EXERCISE

INHERITANCE IN JAVA

EXERCISE OBJECTIVE

Create an exercise to apply the concept of Inheritance in Java. At the end we should observe the following:

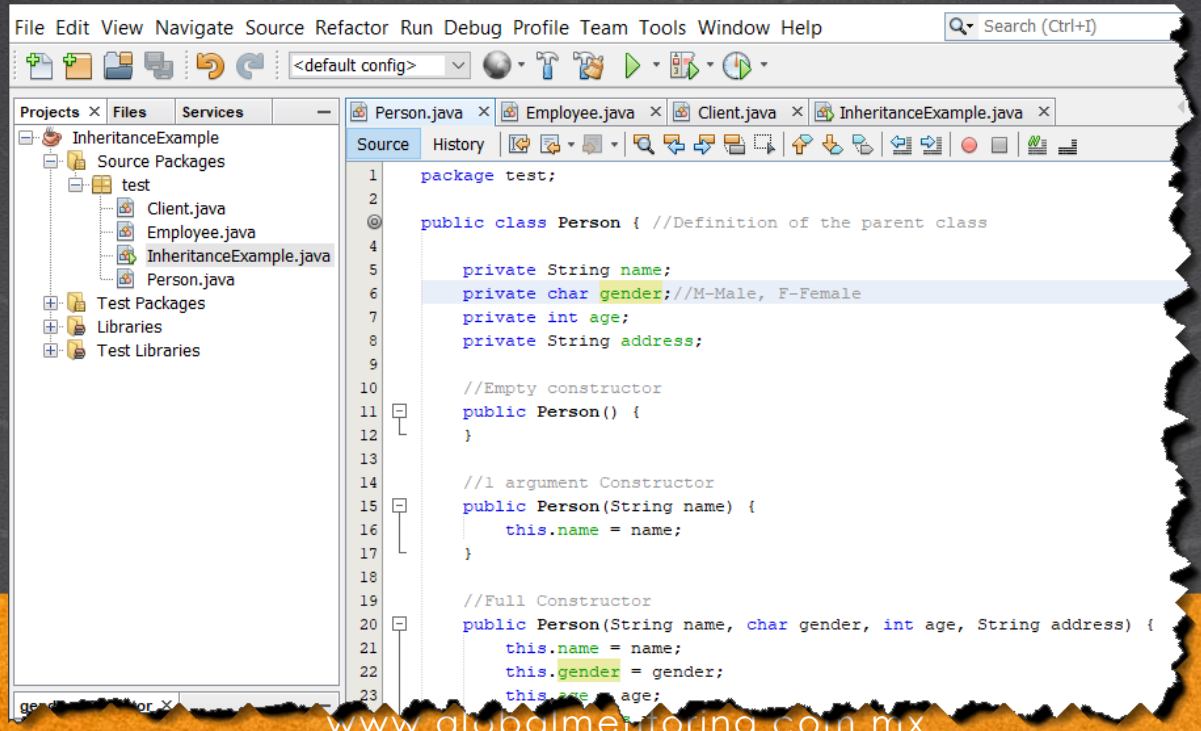
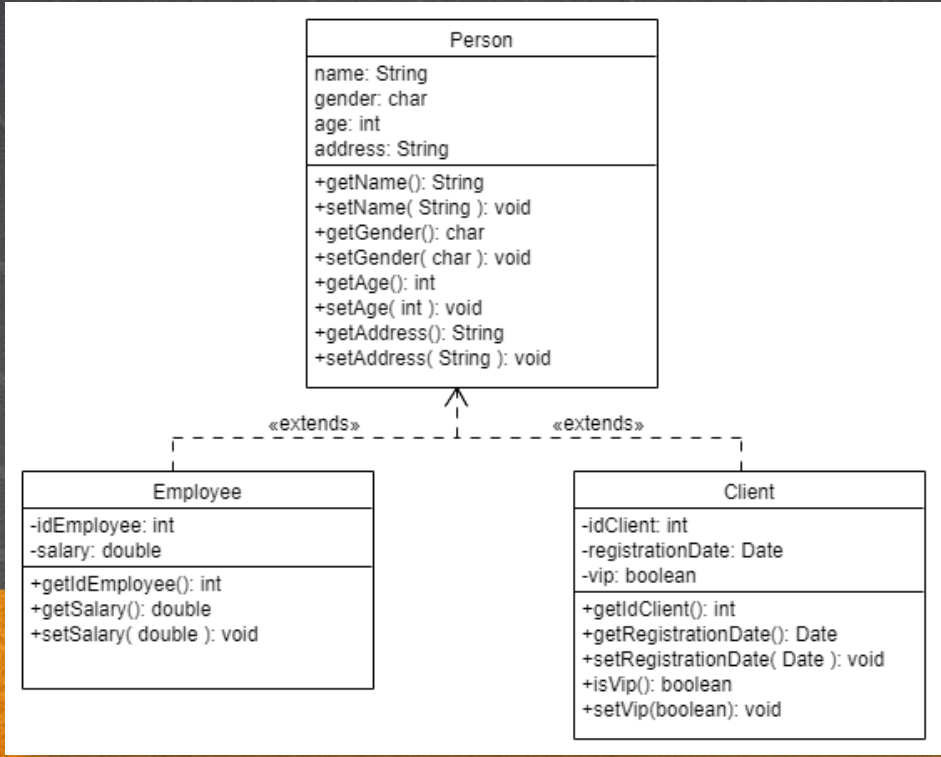


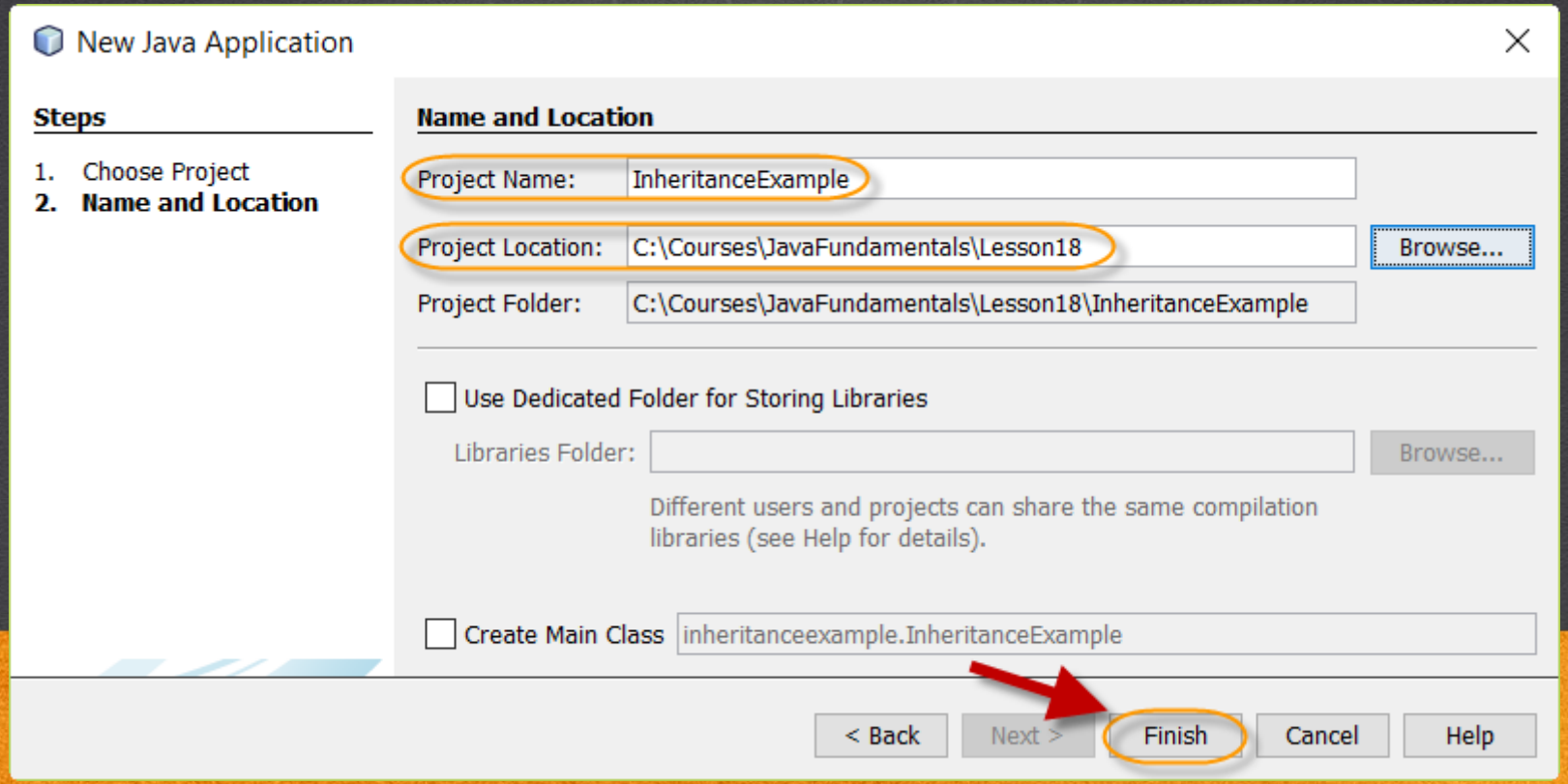
DIAGRAMA DE CLASES

The following is a class diagram of the exercise, created with the tool http://www.umlet.com/umletino/umletino.html:



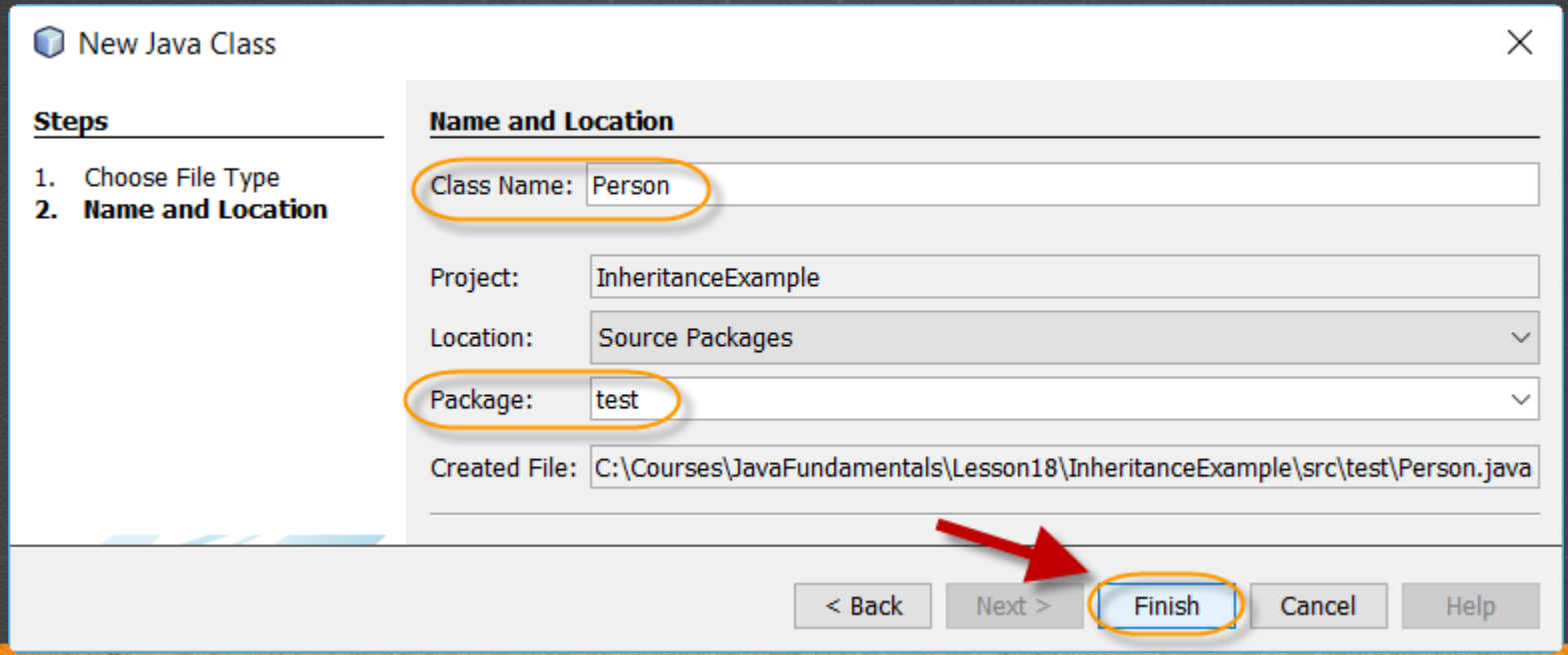
1. CREATE A NEW PROJECT

Create a new project:



2. CREATE A NEW CLASS

Create a new class:



3. MODIFY THE CODE

Person.java:

```
package test;

public class Person { //Definition of the parent class

    private String name;
    private char gender;//M-Male, F-Female
    private int age;
    private String address;

    //Empty constructor
    public Person() {
    }

    //1 argument Constructor
    public Person(String name) {
        this.name = name;
    }

    //Full Constructor
    public Person(String name, char gender, int age, String address) {
        this.name = name;
        this.gender = gender;
        this.age = age;
        this.address = address;
    }
```

3. MODIFY THE CODE

Person.java:

```
public String getName() {
    return name;
}

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

public char getGender() {
    return gender;
}

public void setGender(char gender) {
    this.gender = gender;
}

public int getAge() {
    return age;
}

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

public String getAddress() {
    return address;
}
```

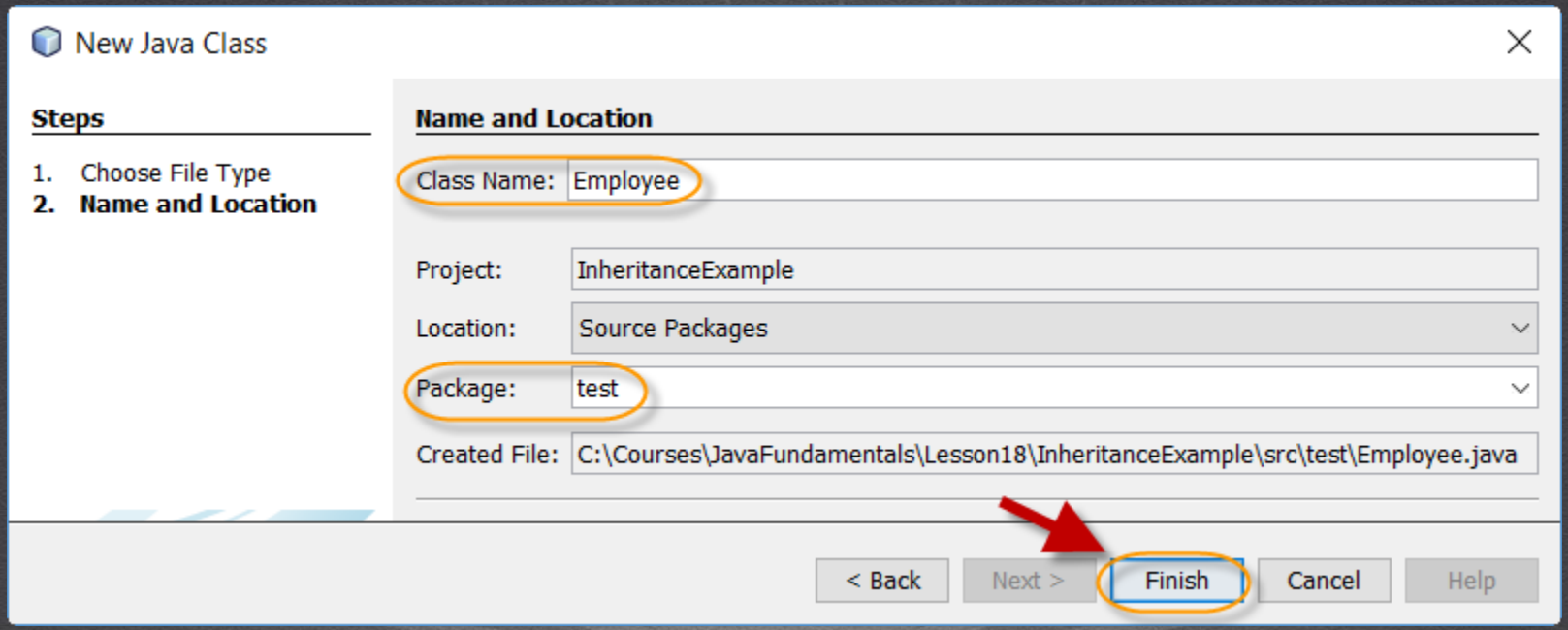
3. MODIFY THE CODE

Person.java:

```
    public void setAddress(String address) {
        this.address = address;
    }

    @Override
    public String toString() {
        return "Person{" + "name=" + name + ", gender=" + gender + ", age=" + age + ", address=" + address + '}';
    }
}
```

4. CREATE THE EMPLOYEE CLASS

Create a new class:



5. MODIFY THE CODE

Employee.java:

```
package test;

public class Employee extends Person {

    private int idEmployee;
    private double salary;
    private static int employeeCounter;

    public Employee(String name, double salary) {
        super(name); //super must be the first line of the constructor
        this.idEmployee = ++employeeCounter;//first increase then assign
        this.salary = salary;
    }

    public int getIdEmployee() {
        return idEmployee;
    }
```

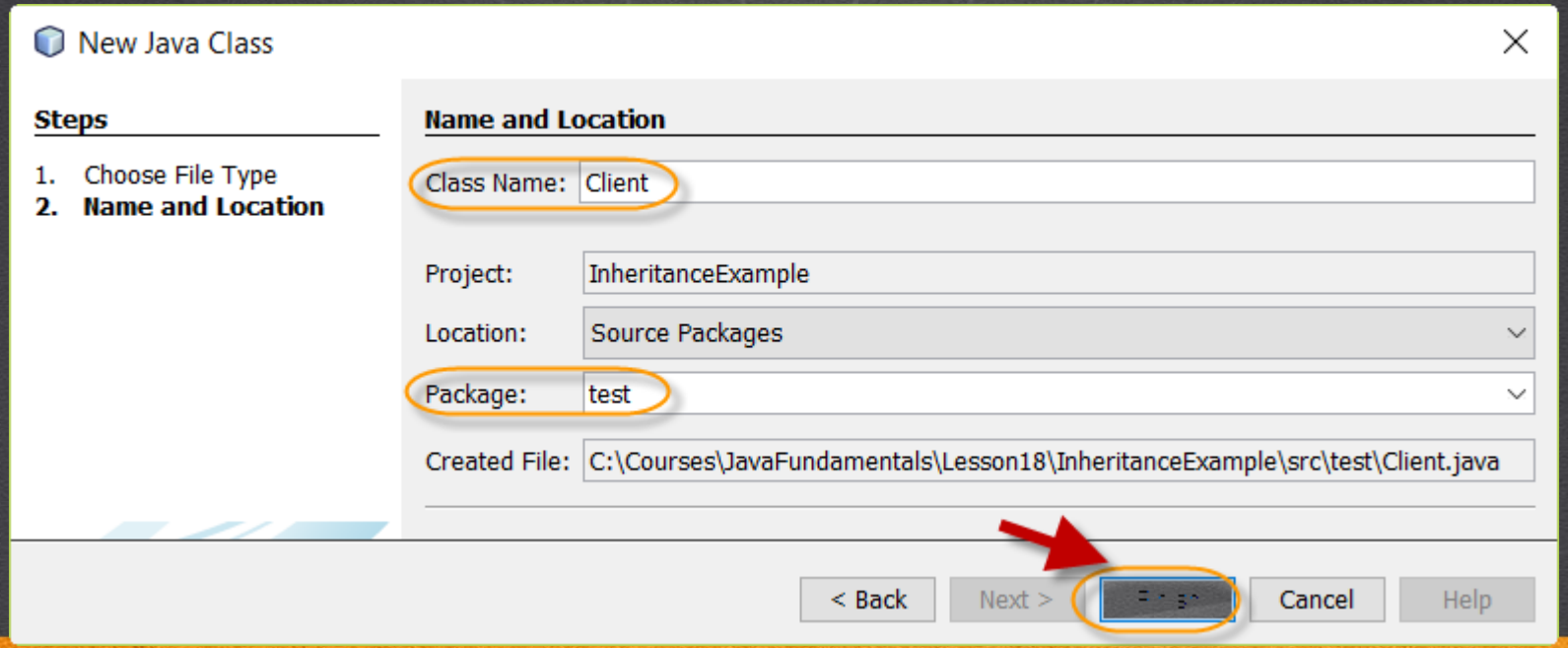
5. MODIFY THE CODE

Employee.java:

```
    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    @Override
    public String toString() {
        //first we print the parent attributes of the parent class
        //after that we print the child attributes of the child class
        return super.toString() + " Empleado{" + "idEmpleado=" + idEmployee + ", sueldo=" + salary + '}';
    }
}
```

6. CREATE THE CLIENT CLASS

Create a new class:



7. MODIFY THE CODE

Client.java:

```
package test;

import java.util.Date;

public class Client extends Person {

    private int idClient;
    private Date registrationDate;
    private boolean vip;
    private static int clientCounter;

    public Client(Date registrationDate, boolean vip) {
        this.idClient = ++clientCounter;
        this.registrationDate = registrationDate;
        this.vip = vip;
    }

    public int getIdClient() {
        return idClient;
    }
```

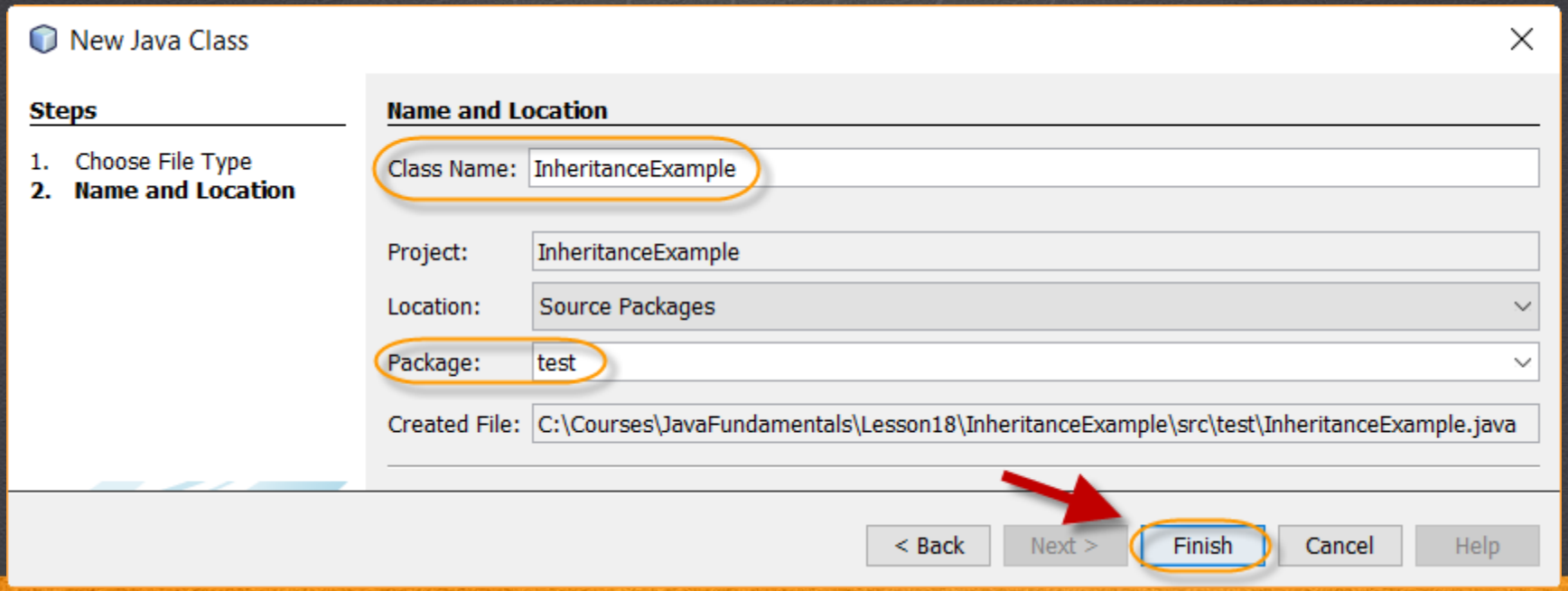
7. MODIFY THE CODE

Client.java:

```
    public Date getRegistrationDate() {
        return registrationDate;
    }

    public void setRegistrationDate(Date registrationDate) {
        this.registrationDate = registrationDate;
    }

    public boolean isVip() {
        return vip;
    }

    public void setVip(boolean vip) {
        this.vip = vip;
    }

    @Override
    public String toString() {
        //first we print the parent attributes of the parent class
        //after that we print the child attributes of the child class
        return super.toString() + " Client{" + "idClient=" + idClient + ", registrationDate=" + registrationDate + ",
vip=" + vip + '}';
    }
}
```

8. CREATE A NEW CLASS

Create a new class:



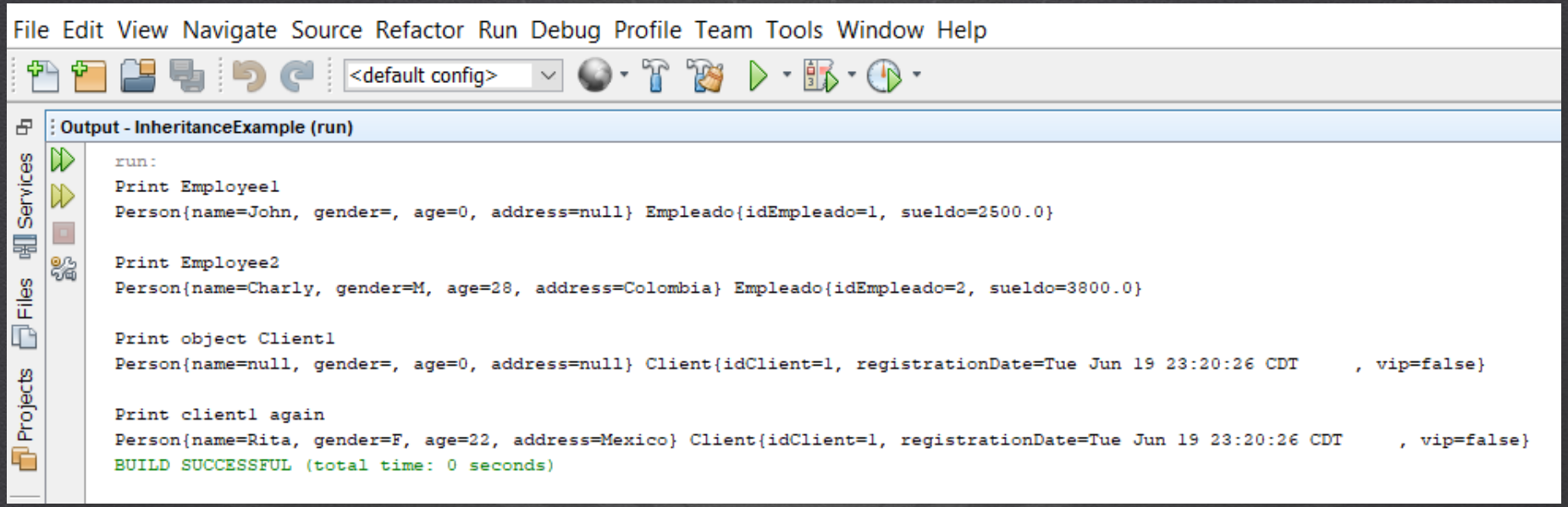
PASO 9. MODIFY THE CODE

InheritanceExample.java:

```
package test;

public class InheritanceExample {

    public static void main(String[] args) {
        Employee e1 = new Employee("John", 2500);
        System.out.println("Print Employee1");
        System.out.println(e1);

        Employee e2 = new Employee("Charly", 3800);
        e2.setAge(28);
        e2.setGender('M');
        e2.setAddress("Colombia");
        System.out.println("\nPrint Employee2");
        System.out.println(e2);

        //When we create a new object of java.util.Date we return the current date
        Client c1 = new Client(new java.util.Date(), false);
        System.out.println("\nPrint object Client1");
        System.out.println(c1);
        //We complete the person object associated with this client object
        c1.setName("Rita");
        c1.setAge(22);
        c1.setGender('F');
        c1.setAddress("Mexico");
        System.out.println("\nPrint client1 again");
        System.out.println(c1);
    }
}
```

10. EXECUTE THE PROJECT

The result is as follows:



EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of Inheritance and several other points.

- We have observed that as the course progresses the code we write becomes more and more interesting. As we move forward we will integrate this theme, which will be the most recurrent when working with Java, either directly or indirectly, but the issue of inheritance is present all the time.

ONLINE COURSE

JAVA FUNDAMENTALS

By: Eng. Ubaldo Acosta