JAVA FUNDAMENTALS COURSE

ENCAPSULATION IN JAVA

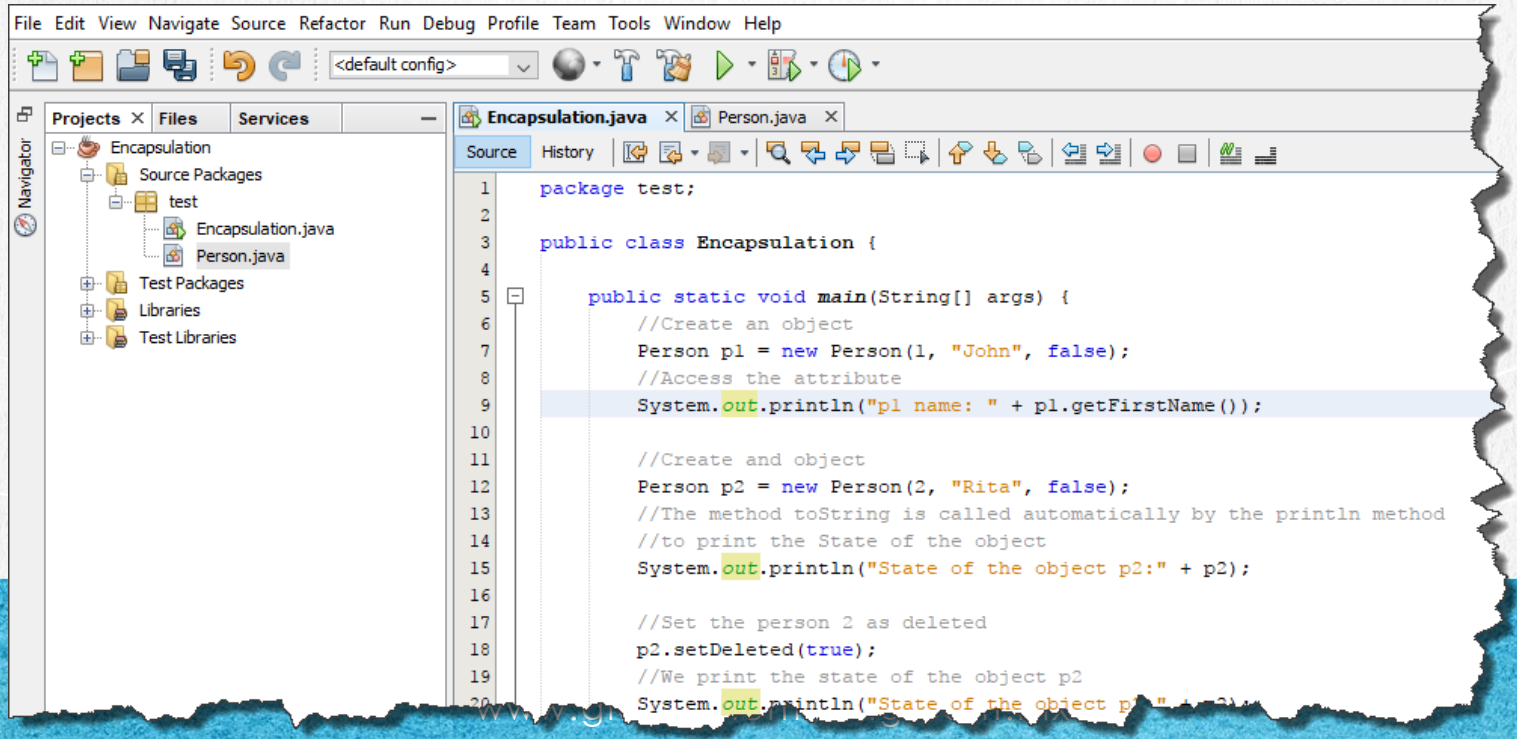
By the Expert: Ubaldo Acosta

Global Mentoring

JAVA ONLINE COURSES UNIVERSITY

EXERCISE OBJECTIVE

Implement the concept of encapsulation in Java. At the end we should observe the following:

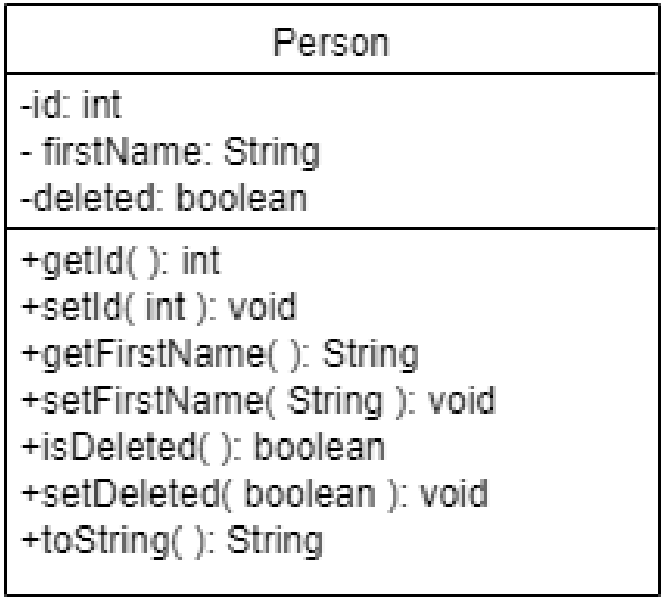


EXERCISE OBJECTIVE

We are going to create a class called Persona, which we will apply the concept of encapsulation as follows:

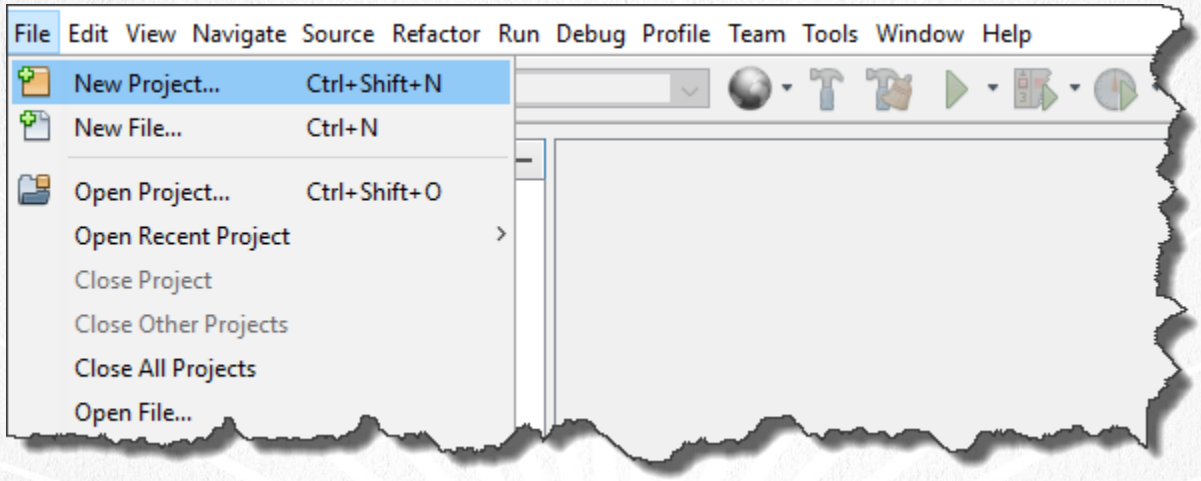
Private attributes

Public methods



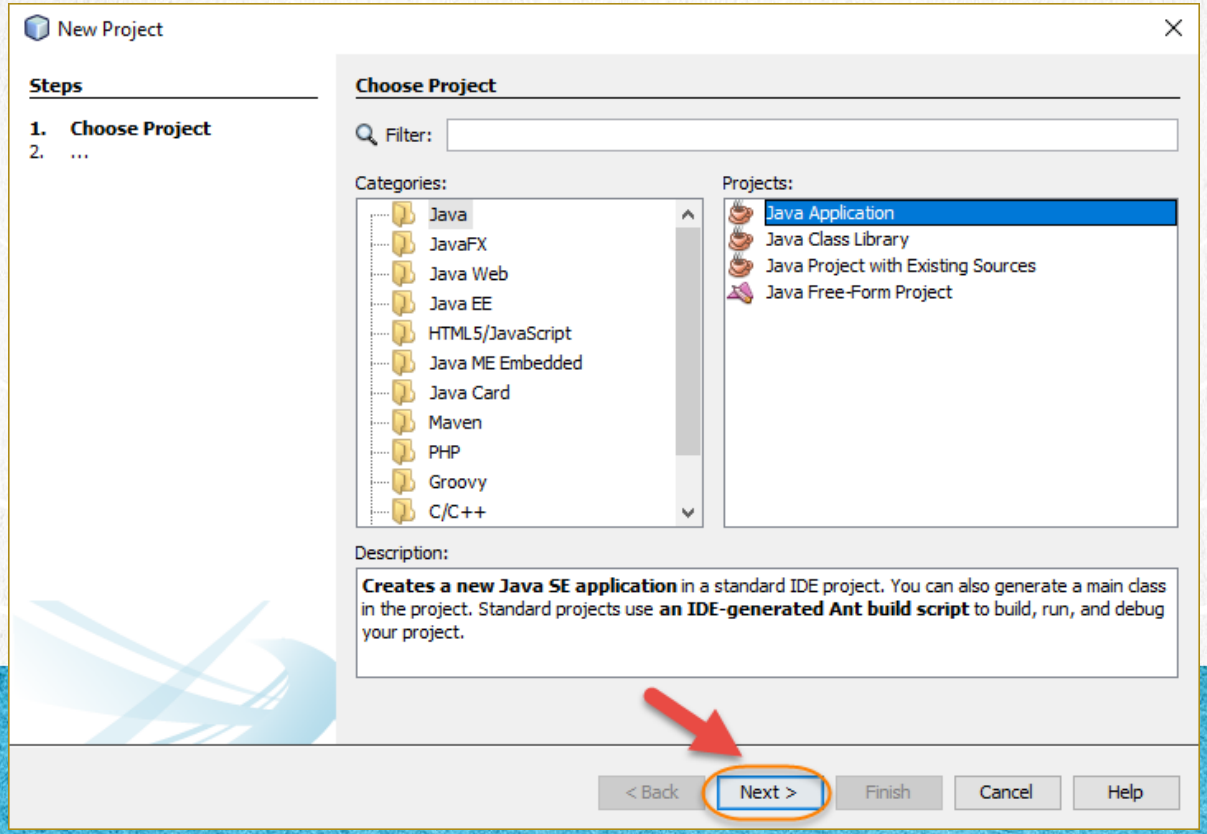
1. CREATE A NEW PROJECT

Create a new project:



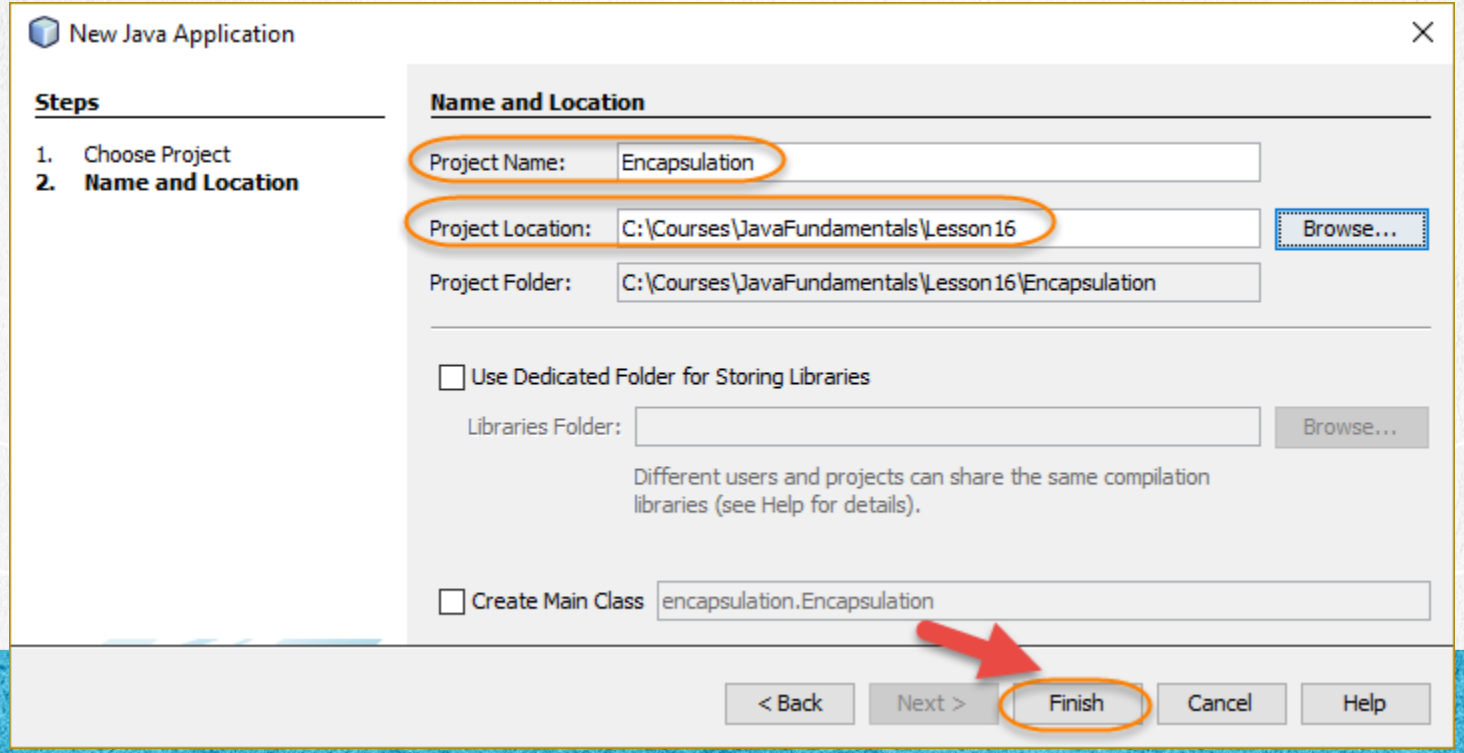
1. CREATE A NEW PROJECT

Create a new project:



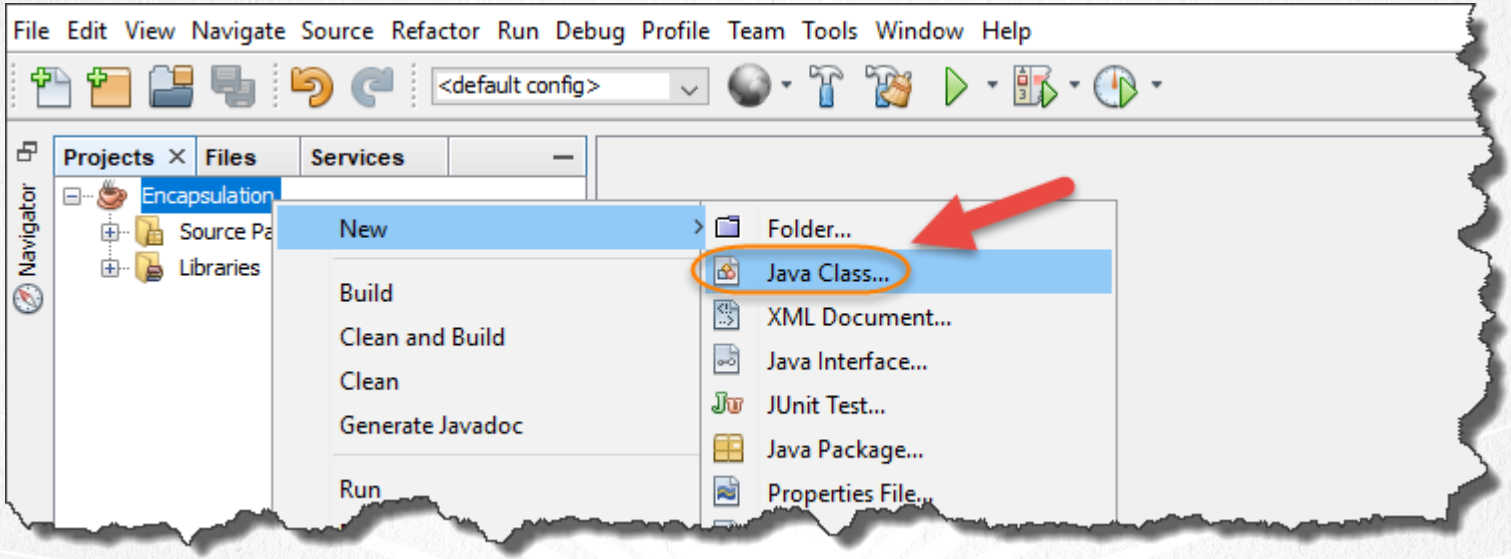
1. CREATE A NEW PROJECT

Create a new project:



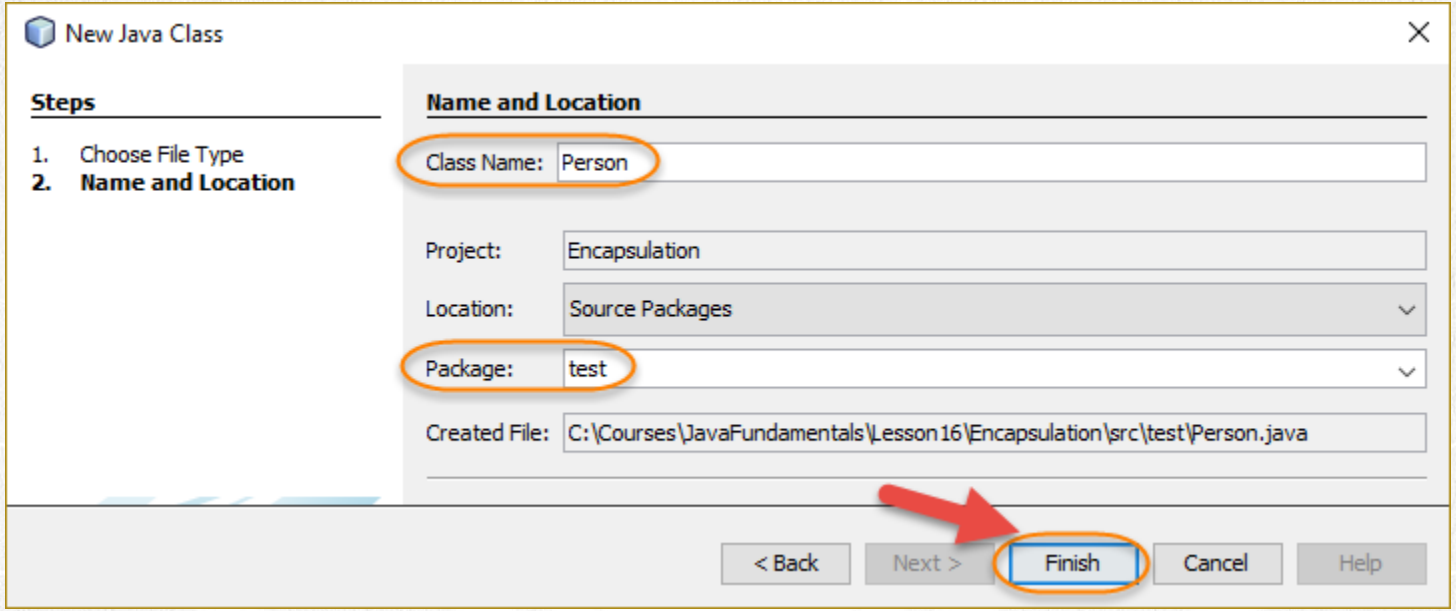
2. CREATE A NEW CLASS

Create a new class:



2. CREATE A NEW CLASS

Create a new class:



3. MODIFY THE CODE

Person.java:

```
package test;

public class Person {

    //Private attributes
    private int id;
    private String firstName;
    private boolean deleted;

    //Empty Constructor
    public Person() {
    }

    //Constructor with 3 arguments
    public Person(int id, String firstName, boolean deleted) {
        this.id = id;
        this.firstName = firstName;
        this.deleted = deleted;
    }
```

3. MODIFY THE CODE

Person.java:

```
    //Public methods to get or set every attribute
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public boolean isDeleted() {
        return deleted;
    }

    public void setDeleted(boolean deleted) {
        this.deleted = deleted;
    }
```

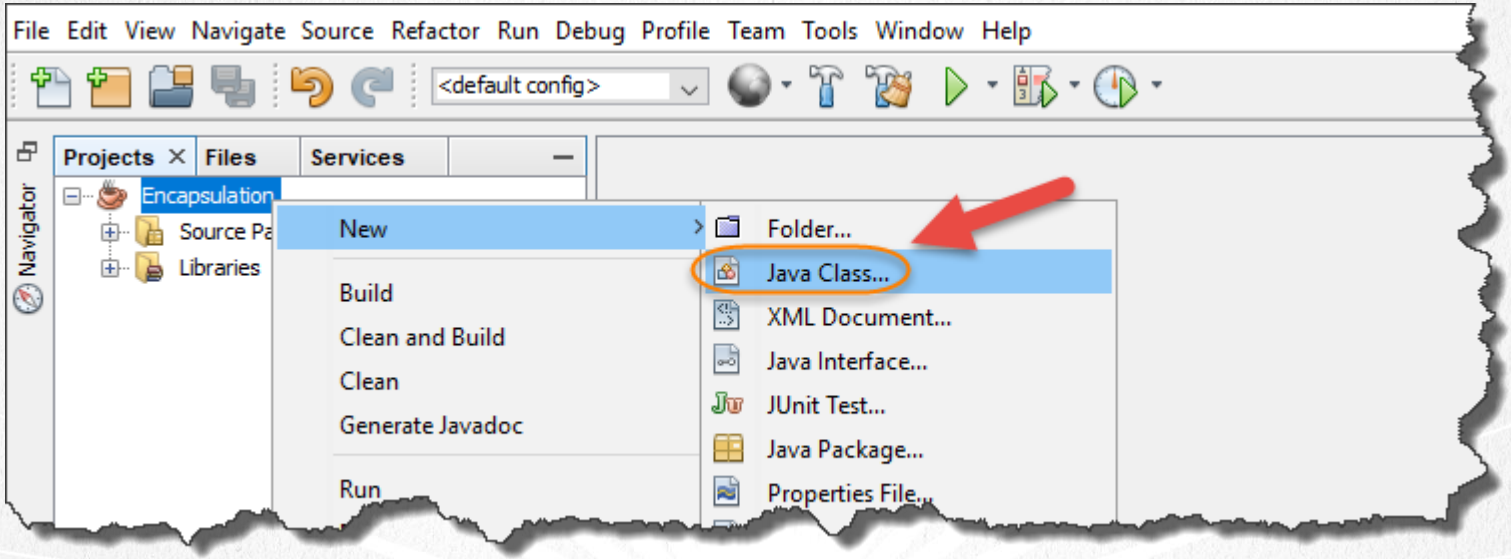
3. MODIFY THE CODE

Person.java:

```
    @Override
    public String toString() {
        return "Person{" + "id=" + id + ", firstName=" + firstName + ", deleted=" + deleted + '}';
    }

}
```

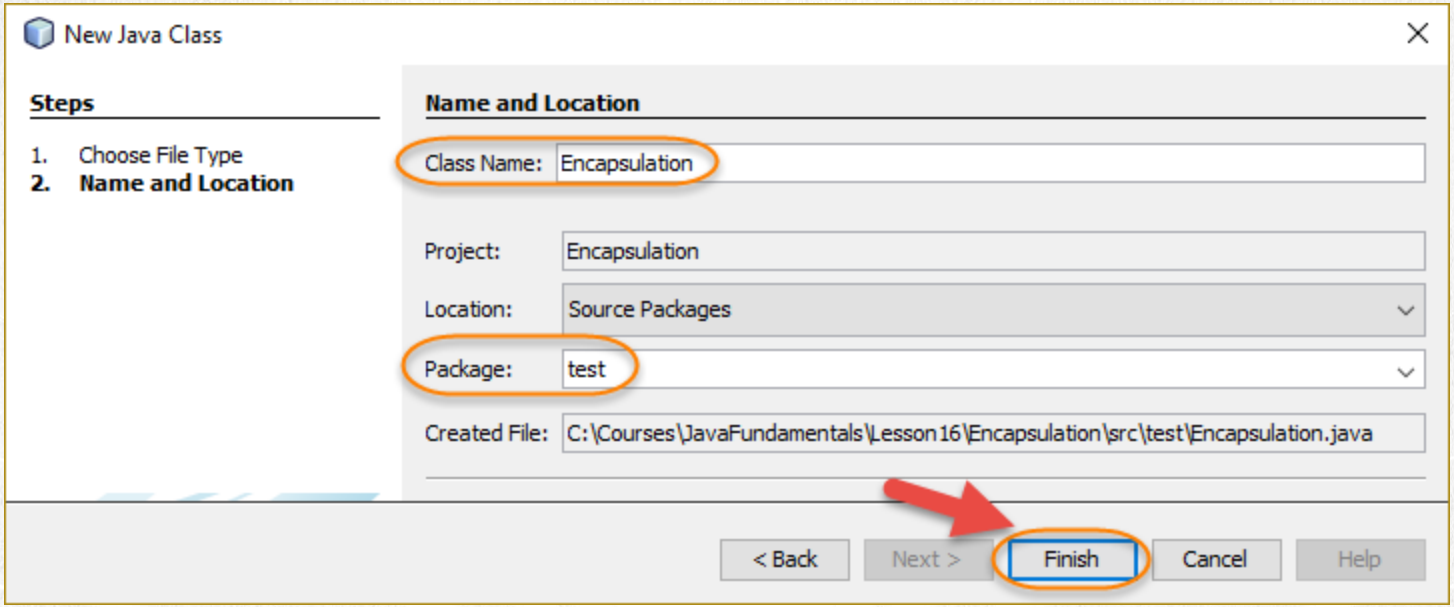
4. CREATE A NEW CLASS

Create a new class:



4. CREATE A NEW CLASS

Create a new class:



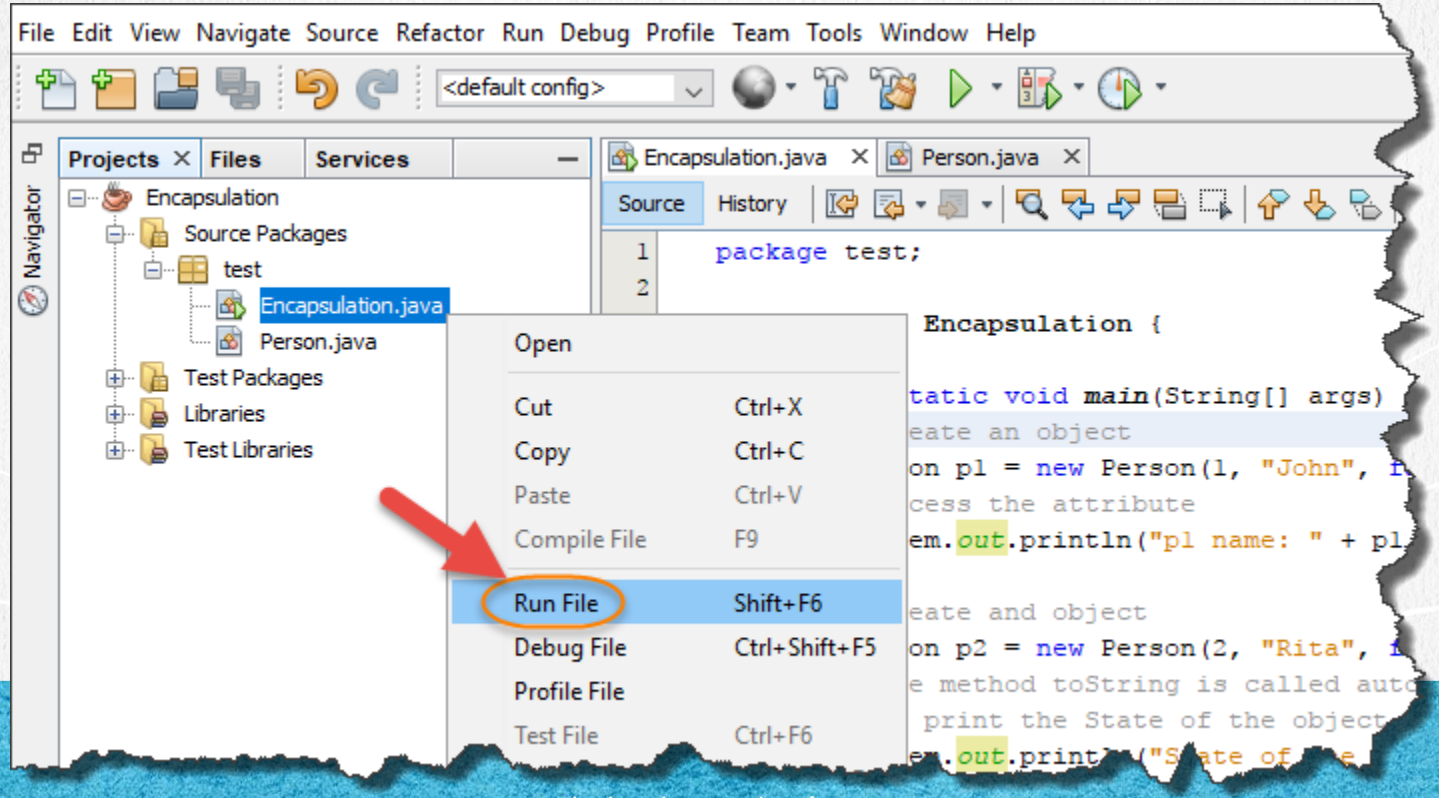
5. MODIFY THE CODE

Encapsulation.java:

```
package test;

public class Encapsulation {

    public static void main(String[] args) {
        //Create an object
        Person p1 = new Person(1, "John", false);
        //Access the attribute
        System.out.println("p1 name: " + p1.getFirstName());

        //Create and object
        Person p2 = new Person(2, "Rita", false);
        //The method toString is called automatically by the println method
        //to print the State of the object
        System.out.println("State of the object p2:" + p2);

        //Set the person 2 as deleted
        p2.setDeleted(true);
        //We print the state of the object p2
        System.out.println("State of the object p2:" + p2);

        //Create an object with the default values
        Person p3 = new Person();
        //We print the state of the object p3
        System.out.println("State of the object p3:" + p3);
    }
}
```

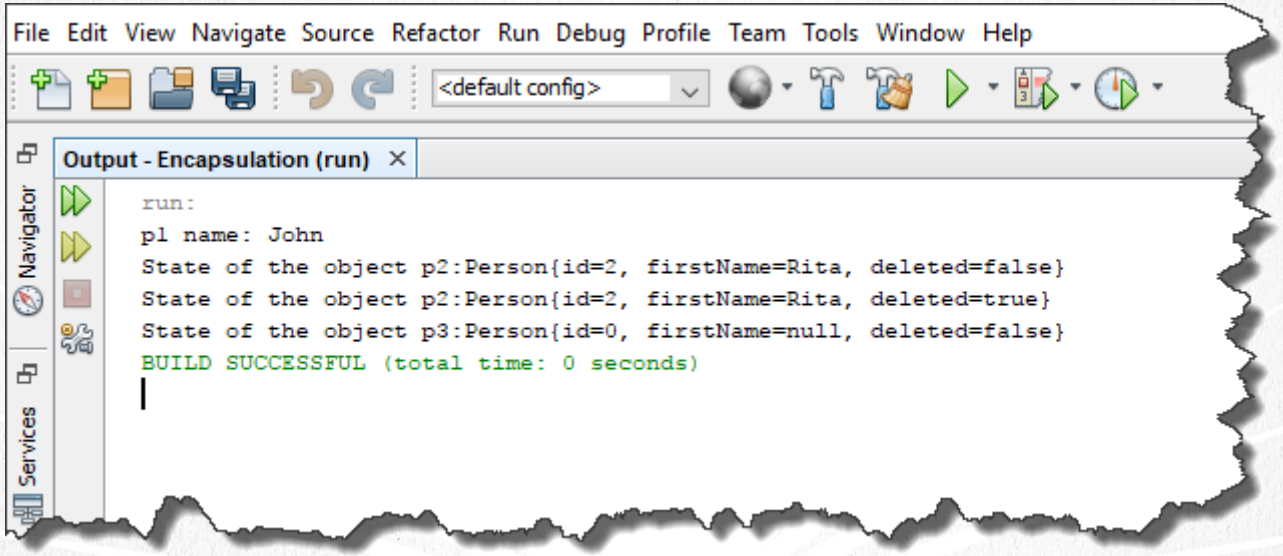
6. PROJECT EXECUTION

Execute the project:



6. PROJECT EXECUTION

The result is as follows:



EXERCISE CONCLUSION

With this exercise we have put into practice the concept of Encapsulation. We have seen how to use the private and public access modifiers to achieve this goal.

It is important to understand this concept in detail, since it is the basis for many of the classes that we will be developing throughout this and the following courses.

Note: The toString () method allows us to convert a Java object into a string, so we use it to send the state of the object in Java. To understand the explanation of this method in detail we need to have studied the inheritance and polymorphism issues, however, applying it, as we have seen, is really simple, we just have to add it to our class and concatenate the attributes to show the state of our object, it is say, the values of each attribute, that is all we have to do to implement it, however to understand its operation in detail we will study it later.

ONLINE COURSE

JAVA FUNDAMENTALS

By: Eng. Ubaldo Acosta