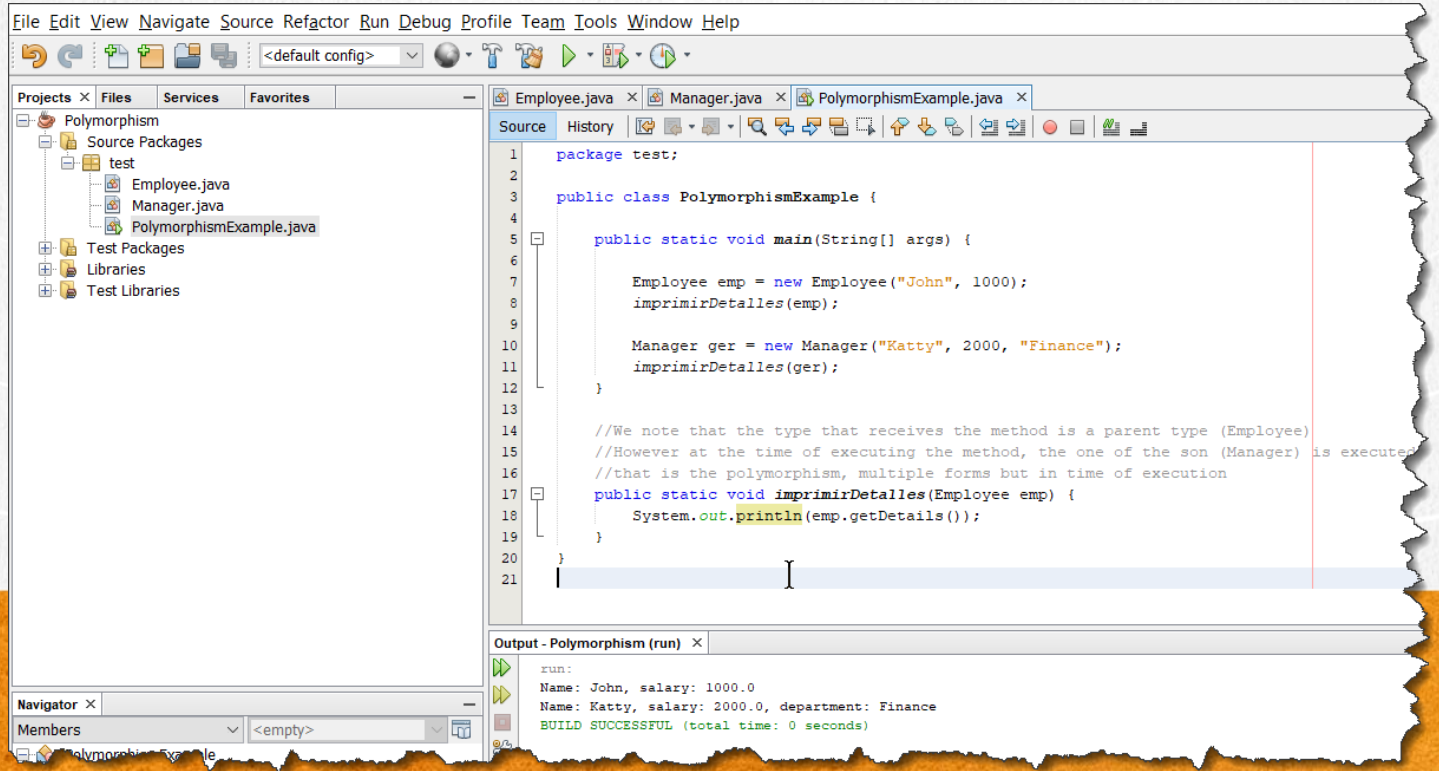
JAVA PROGRAMMING COURSE

EXERCISE

POLYMORPHISM IN JAVA

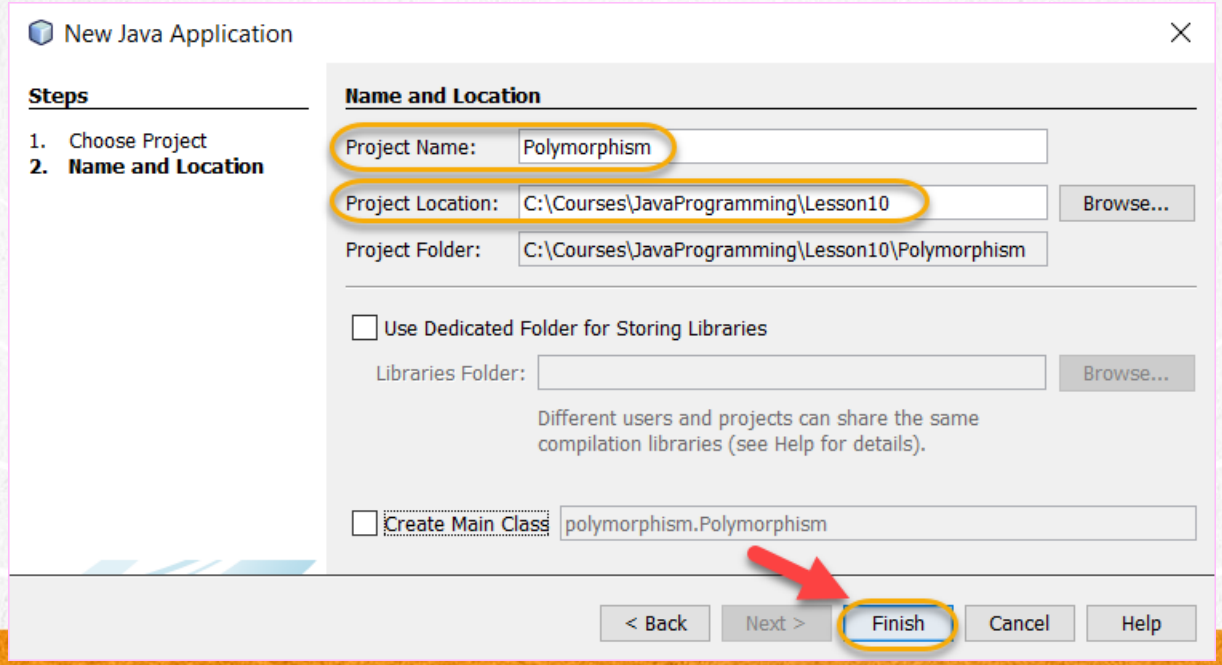
EXERCISE OBJECTIVE

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



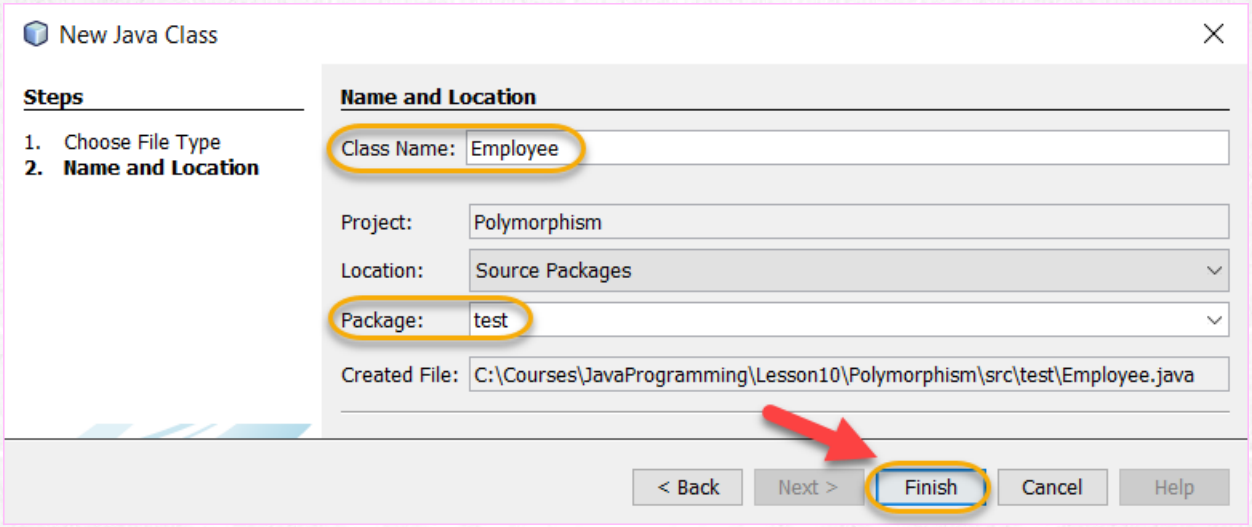
1. CREATE A NEW PROJECT

Create a new project:



2. CREATE A NEW CLASS

Create a new class:



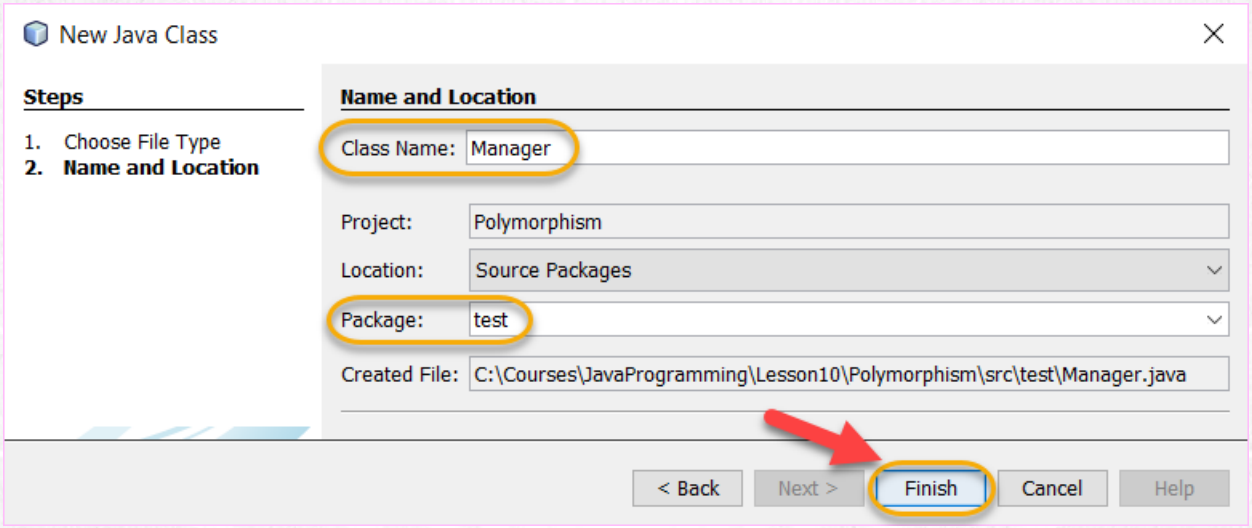
3. MODIFY THE CODE

Employee.java:

```java
package test;

public class Employee {

    protected String name;
    protected double salary;

    protected Employee(String name, double salary){
        this.name = name;
        this.salary = salary;
    }

    public String getDetails(){
        return "Name: " + name + ", salary: " + salary;
    }

      public String getName() {
        return name;
    }

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

    public double getSalary() {
        return salary;
    }

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

4. CREATE A NEW CLASS

Create a new class:



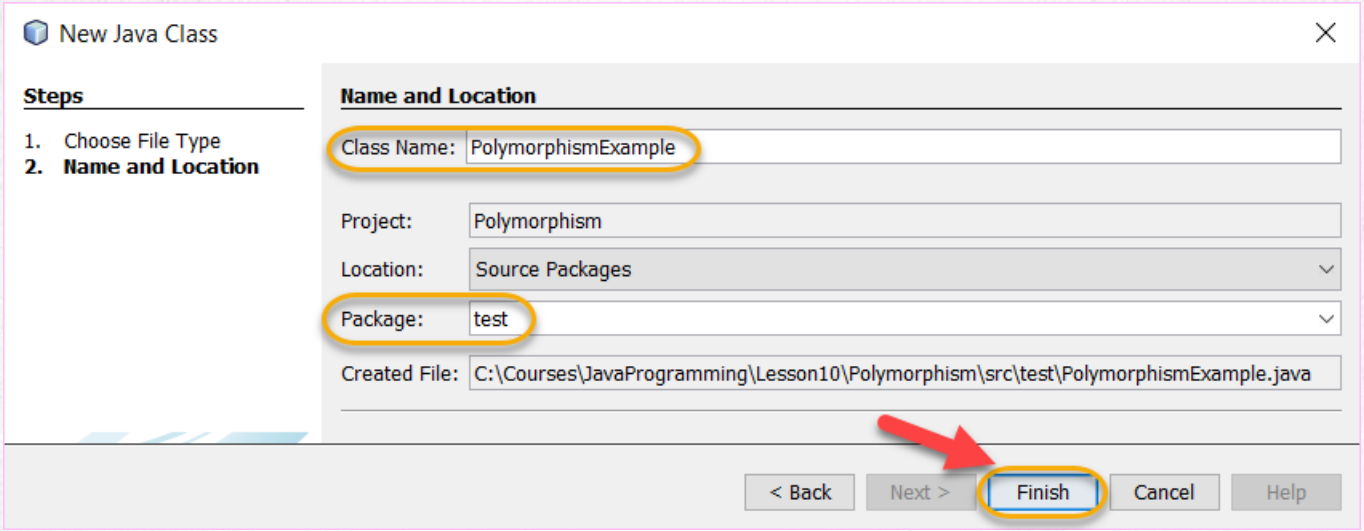
5. MODIFY THE CODE

Manager.java:

```
package test;

public class Manager extends Employee {

    private String department;

    public Manager(String name, double salary, String department) {
        super(name, salary);
        this.department = department;
    }

    //Override the inherited parent method, the @Override annotation is optional
    public String getDetails() {
        //In order not to repeat the code, we can use parent method
        //and concatenate the child attribute to complete the information
        return super.getDetails() + ", department: " + department;
    }

    public String getDepartment() {
        return department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }
}
```

6. CREATE A NEW CLASS

Create a new class:

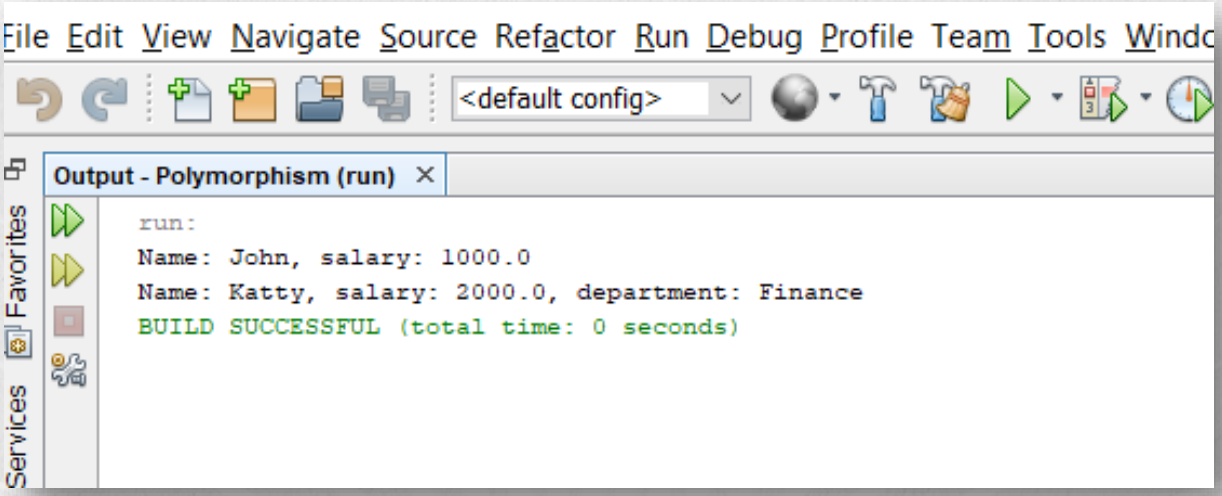


7. MODIFY THE CODE

PolymorphismExample.java:

```
package test;

public class PolymorphismExample {

    public static void main(String[] args) {

        Employee emp = new Employee("John", 1000);
        printDetails(emp);

        Manager ger = new Manager("Katty", 2000, "Finance");
        printDetails(ger);
    }

    //We note that the type that receives the method is a parent type (Employee)
    //However at the time of executing the method, the one of the son (Manager) is executed
    //that is the polymorphism, multiple forms but in time of execution
    public static void printDetails(Employee emp) {
        System.out.println(emp.getDetails());
    }
}
```

8. EXECUTE THE CODE



EXERCISE CONCLUSION

We have put into practice the concept of polymorphism, as well as the overriding of methods and the call to hidden methods due to overriding using the super keyword.

We saw that to apply the concept of polymorphism it is necessary to rely on a hierarchy of classes, and to use a type of data that can store object references in the desired class hierarchy.

ONLINE COURSE

JAVA PROGRAMMING

By: Eng. Ubaldo Acosta