

JAVA FUNDAMENTALS COURSE

EXERCISE

HANDLING STRINGS IN JAVA



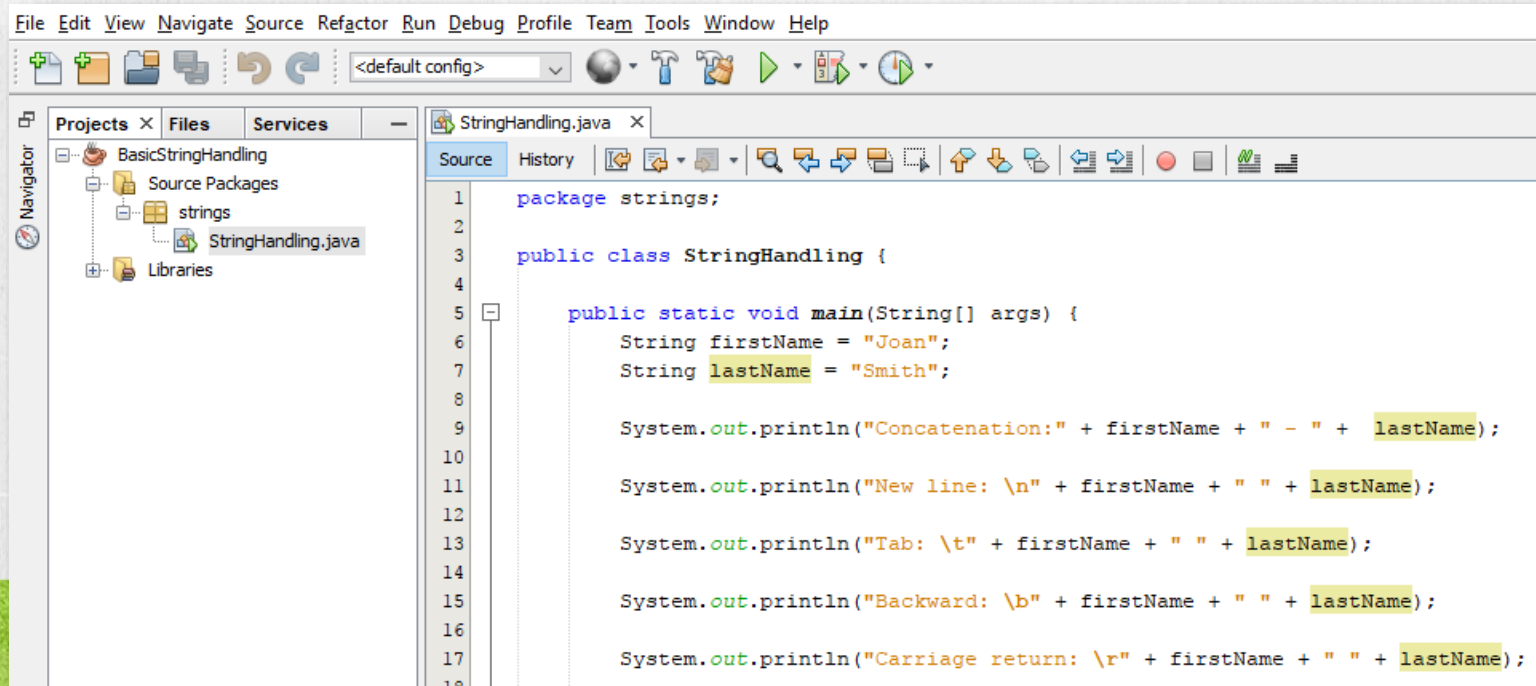
Experiencia y Conocimiento para tu vida

JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Create a program to practice String management in Java. At the end we should observe the following:

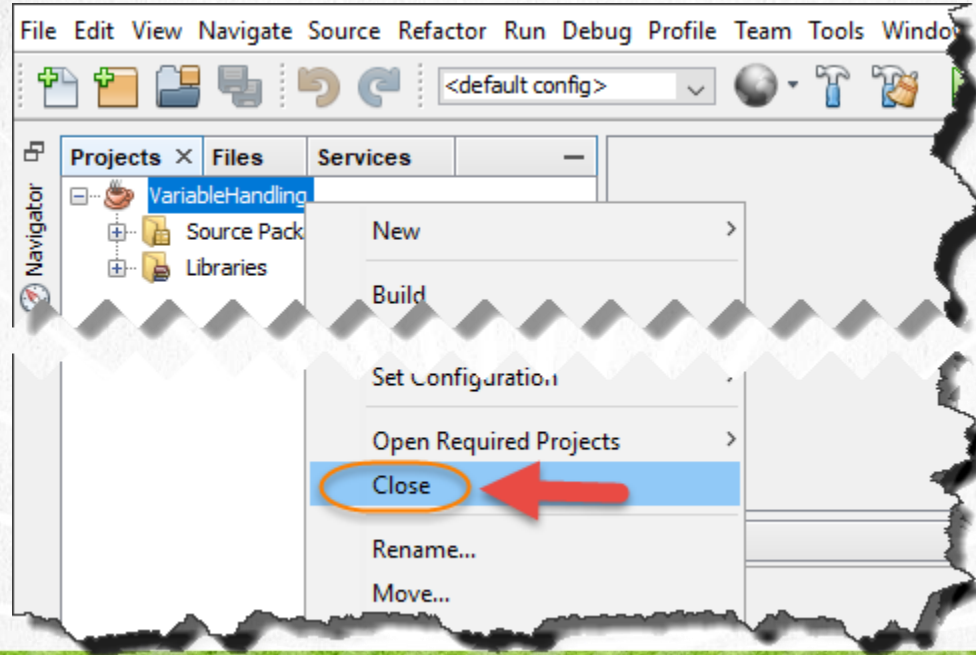


The screenshot shows an IDE window titled 'StringHandling.java'. The left sidebar contains a 'Navigator' pane with a project tree showing 'BasicStringHandling' > 'Source Packages' > 'strings' > 'StringHandling.java'. The main editor area shows the following Java code:

```
1 package strings;
2
3 public class StringHandling {
4
5     public static void main(String[] args) {
6         String firstName = "Joan";
7         String lastName = "Smith";
8
9         System.out.println("Concatenation:" + firstName + " - " + lastName);
10
11        System.out.println("New line: \n" + firstName + " " + lastName);
12
13        System.out.println("Tab: \t" + firstName + " " + lastName);
14
15        System.out.println("Backward: \b" + firstName + " " + lastName);
16
17        System.out.println("Carriage return: \r" + firstName + " " + lastName);
18    }
```

CLOSE ANY OTHER PROJECT

Preferably, close any other open project:

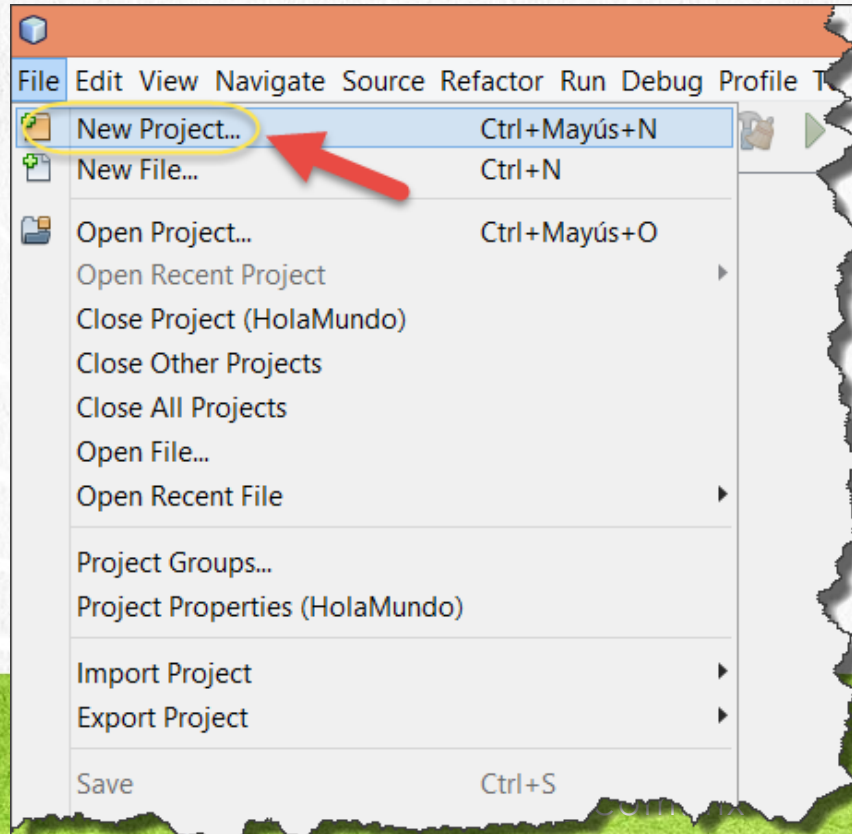


JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

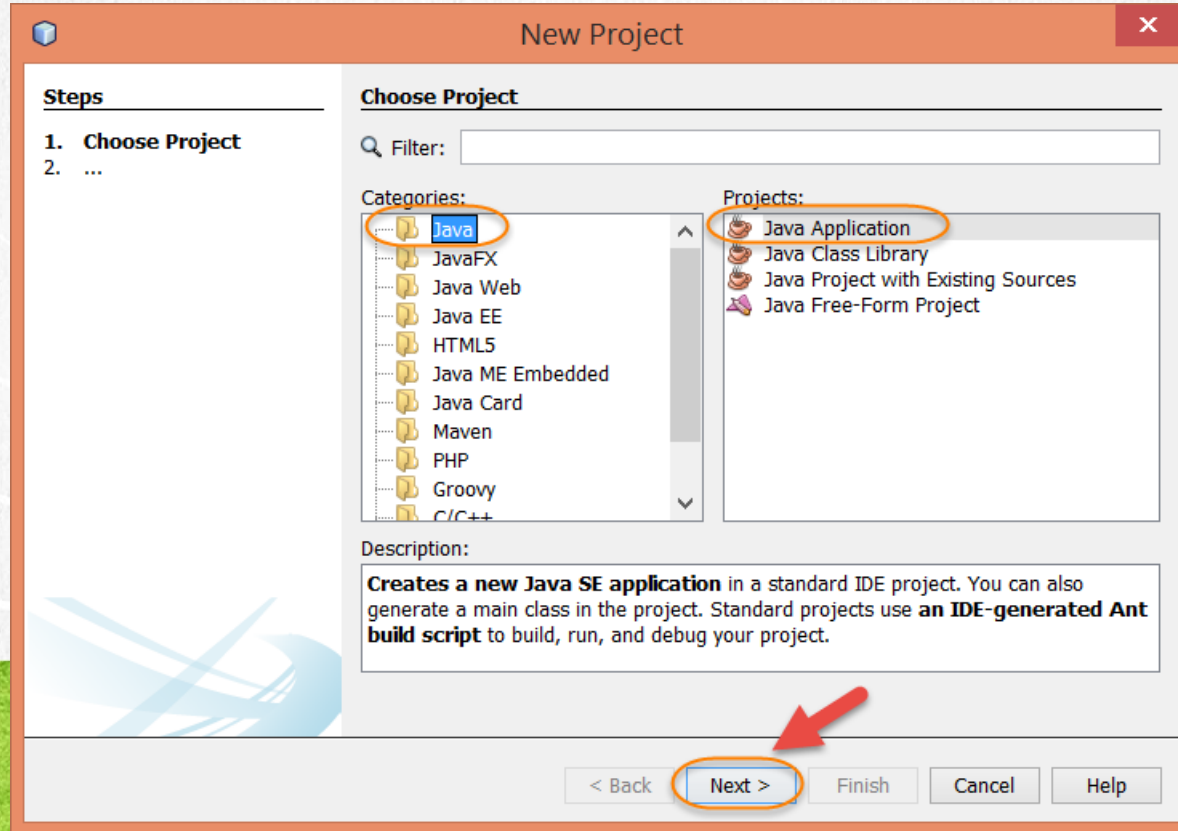
1. CREATE A NEW PROJECT

Let's create a new Project:



1. CREATE A NEW PROJECT (CONT)

We are going to create a new project called BasicStringHandling:



1. CREATE A NEW PROJECT (CONT)

We are going to create a new project called BasicStringHandling. We give the values as follows:

New Java Application

Steps

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

Name and Location

Project Name: BasicStringHandling

Project Location: C:\Courses\JavaFundamentals\Lesson02 **Browse...**

Project Folder: C:\Courses\JavaFundamentals\Lesson02\BasicStringHandling

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: **Browse...**

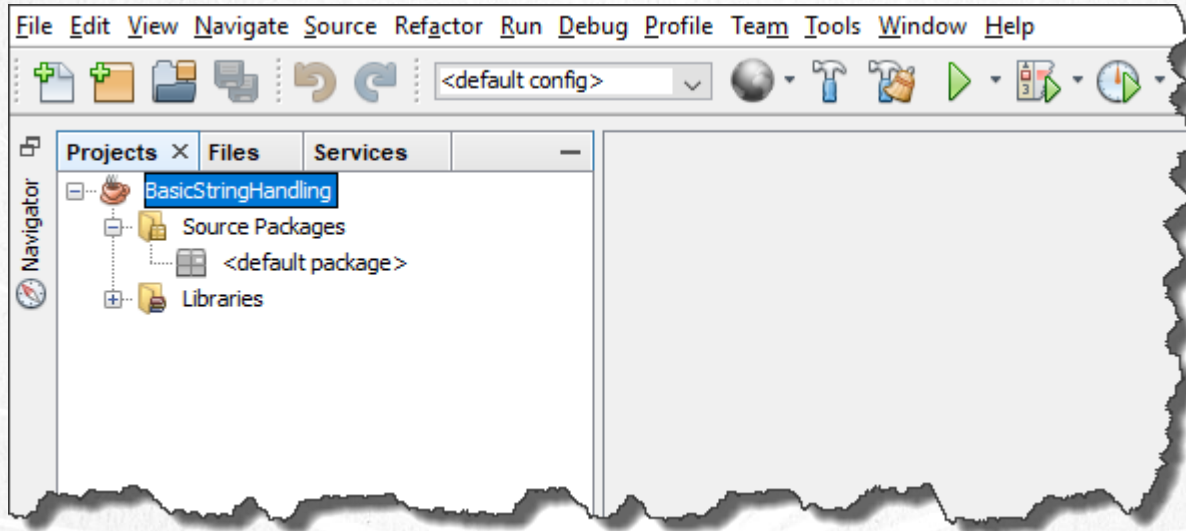
Different users and projects can share the same compilation libraries (see Help for details).

☒ **Create Main Class** basicstringhandling.BasicStringHandling

< Back Next > **Finish** Cancel Help

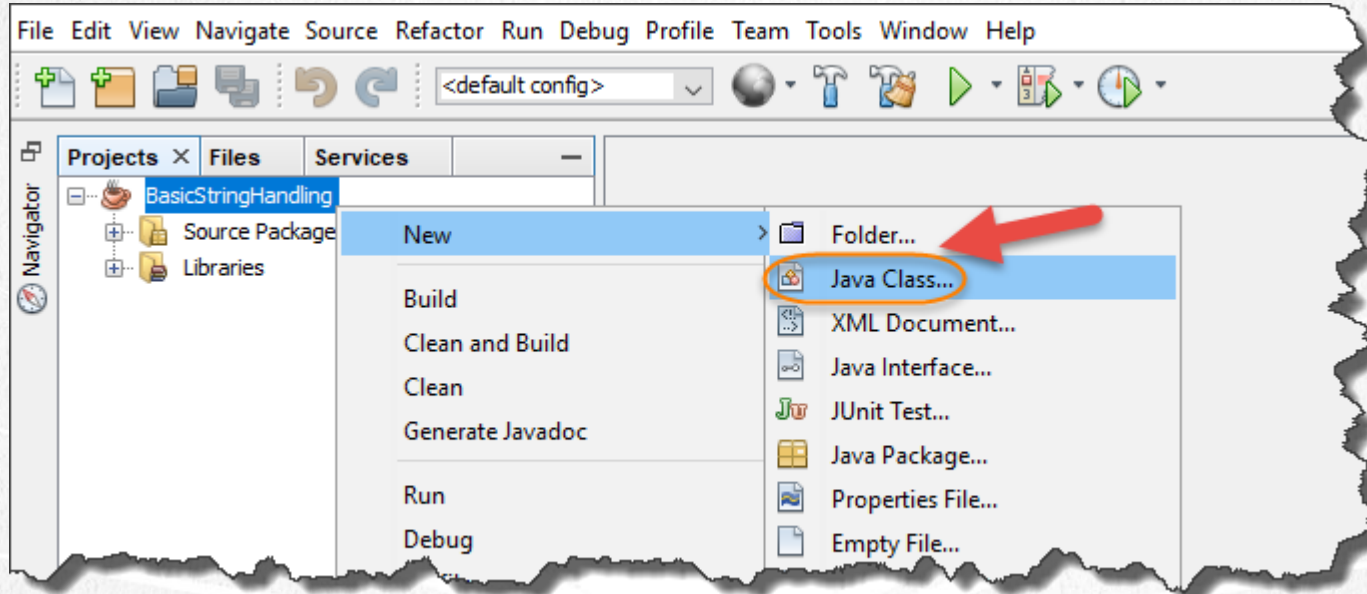
1. CREATE A NEW PROJECT (CONT)

So, the project already has the desired structure:



2. CREATE A NEW CLASS

Create a new class called StringHandling.java:



2. CREATE A NEW CLASS (CONT)

Create a new class called `StringHandling`, inside the `strings` package. The result is as show below:

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

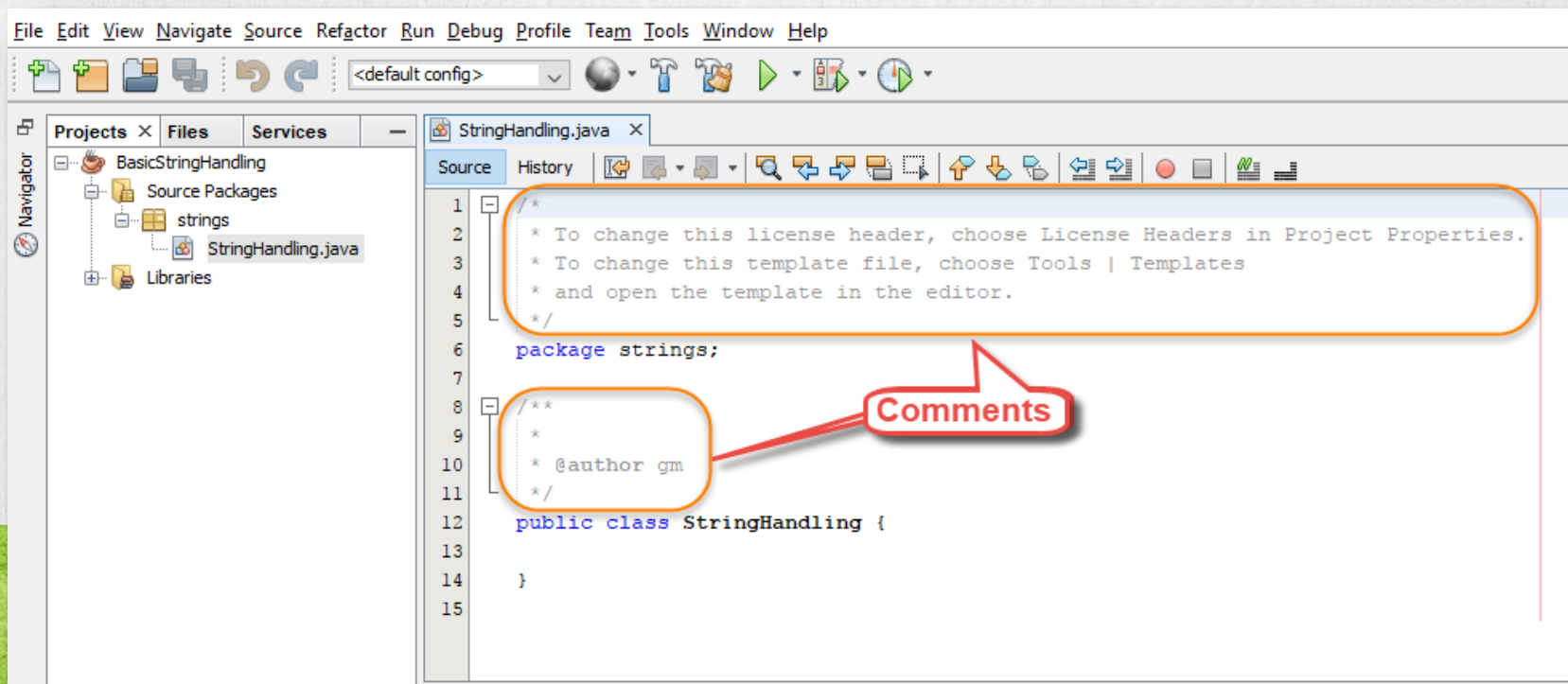
Package:

Created File:

< Back Next > **Finish** Cancel Help

2. CREATE A NEW CLASS (CONT)

The class has the desired structure. We can see some comments that do not affect the program. A comment can be placed with `//` or `/* */` or `/** */` for JavaDoc. We'll study this topic later in the course. We'll modify the class with the following code:



3. MODIFY THE CODE

[StringHandling.java:](#)

Click to download it

```
package strings;

public class StringHandling {

    public static void main(String[] args) {
        String firstName = "Joan";
        String lastName = "Smith";

        System.out.println("Concatenation:" + firstName + " - " + lastName);

        System.out.println("New line: \n" + firstName + " " + lastName);

        System.out.println("Tab: \t" + firstName + " " + lastName);

        System.out.println("Backward: \b" + firstName + " " + lastName);

        System.out.println("Carriage return: \r" + firstName + " " + lastName);

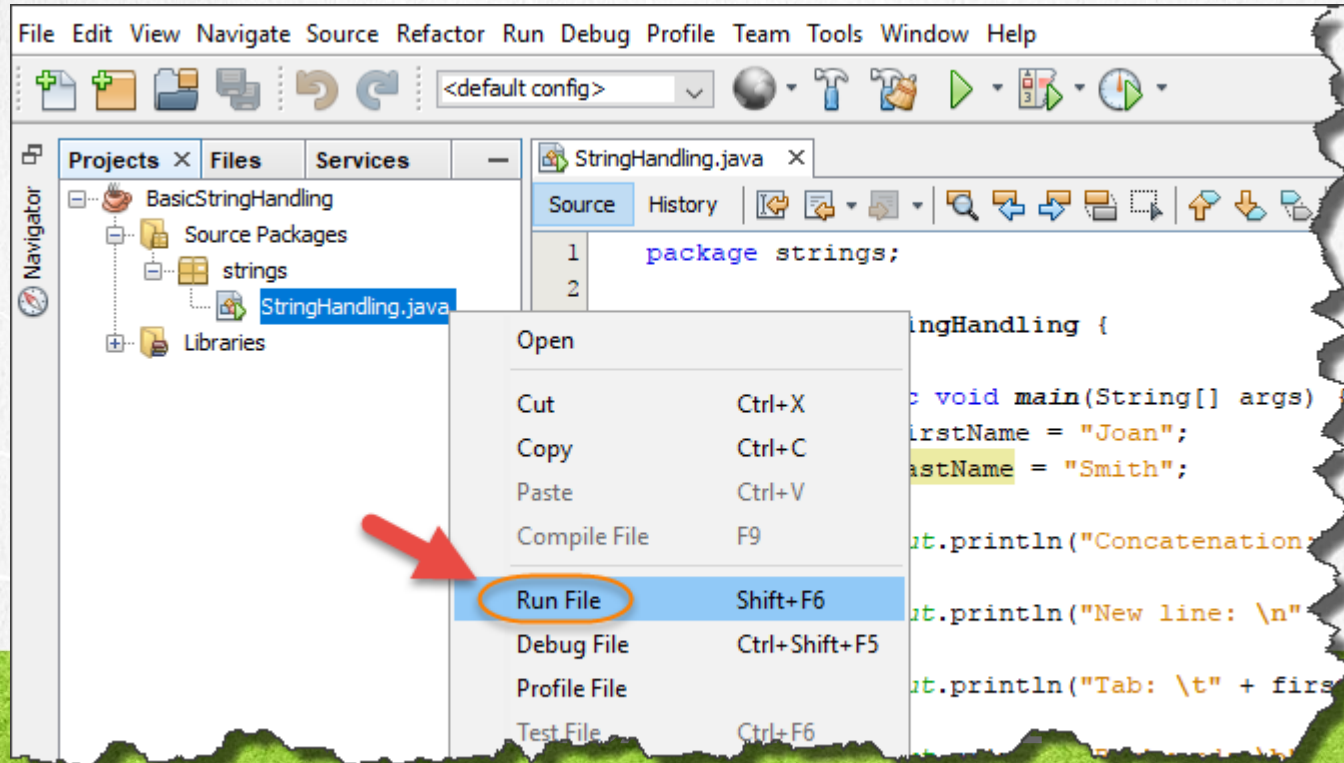
        System.out.println("Simple quote: \' " + firstName + " " + lastName + "\'");

        System.out.println("Double quote: \" " + firstName + " " + lastName + "\"");

    }
}
```

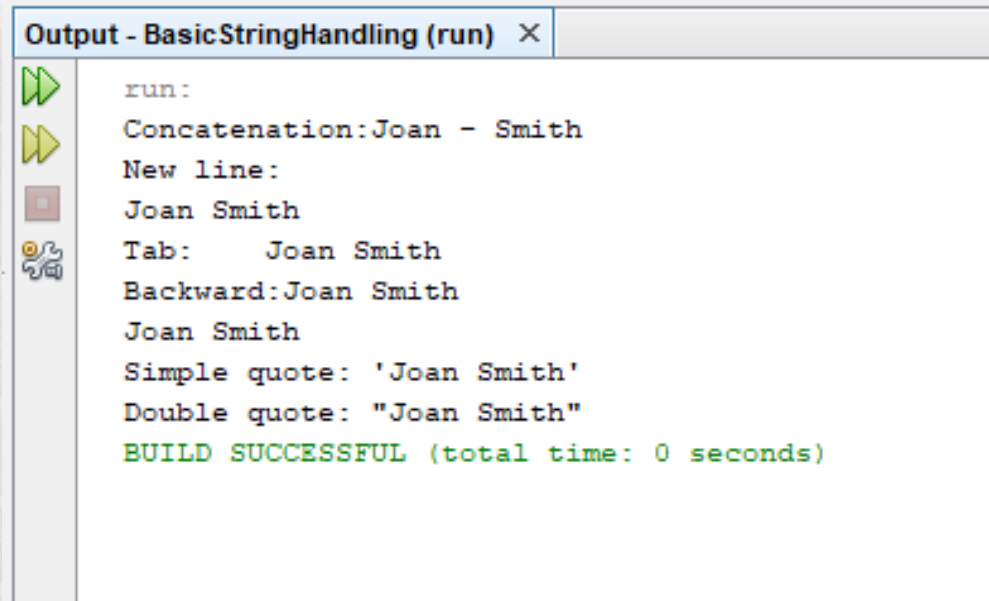

3. EXECUTE THE PROJECT

We execute our project. We give right click -> Run:



3. EXECUTE THE PROJECT (CONT)

The result is as follows:



The screenshot shows an IDE output window titled "Output - BasicStringHandling (run) X". On the left side of the window, there is a vertical toolbar with icons for running (a green play button), stepping through code (a yellow play button), stopping (a red square), and debugging (a bug icon). The main area of the window displays the output of the program in a monospaced font. The output consists of several lines: "run:", "Concatenation:Joan - Smith", "New line:", "Joan Smith", "Tab: Joan Smith", "Backward:Joan Smith", "Joan Smith", "Simple quote: 'Joan Smith'", "Double quote: \"Joan Smith\"", and "BUILD SUCCESSFUL (total time: 0 seconds)".

```
run:
Concatenation:Joan - Smith
New line:
Joan Smith
Tab:    Joan Smith
Backward:Joan Smith
Joan Smith
Simple quote: 'Joan Smith'
Double quote: "Joan Smith"
BUILD SUCCESSFUL (total time: 0 seconds)
```

EXERCISE CONCLUSION

- With this exercise we have put into practice the most basic String management in Java.
- The issue of String management we will see in more detail later, but we have discussed this with the aim that we begin to be familiar with this type of data that is one of the most used in Java.



Experiencia y Conocimiento para tu vida

JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

ONLINE COURSE

JAVA FUNDAMENTALS

Author: Ubaldo Acosta



JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx