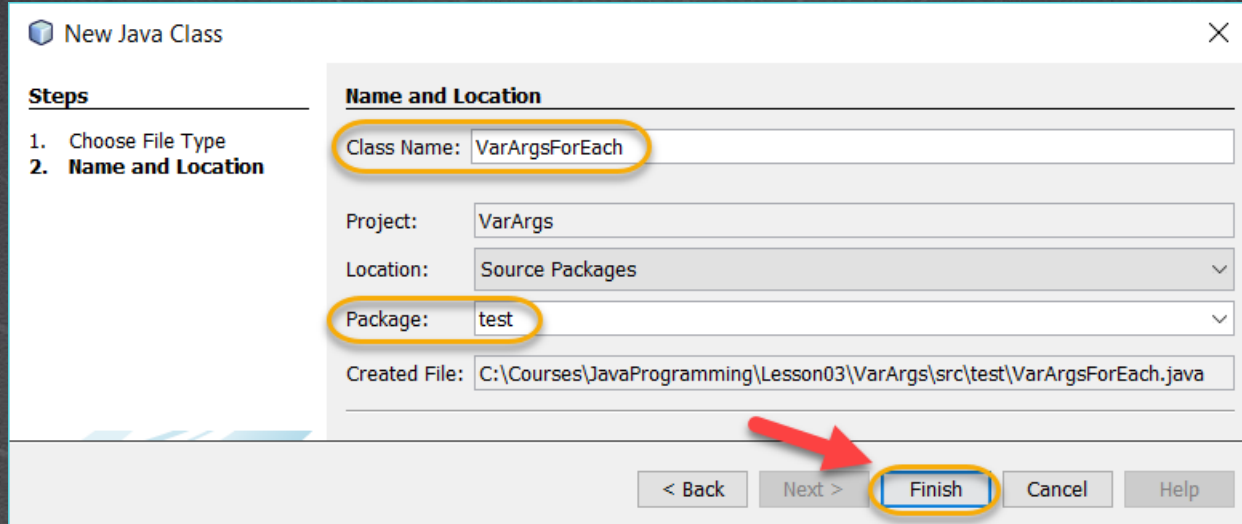
```
run:
Number of elements:7
Element: 15
Element: 20
Element: 3
Element: 61
Element: 75
Element: 18
Element: 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

5. CREATE A NEW CLASS

Create a new Java class:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

6. MODIFY THE CODE

VarArgsForEach.java:

```
package test;

public class VarArgsForEach {

    public static void main(String[] args) {

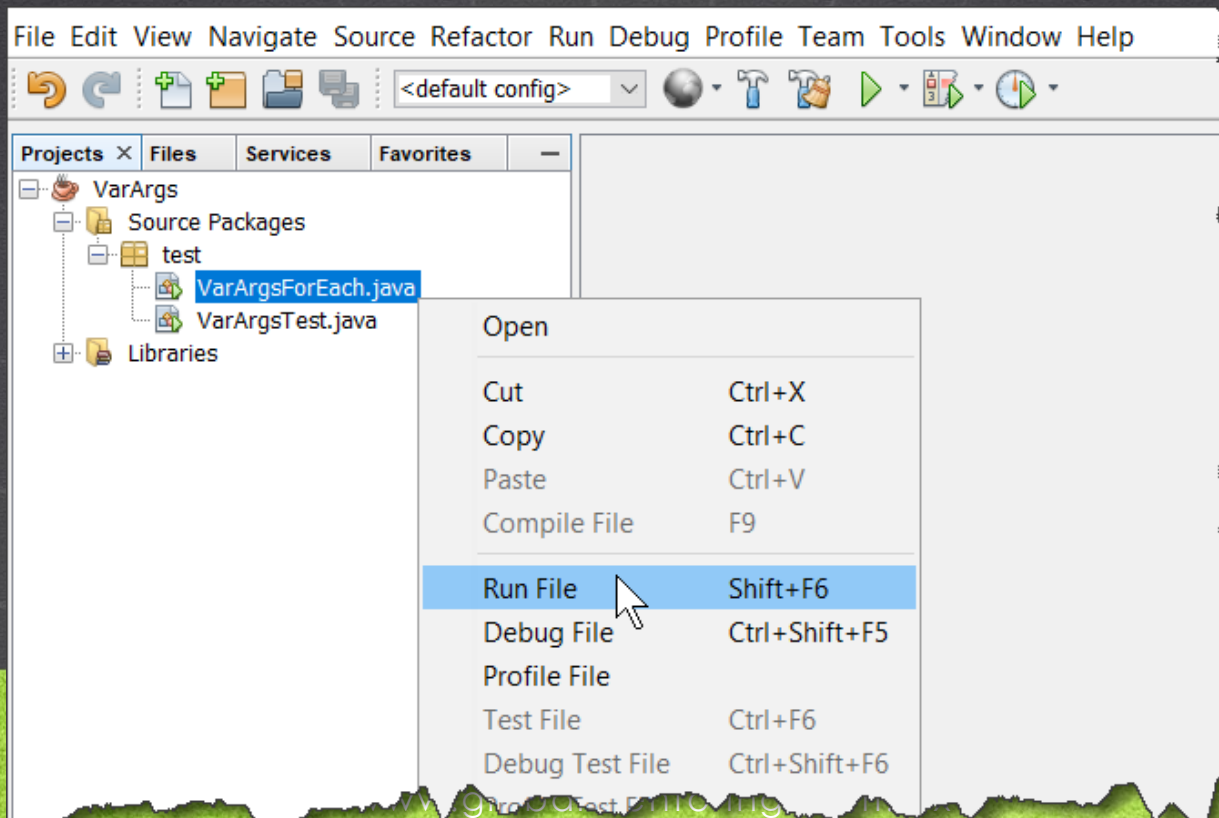
        printNumbersForEach(15,20,3,61,75,18,10);

    }

    public static void printNumbersForEach(int... numbers){
        System.out.println("Number of elements: " + numbers.length);
        //We will use a forEach loop instead of a normal for loop
        for(int number: numbers){
            System.out.println("The number is:" + number );
        }
    }
}
```

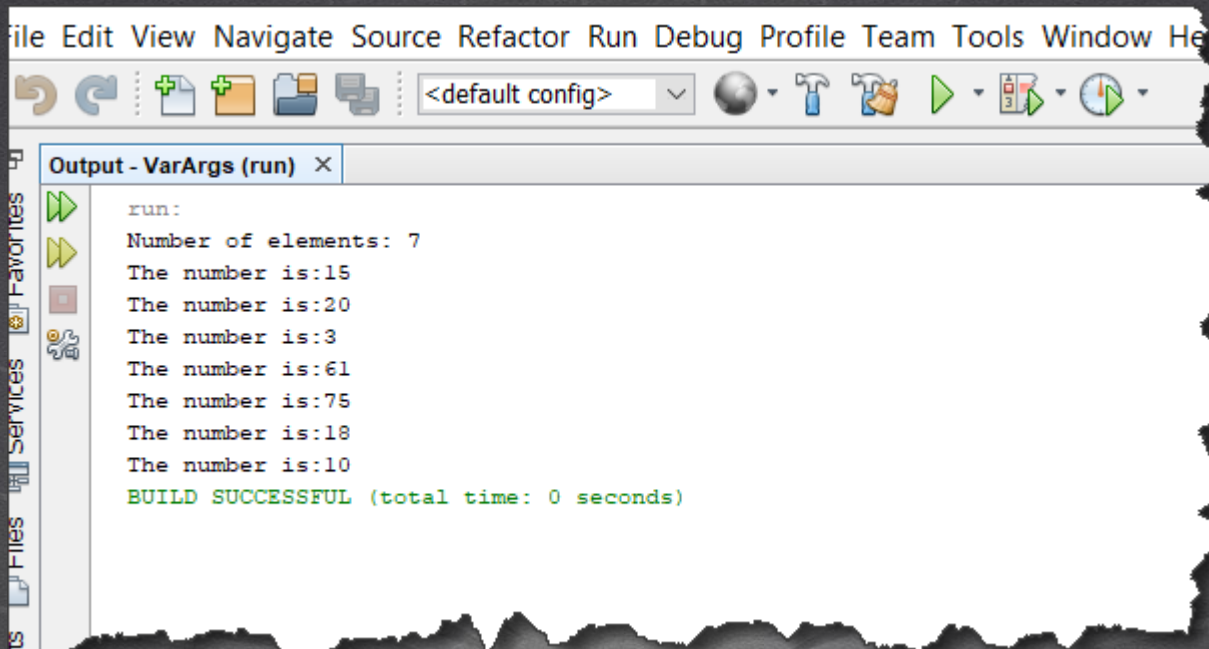
7. EXECUTE THE CLASS

Execute the class:



7. EXECUTE THE CLASS (CONT)

The result is as follows:



The screenshot shows an IDE window with the title 'Output - VarArgs (run)'. The output text is as follows:

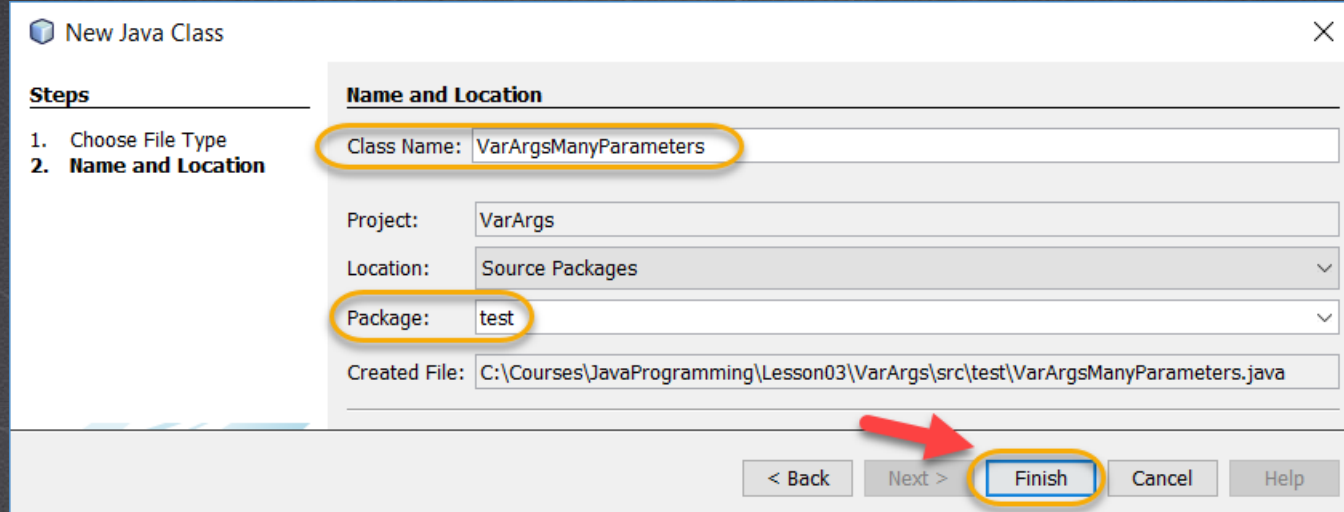
```
run:  
Number of elements: 7  
The number is:15  
The number is:20  
The number is:3  
The number is:61  
The number is:75  
The number is:18  
The number is:10  
BUILD SUCCESSFUL (total time: 0 seconds)
```

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

8. CREATE A NEW CLASS

Create a new Java class:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

9. MODIFY THE CODE

VarArgsManyParameters.java:

```
package test;

public class VarArgsManyParameters {

    public static void main(String[] args) {

        printManyArguments("John", true, 15,20,14);

    }

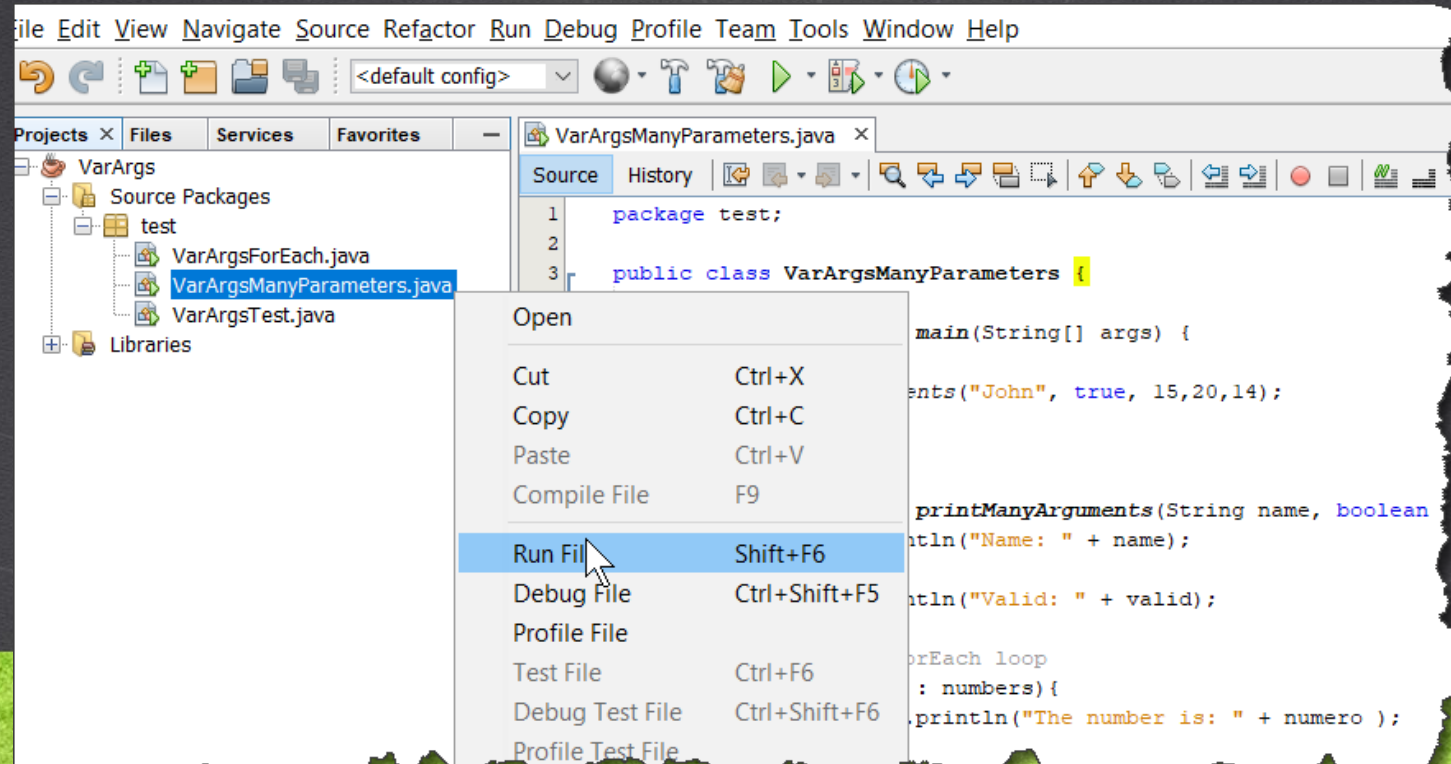
    public static void printManyArguments(String name, boolean valid, int... numbers){
        System.out.println("Name: " + name);

        System.out.println("Valid: " + valid);

        //We use the forEach loop
        for(int number : numbers){
            System.out.println("The number is: " + number);
        }
    }
}
```

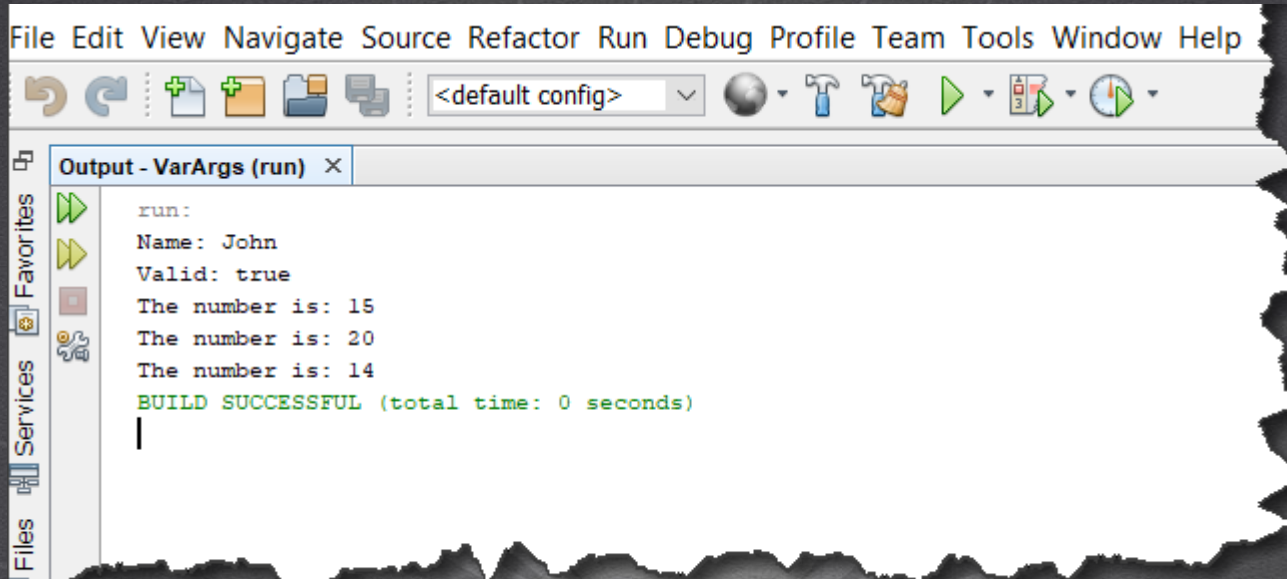

10. EXECUTE THE CLASS

Execute the class:



10. EXECUTE THE CLASS (CONT)

The result is as follows:



The screenshot shows an IDE's Output window titled "Output - VarArgs (run)". The window displays the following text:

```
run:  
Name: John  
Valid: true  
The number is: 15  
The number is: 20  
The number is: 14  
BUILD SUCCESSFUL (total time: 0 seconds)
```

The IDE's interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help) and a toolbar with various icons for file operations and execution. The left sidebar shows a "Favorites" list and a "Services" section.

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of variable arguments (varargs) in Java.
- We have seen that in addition to the use of variable arguments, we can combine it with the concept of forEach to go through each of the arguments of the created vararg array.



JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

ONLINE COURSE

JAVA PROGRAMMING

By: Eng. Ubaldo Acosta



JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx