JAVA PROGRAMMING COURSE

EXERCISE

JAVABEANS IN JAVA

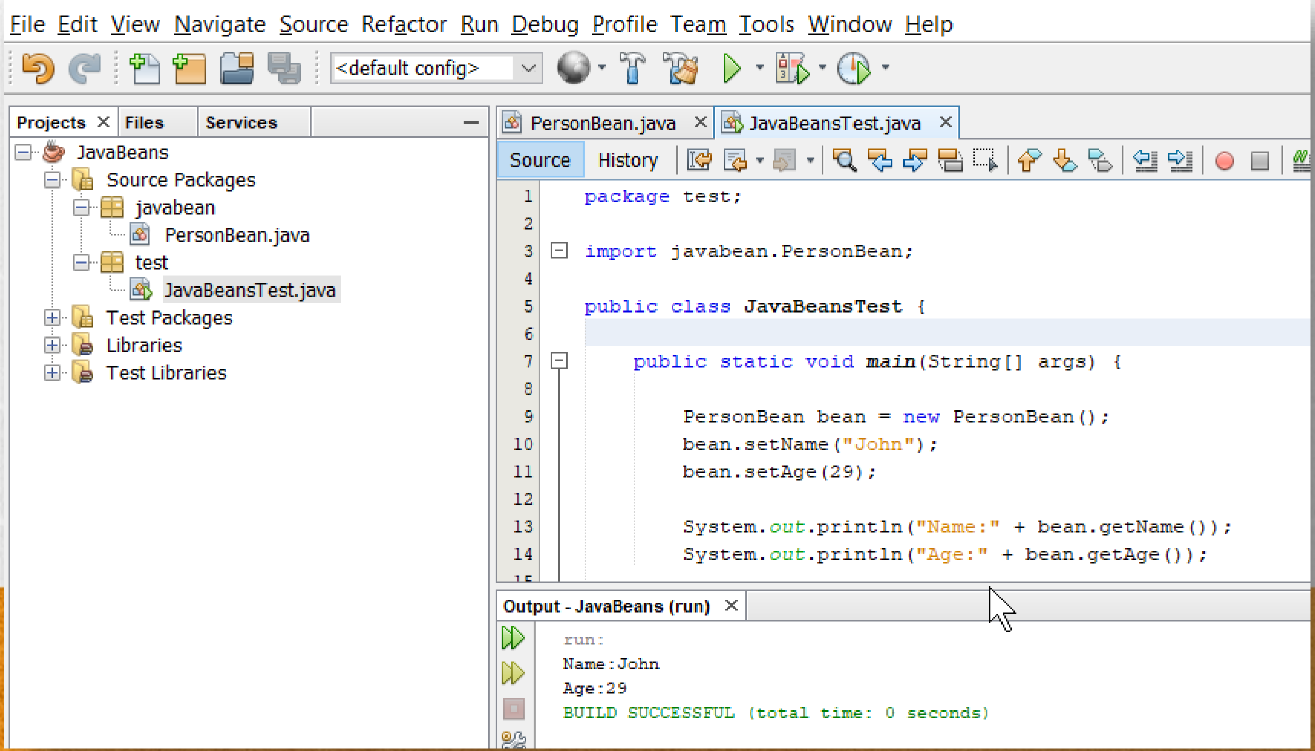
Global Mentoring

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Put into practice the concept of JavaBeans in Java. At the end we should observe the following:

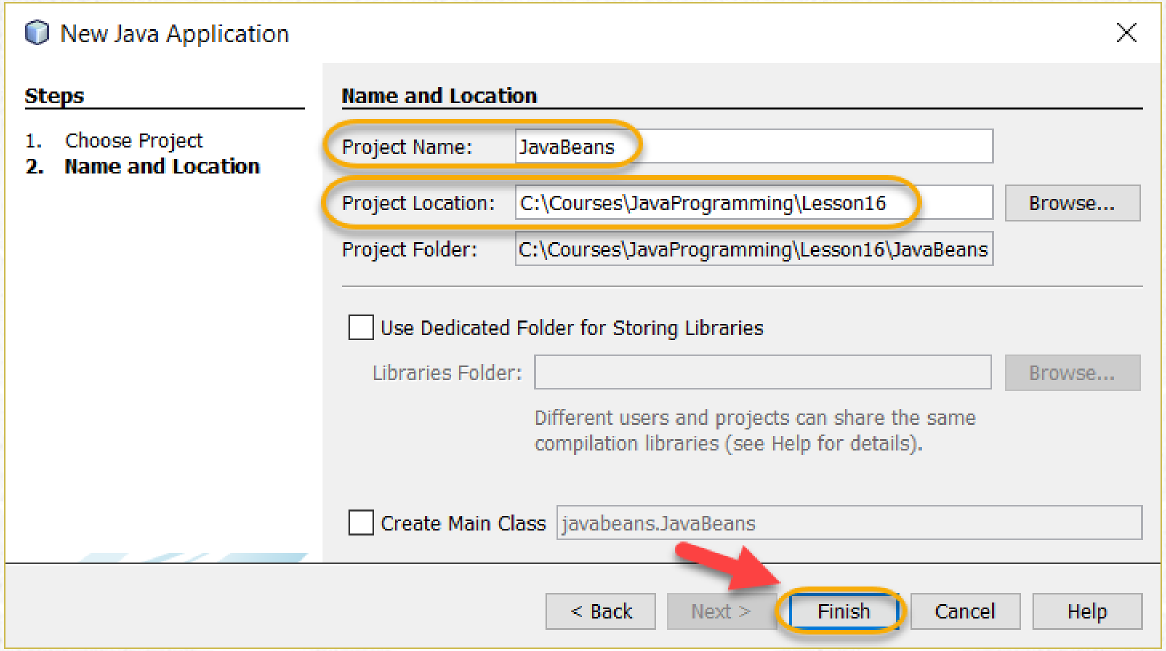


```java
package test;

import javabean.PersonBean;

public class JavaBeansTest {

    public static void main(String[] args) {

        PersonBean bean = new PersonBean();
        bean.setName("John");
        bean.setAge(29);

        System.out.println("Name:" + bean.getName());
        System.out.println("Age:" + bean.getAge());
```

```
run:
Name:John
Age:29
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: JavaBeans

Project Location: C:\Courses\JavaProgramming\Lesson16

Project Folder: C:\Courses\JavaProgramming\Lesson16\JavaBeans

□ Use Dedicated Folder for Storing Libraries

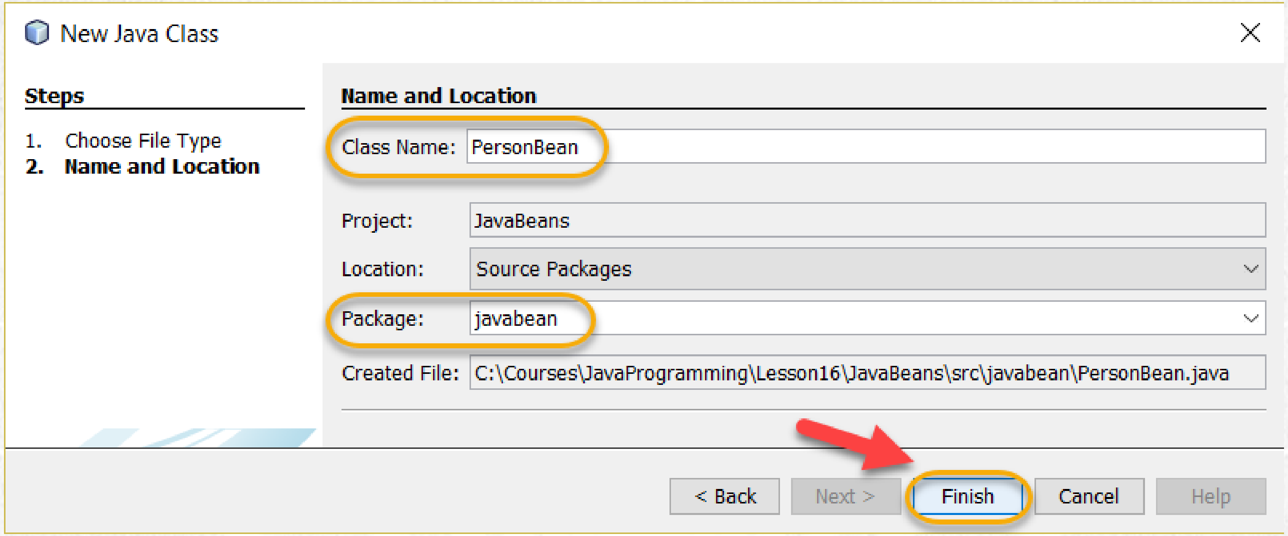
Libraries Folder:

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

□ Create Main Class javabeans.JavaBeans

2. CREATE A NEW CLASS

Create a new class:



Name and Location

Class Name: PersonBean

Project: JavaBeans

Location: Source Packages

Package: javabean

Created File: C:\Courses\JavaProgramming\Lesson16\JavaBeans\src\javabean\PersonBean.java

3. MODIFY THE CODE

PersonBean.java:

```java
package javabean;

//1. Implement the Serializable interface of the java.io package
public class PersonBean implements java.io.Serializable {

    //2. Each property is of private type
    private String name;
    private int age;

    //3. Always have a Constructor without arguments
    //Other Constructors are optional
    public PersonBean() {
    }

    //Constructor of the JavaBean with 2 arguments (Not required)
    public PersonBean(String name, int age) {
        this.name = name;
        this.age = age;
    }
```

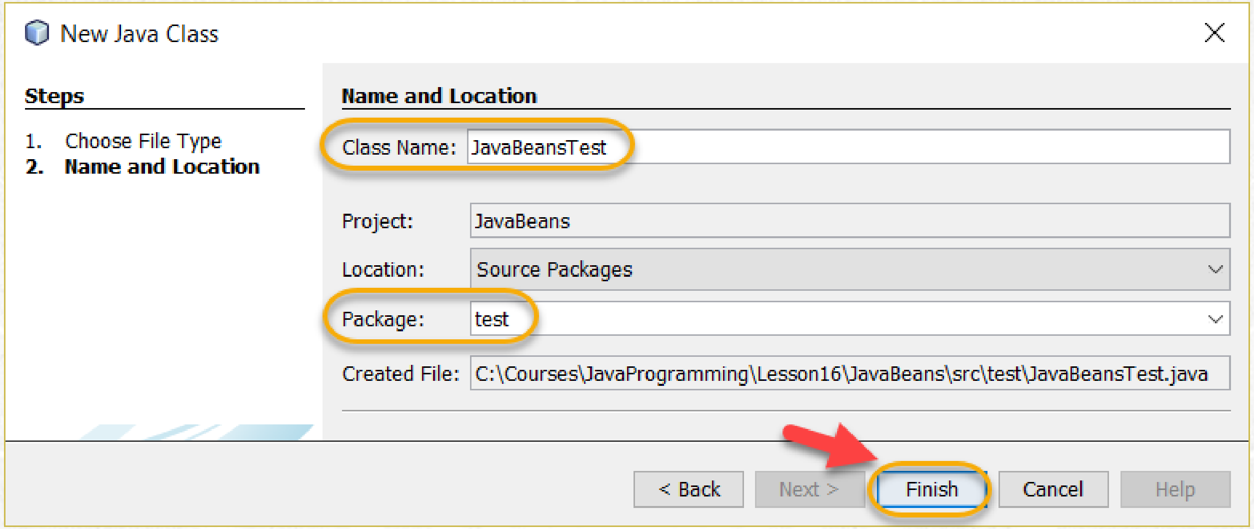
3. MODIFY THE CODE

PersonBean.java:

```java
    //4. For each property add a get and set
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

4. CREATE A NEW CLASS

Create a new class:



New Java Class

Steps

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

Name and Location

Class Name: JavaBeansTest

Project: JavaBeans

Location: Source Packages

Package: test

Created File: C:\Courses\JavaProgramming\Lesson16\JavaBeans\src\test\JavaBeansTest.java

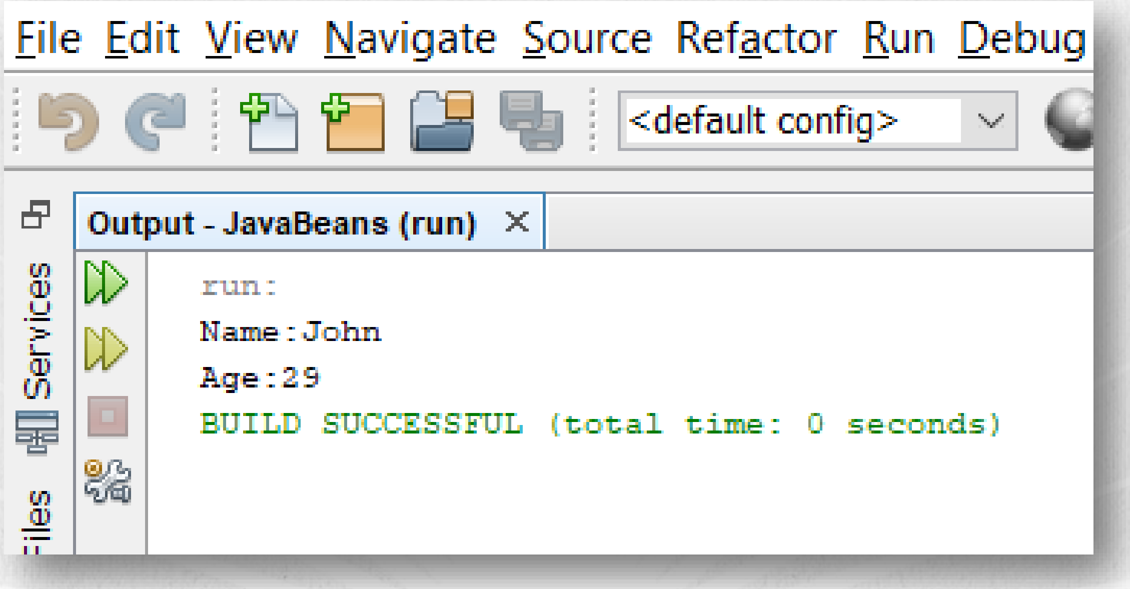
5. MODIFY THE CODE

JavaBeansTest.java:

```java
package test;

import javabean.PersonBean;

public class JavaBeansTest {

    public static void main(String[] args) {

        PersonBean bean = new PersonBean();
        bean.setName("John");
        bean.setAge(29);

        System.out.println("Name:" + bean.getName());
        System.out.println("Age:" + bean.getAge());

    }
}
```

6. EXECUTE THE PROJECT

The result is as follows:



```
run:
Name:John
Age:29
BUILD SUCCESSFUL (total time: 0 seconds)
```

EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of JavaBeans.

- We have seen that JavaBeans are very similar to the classes we have been creating, and formalize the minimum elements that our Java classes must comply with in order to be considered JavaBean. That is to say, that a JavaBean class is not very different from what we have already been working on, we just have to make sure that it contains at least the elements that we have already defined.

ONLINE COURSE

JAVA PROGRAMMING

By: Eng. Ubaldo Acosta

Global Mentoring