

JAVA FUNDAMENTALS COURSE

EXERCISE

METHOD OVERLOADING IN JAVA

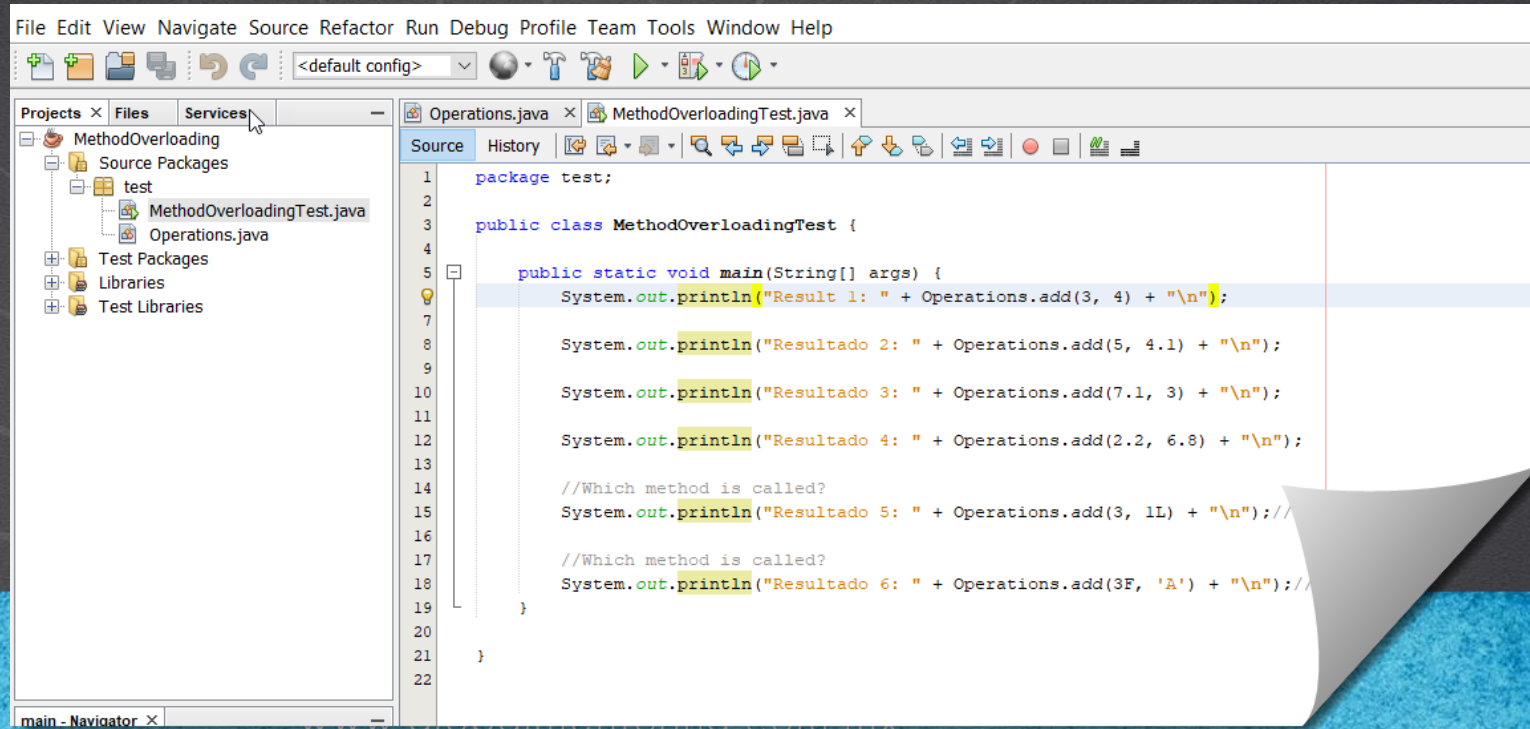


JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

With this exercise we will put into practice the concept of overload of Methods in Java. At the end we should observe the following:



1. CREATE A 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 FUNDAMENTALS COURSE

www.globalmentoring.com.mx

2. CREATE A CLASS

Create a new 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 FUNDAMENTALS COURSE

www.globalmentoring.com.mx

3. MODIFY THE CODE

Operations.java:

```
package test;

public class Operations {

    //add Method
    public static int add(int a, int b) {
        System.out.println("add (int, int) method");
        return a + b;
    }

    //add method overloading
    public static double add(double a, double b) {
        System.out.println("add (double, double) method");
        return a + b;
    }

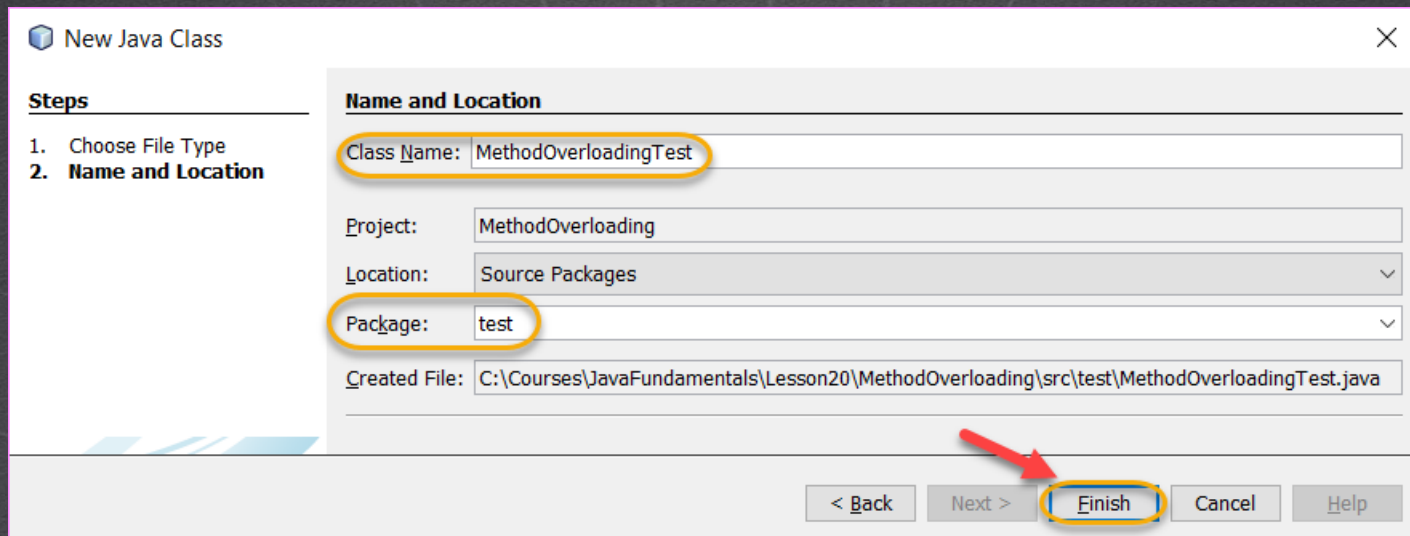
    //add method overloading
    public static double add(int a, double b) {
        System.out.println("add (int, double) method:");
        return a + b;
    }

    //add method overloading
    public static double add(double a, int b) {
        System.out.println("add (double, int) method:");
        return a + b;
    }

}
```

4. CREATE A CLASS

Create a new class:



New Java Class

Steps

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

Name and Location

Class Name: MethodOverloadingTest

Project: MethodOverloading

Location: Source Packages

Package: test

Created File: C:\Courses\JavaFundamentals\Lesson20\MethodOverloading\src\test\MethodOverloadingTest.java

< Back Next > **Finish** Cancel Help

JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

5. MODIFY THE CODE

MethodOverloadingTest.java:

```
package test;

public class MethodOverloadingTest {

    public static void main(String[] args) {
        System.out.println("\nResult 1: " + Operations.add(3, 4));

        System.out.println("\nResultado 2: " + Operations.add(5, 4.1));

        System.out.println("\nResultado 3: " + Operations.add(7.1, 3));

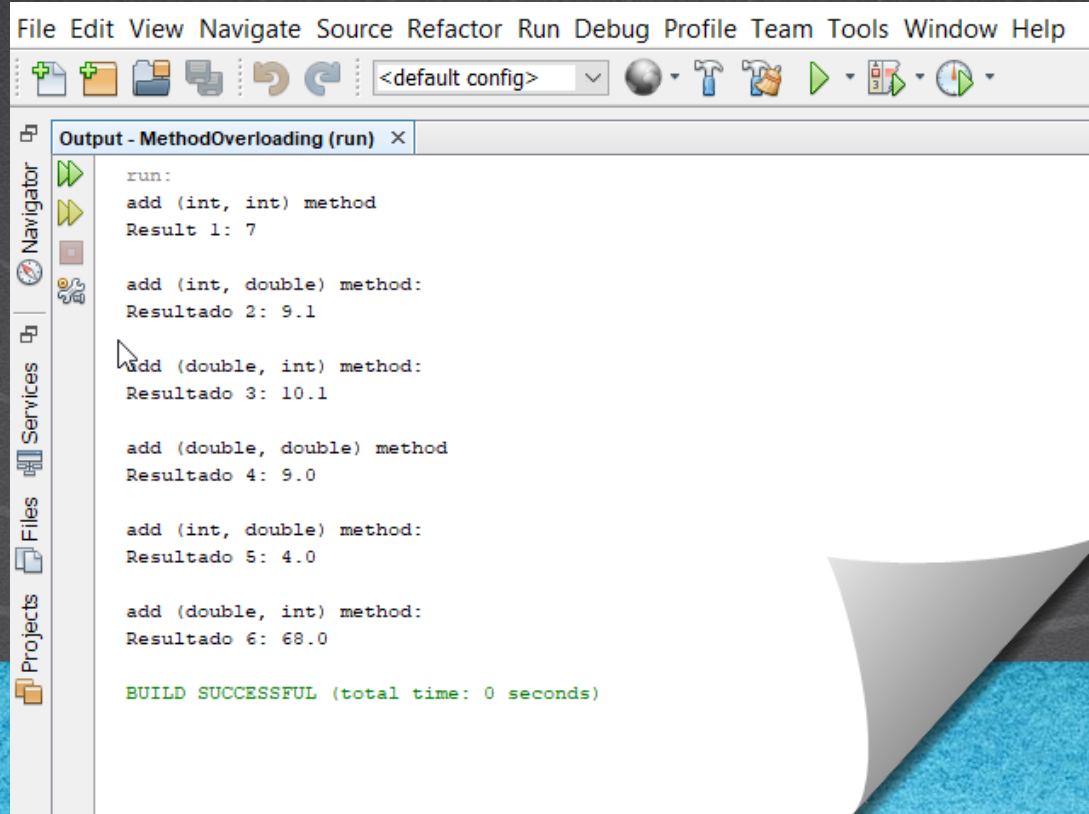
        System.out.println("\nResultado 4: " + Operations.add(2.2, 6.8));

        //Which method is called?
        System.out.println("\nResultado 5: " + Operations.add(3, 1L)); //L or l means Long

        //Which method is called?
        System.out.println("\nResultado 6: " + Operations.add(3F, 'A')); //F or f means Float
    }
}
```


PASO 6. EXECUTE THE PROJECT

The result is as follows:



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - MethodOverloading (run) X
run:
add (int, int) method
Result 1: 7

add (int, double) method:
Resultado 2: 9.1

add (double, int) method:
Resultado 3: 10.1

add (double, double) method
Resultado 4: 9.0

add (int, double) method:
Resultado 5: 4.0

add (double, int) method:
Resultado 6: 68.0

BUILD SUCCESSFUL (total time: 0 seconds)
```


EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of overloading methods in Java.
- This issue is very similar to the overload of builders, but as we have observed there are important differences and it is well worth having each of these concepts clearly.

ONLINE COURSE

JAVA FUNDAMENTALS

By: Eng. Ubaldo Acosta



JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx