

HIBERNATE & JPA COURSE

EXERCISE

LIFECYCLE IN HIBERNATE/JPA



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

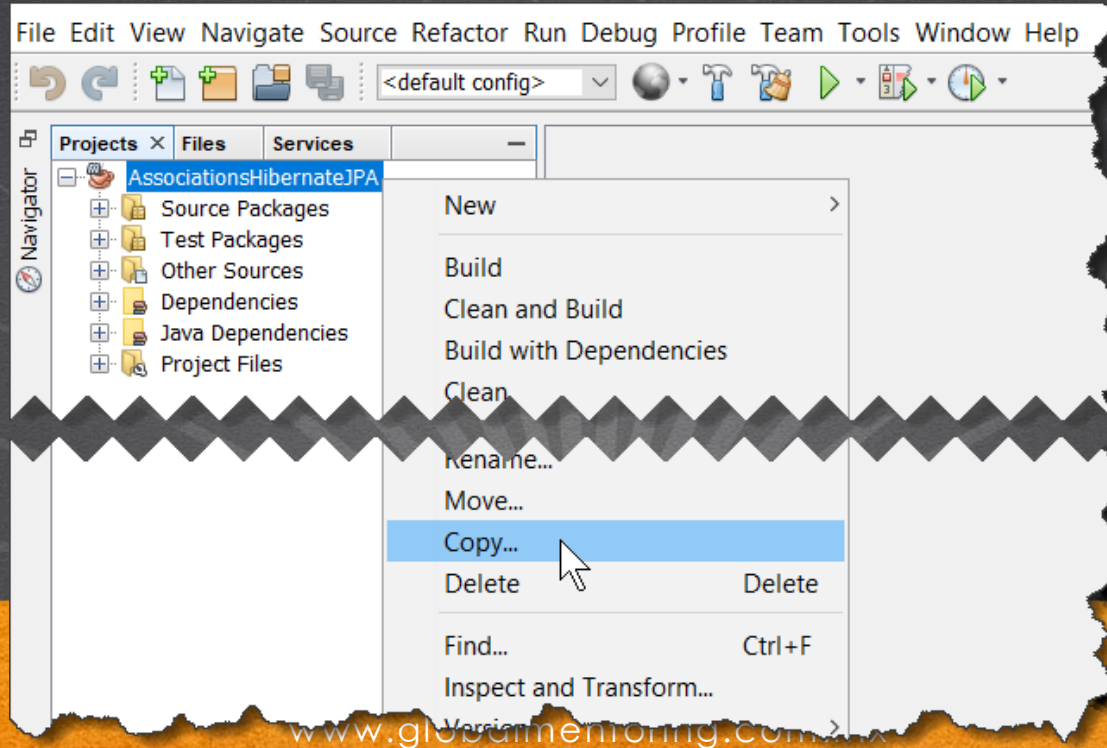
EXERCISE OBJECTIVE

Implement the Life Cycle of entity objects in Hibernate / JPA.
At the end we will see:

```
Output X
Run (State1Persistent) X Retriever Output X
...
20:27:43 [main] INFO org.hibernate.Version - HHH000412: Hibernate Core (5.3.6.Final)
20:27:43 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
20:27:43 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations (5.0.4.Final)
20:27:43 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
20:27:43 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sms_db]
20:27:43 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
20:27:43 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
20:27:43 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size: 20 (min=1)
20:27:44 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
20:27:44 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
20:27:44 [main] DEBUG org.hibernate.SQL - insert into address (country, deleted, street_name, street_number, version) values (?, ?, ?, ?, ?)
Hibernate: insert into address (country, deleted, street_name, street_number, version) values (?, ?, ?, ?, ?)
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [USA]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [0]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [VARCHAR] - [Star Avenue]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [51]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [INTEGER] - [0]
New contact:Address{idAddress=2, streetName=Star Avenue, streetNumber=51, country=USA, version=0, deleted=0}
```

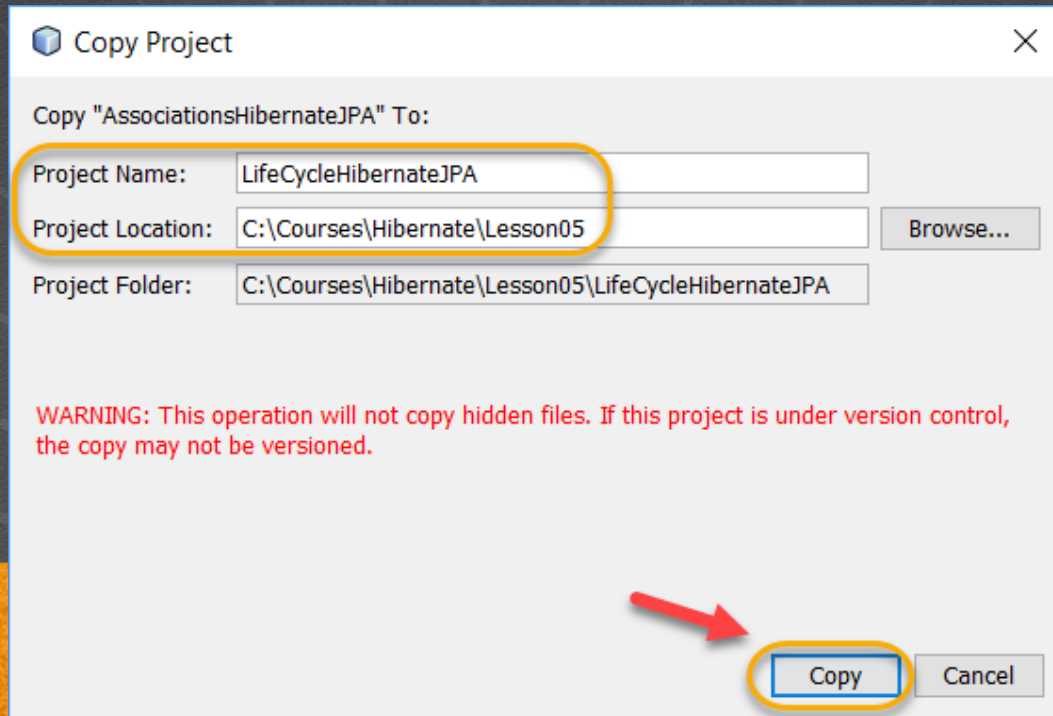
1. CREATE A PROJECT

We copy and paste the previous project
AsociacionesHibernateJPA:



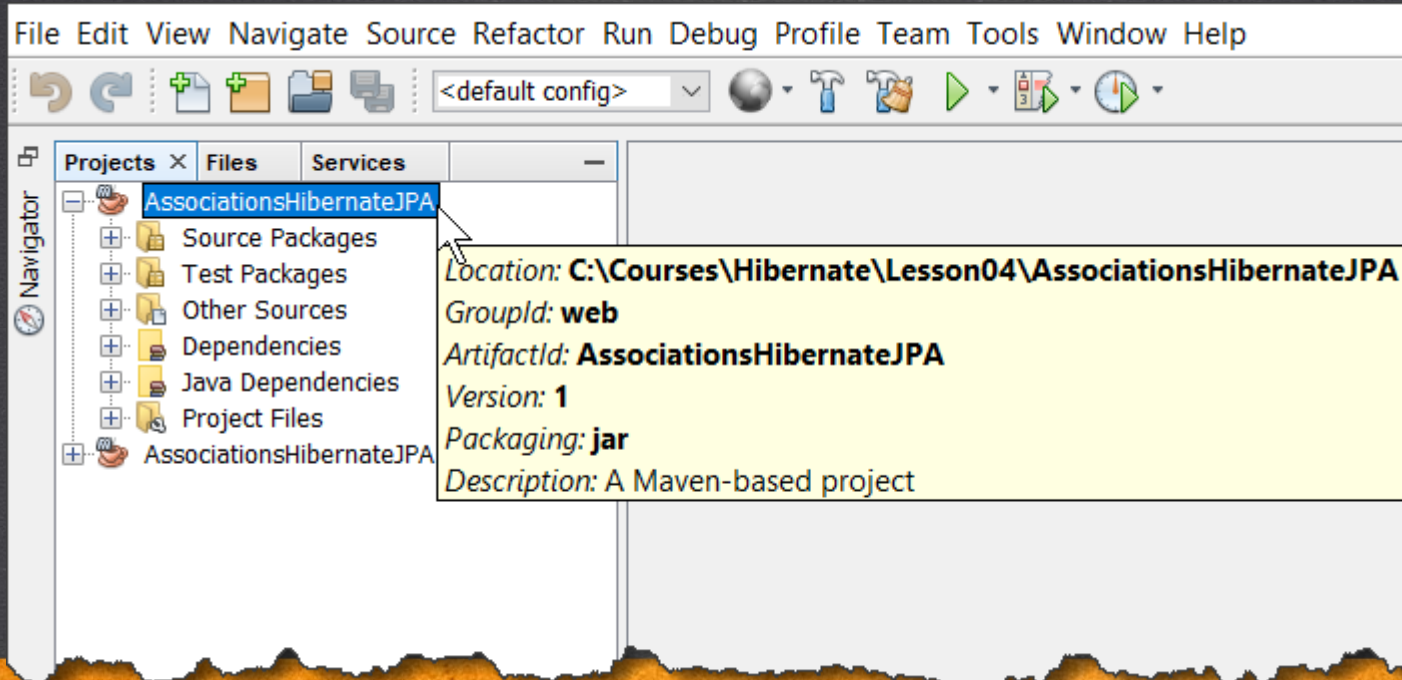
1. CREATE A PROJECT

We copy and paste the previous project
AsociacionesHibernateJPA:



2. CLOSE THE PREVIOUS PROJECT

We review the project we want to close:

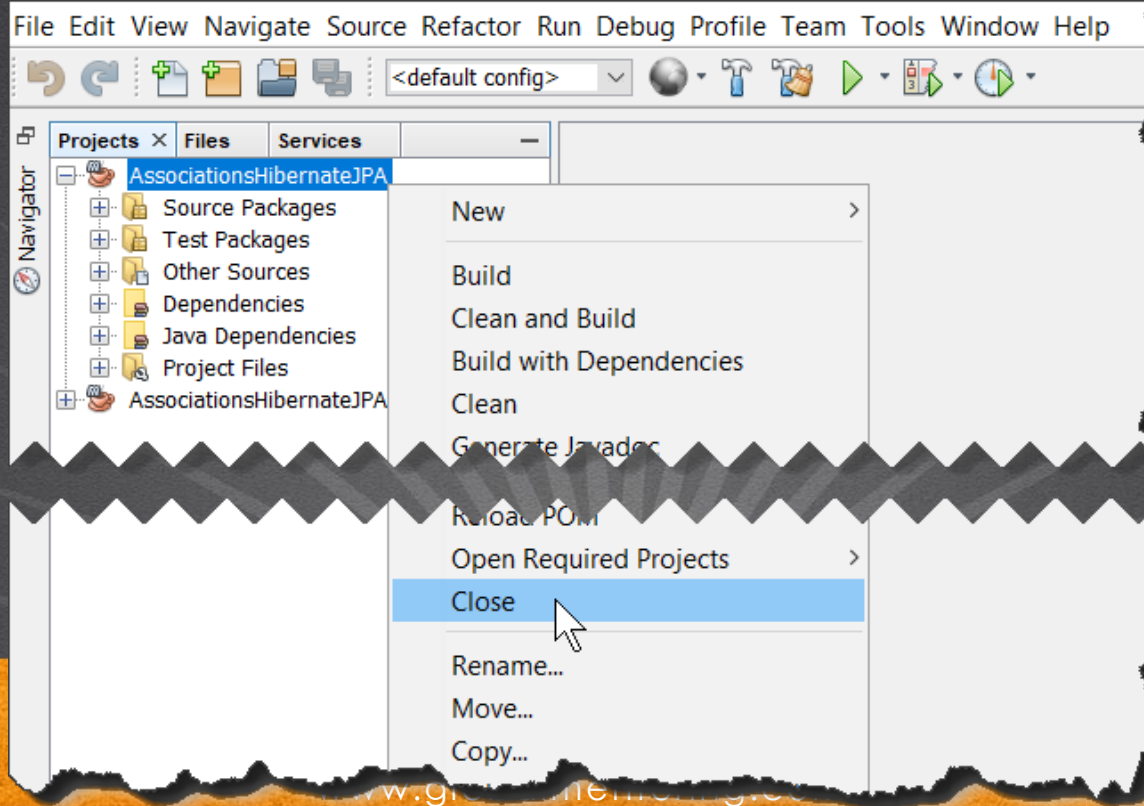


HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

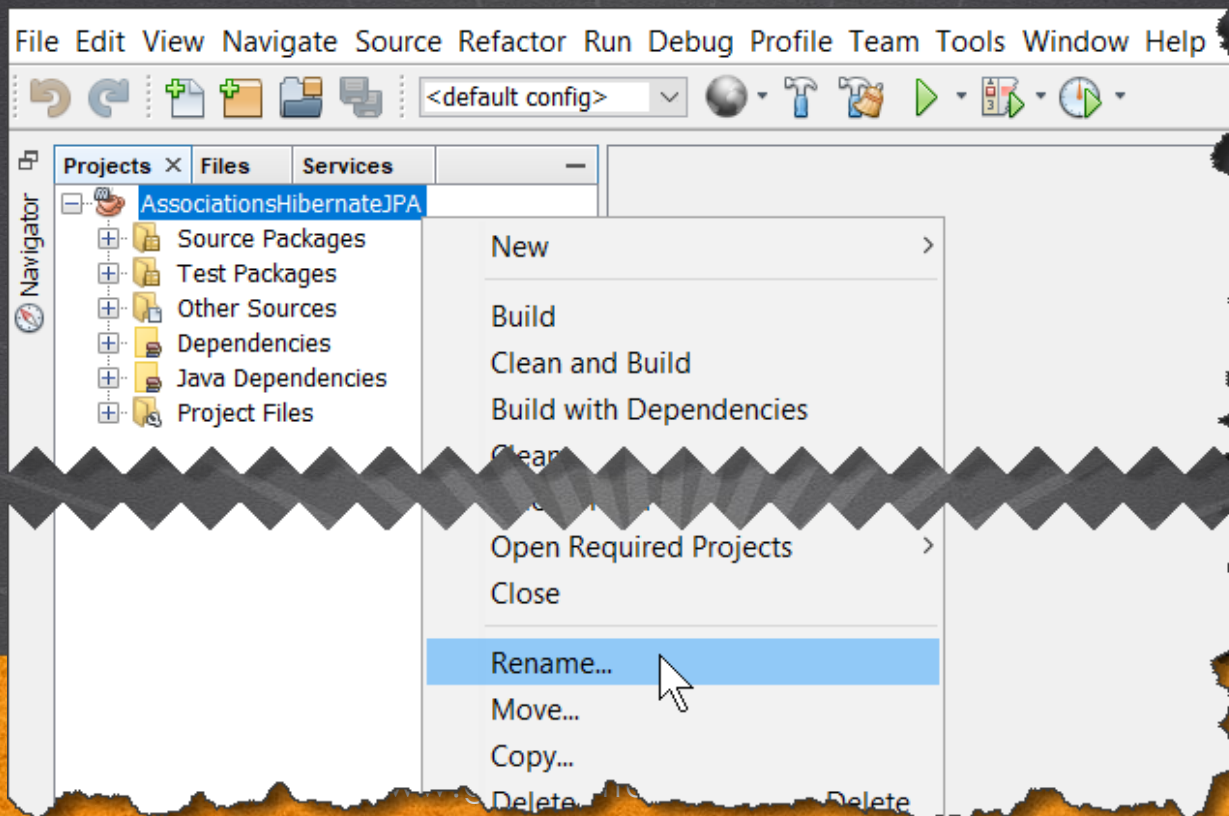
2. CLOSE THE PREVIOUS PROJECT

We close the project that we no longer use:



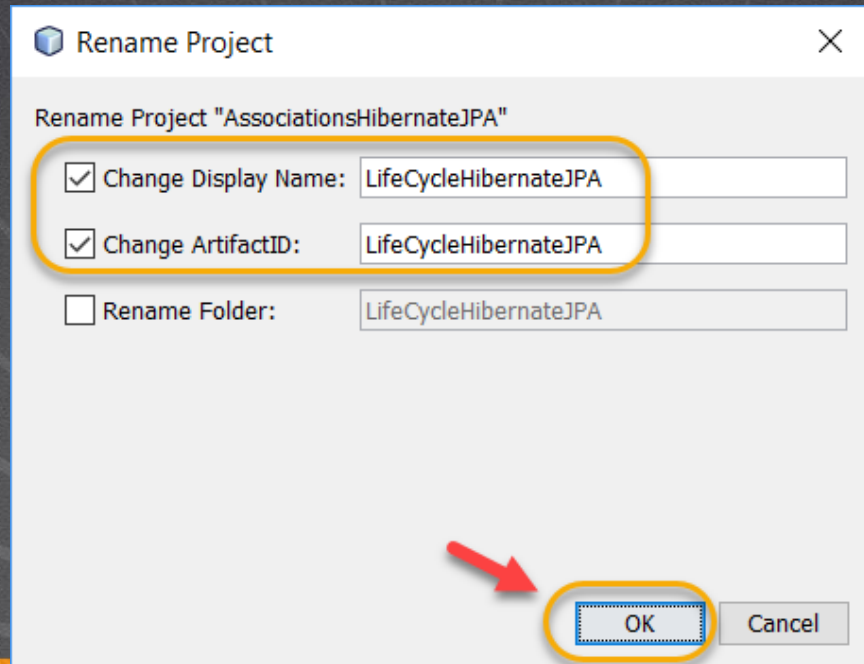
3. RENAME THE PROJECT

We rename the project:



3. RENAME THE PROJECT

We rename the project:

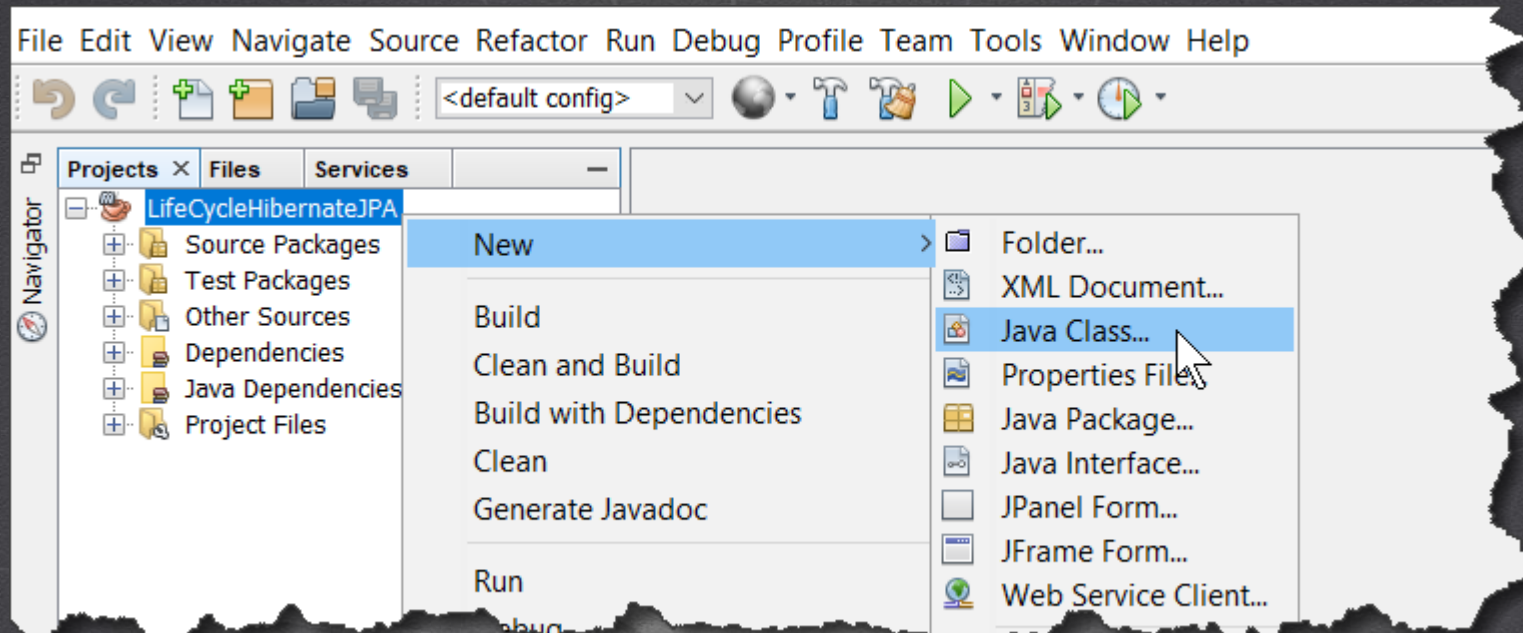


HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

4. CREATE A JAVA CLASS

We create a Java class:

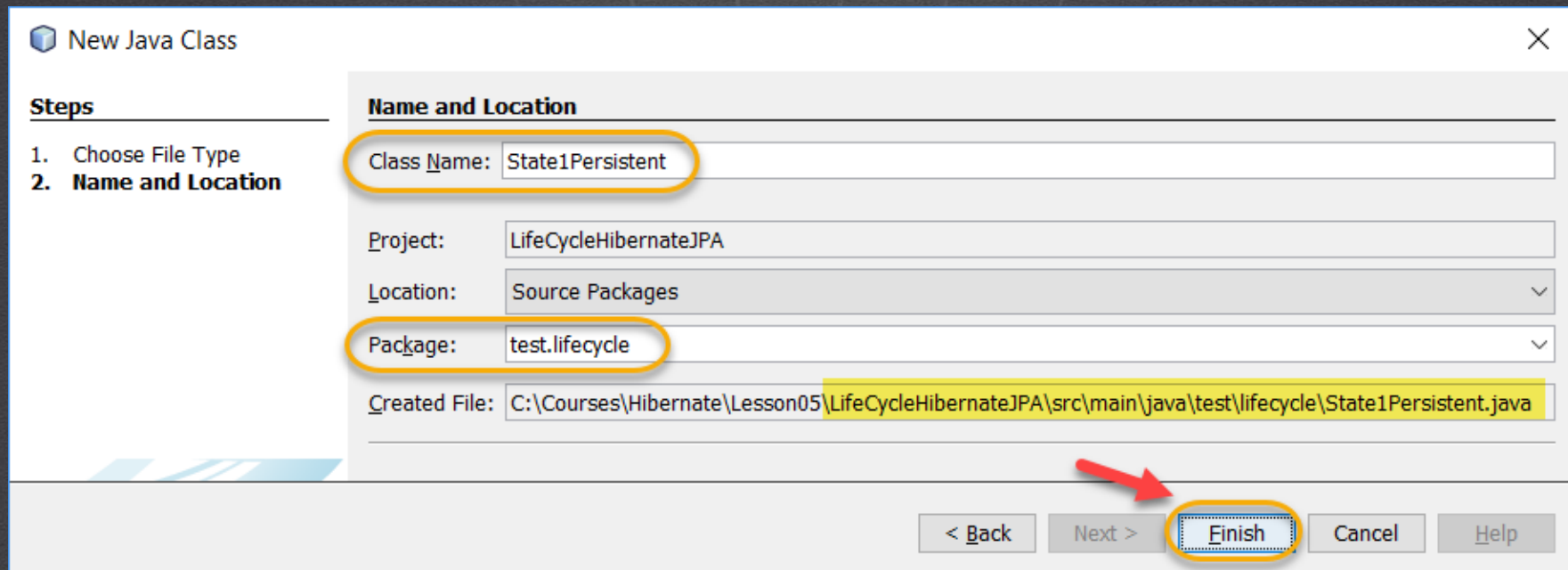


HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

4. CREATE A JAVA CLASS

We create a Java class:



New Java Class

Steps

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

Name and Location

Class Name: State1Persistent

Project: LifeCycleHibernateJPA

Location: Source Packages

Package: test.lifecycle

Created File: C:\Courses\Hibernate\Lesson05\LifeCycleHibernateJPA\src\main\java\test\lifecycle\State1Persistent.java

< Back Next > **Finish** Cancel Help

HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

5. MODIFY THE CODE

State1Persistent.java:

Click to download

```
package test.lifecycle;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import model.Address;

public class State1Persistent {

    public static void main(String[] args) {
        /*We use the JPA Persistence Unit*/
        EntityManagerFactory factory = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = factory.createEntityManager();
        /**
         * Objective: Transient to persistent state
         */

        //1. We create the object (new or transitive state)
        Address address = new Address();
        address.setStreetName("Star Avenue");
        address.setStreetNumber("51");
        address.setCountry("USA");
    }
}
```


5. MODIFY THE CODE

State1Persistent.java:

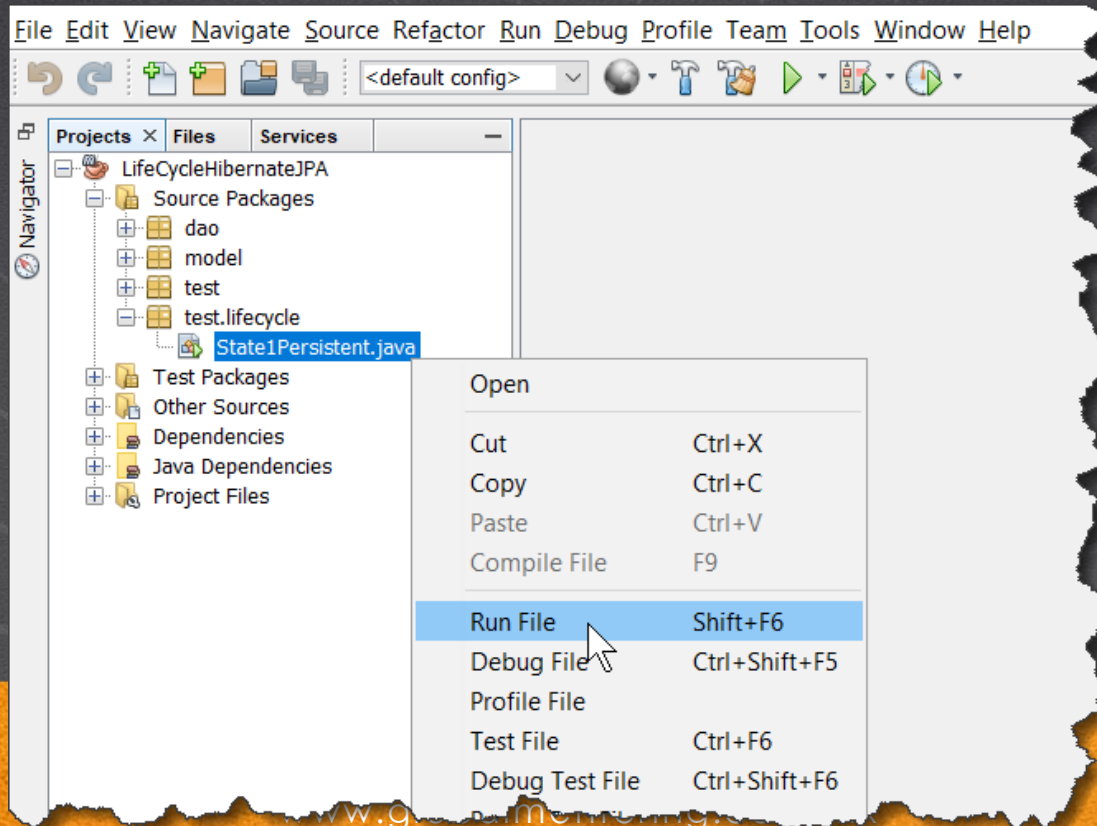
Click to download

```
//2. We persist the object (it changes state to persistent)
//We start the transaction
//We will not use the DAO classes in these exercises
try {
    // We start a transaction
    em.getTransaction().begin();
    // Insert the new address
    em.persist(address);
    // We finish the transaction
    em.getTransaction().commit();
} catch (Exception ex) {
    System.out.println("Error inserting object:" + ex.getMessage());
} finally {
    if (em != null) {
        em.close();
    }
}

//The address object changes from persistent state to detached
//at the close of the session
System.out.println("New contact:" + address);
}
```

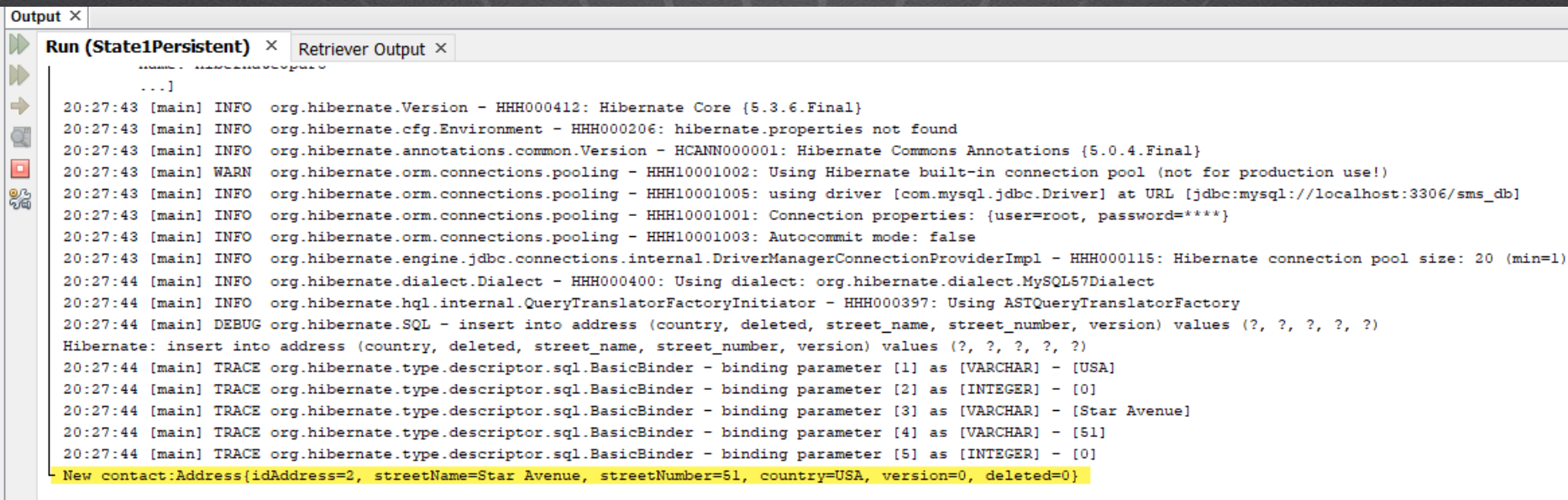
6. EXECUTE THE CLASS

Execute the class:



6. EXECUTE THE CODE

We observe the result of inserting a new Address type object. The value of idAddress may vary depending on the records that are in the database in the address table:

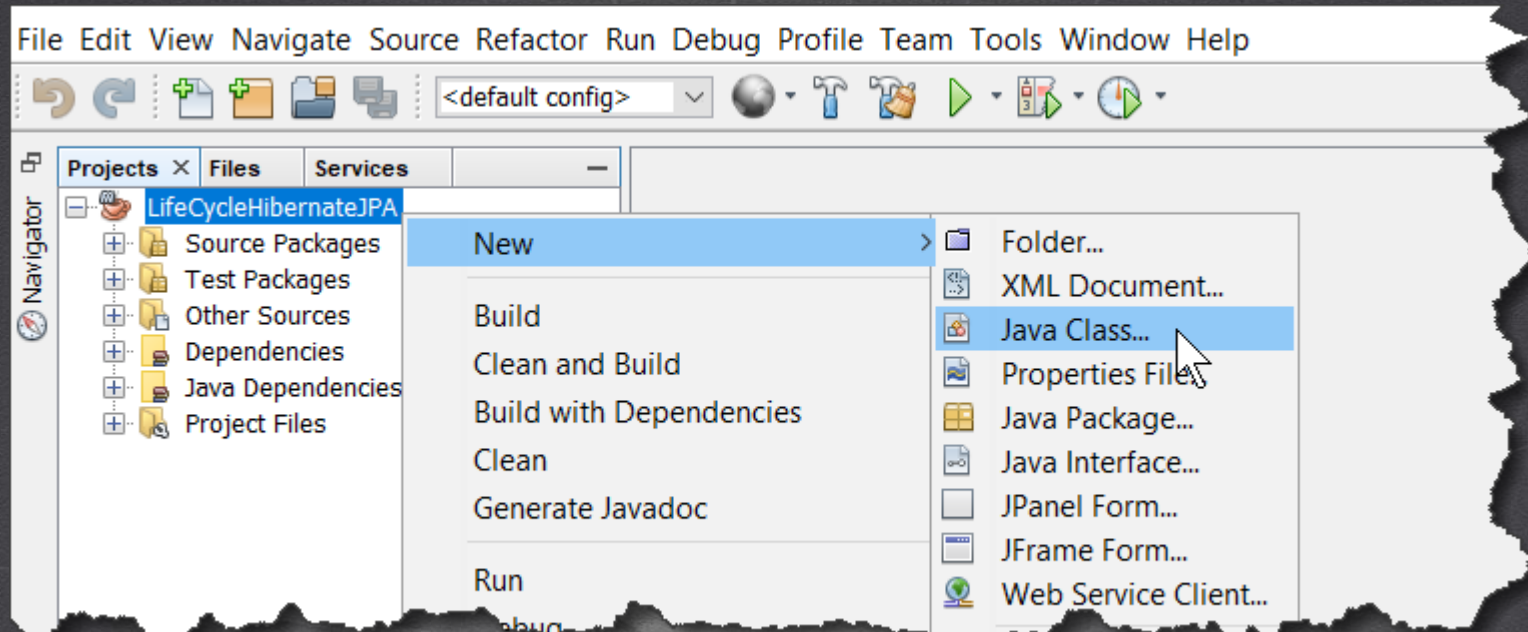


The screenshot shows an IDE's Output window with the following content:

```
Output X
Run (State1Persistent) X Retriever Output X
...
20:27:43 [main] INFO org.hibernate.Version - HHH000412: Hibernate Core (5.3.6.Final)
20:27:43 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
20:27:43 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations (5.0.4.Final)
20:27:43 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
20:27:43 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sms_db]
20:27:43 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
20:27:43 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
20:27:43 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size: 20 (min=1)
20:27:44 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
20:27:44 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
20:27:44 [main] DEBUG org.hibernate.SQL - insert into address (country, deleted, street_name, street_number, version) values (?, ?, ?, ?, ?)
Hibernate: insert into address (country, deleted, street_name, street_number, version) values (?, ?, ?, ?, ?)
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [USA]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [0]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [VARCHAR] - [Star Avenue]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [51]
20:27:44 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [INTEGER] - [0]
New contact:Address{idAddress=2, streetName=Star Avenue, streetNumber=51, country=USA, version=0, deleted=0}
```


7. CREATE A NEW CLASS

Create a new Java class:



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

7. CREATE A JAVA CLASS

Create a Java class:

New Java Class

Steps

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

Name and Location

Class Name: State2RetrivePersistentObject

Project: LifeCycleHibernateJPA

Location: Source Packages

Package: test.lifecycle

Created File: C:\Courses\Hibernate\Lesson05\LifeCycleHibernateJPA\src\main\java\test\lifecycle\State2RetrivePersistentObject.java

< Back Next > **Finish** Cancel Help

HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

8. MODIFY THE FILE

[State2RetrivePersistentObject.java:](#)

Click to download

```
package test.lifecycle;

import javax.persistence.*;
import javax.persistence.Persistence;
import model.Address;

public class State2RetrivePersistentObject {

    public static void main(String[] args) {
        EntityManagerFactory factory = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = factory.createEntityManager();
        /**
         * Objective: Recover a persistent object with Hibernate / JPA
         */

        //There are 2 ways to perform this action
        //1. Method get, return null if you can not find the object
        Address address1 = null;

        try {
            em.getTransaction().begin();
            //We provide the ID to recover, in JPA we use find instead of get
            address1 = (Address) em.find(Address.class, 2);
            System.out.println("Address retrieved with find:" + address1);
            em.getTransaction().commit(); //we make flush
        } catch (Exception e) {
            em.getTransaction().rollback();
            e.printStackTrace(System.out);
        }
    }
}
```


8. MODIFY THE FILE

State2RetrivePersistentObject.java:

Click to download

```
//The object of address1 changes to edo. detached at the close of the session

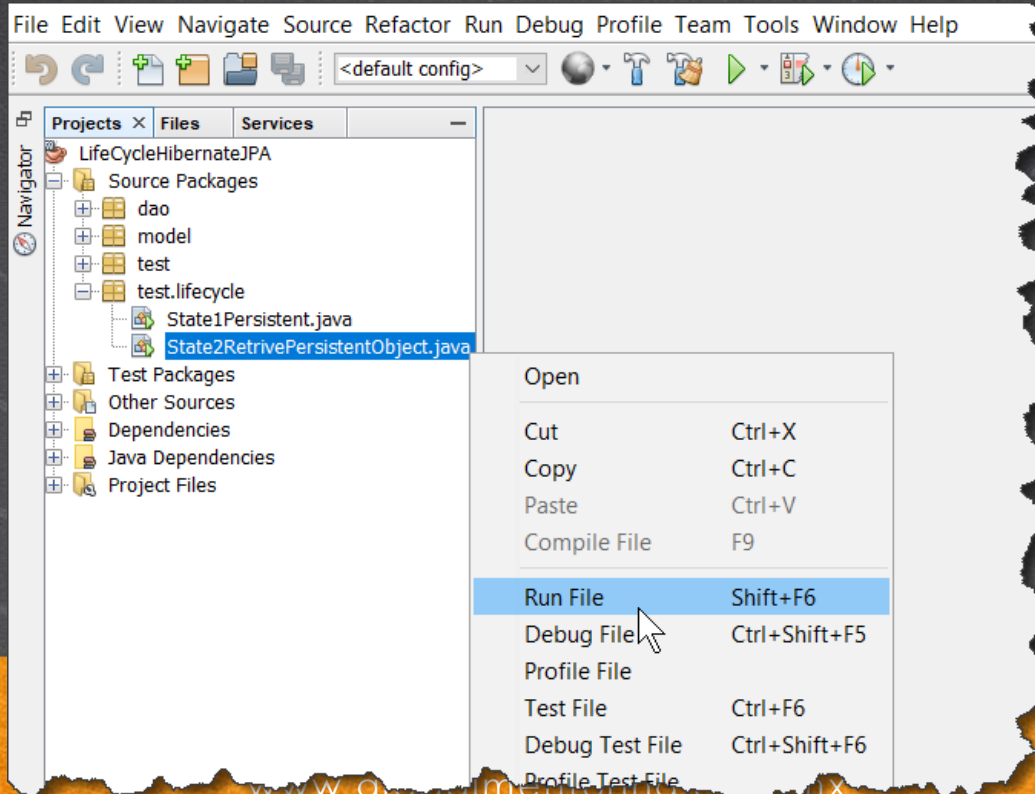
//2. Method load, return ObjectNotFoundException
//if you can not find the id provided
Address address2 = null;
//We create a second transaction
try {
    em.getTransaction().begin();

    //We provide the id to recover, it does not get relationships (lazy)
    //In JPA we have getReference as another option
    address2 = (Address) em.getReference(Address.class, 2);
    System.out.println("Address retrieved with getReference:" + address2);
    em.getTransaction().commit();
} catch (Exception e) {
    em.getTransaction().rollback();
    e.printStackTrace(System.out);
} finally {
    if (em != null) {
        em.close();
    }
}

//The address2 object changes to detached state
//when closing transaction 2
}
```

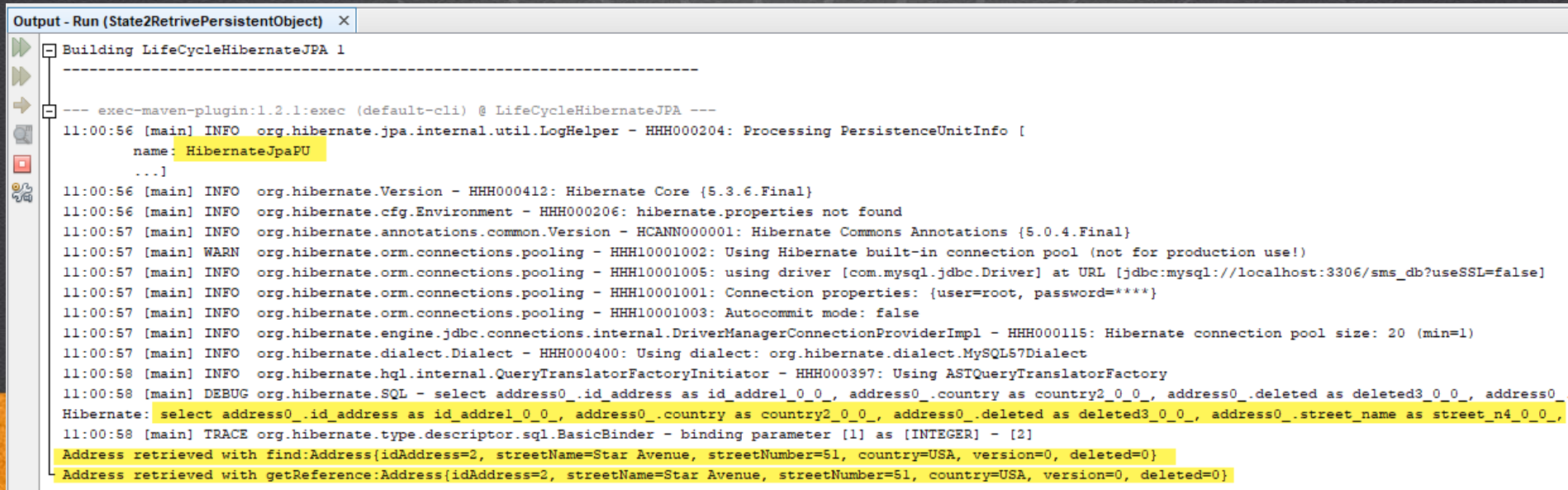
9. EXECUTE THE CODE

Execute the class:



9. EXECUTE THE CODE

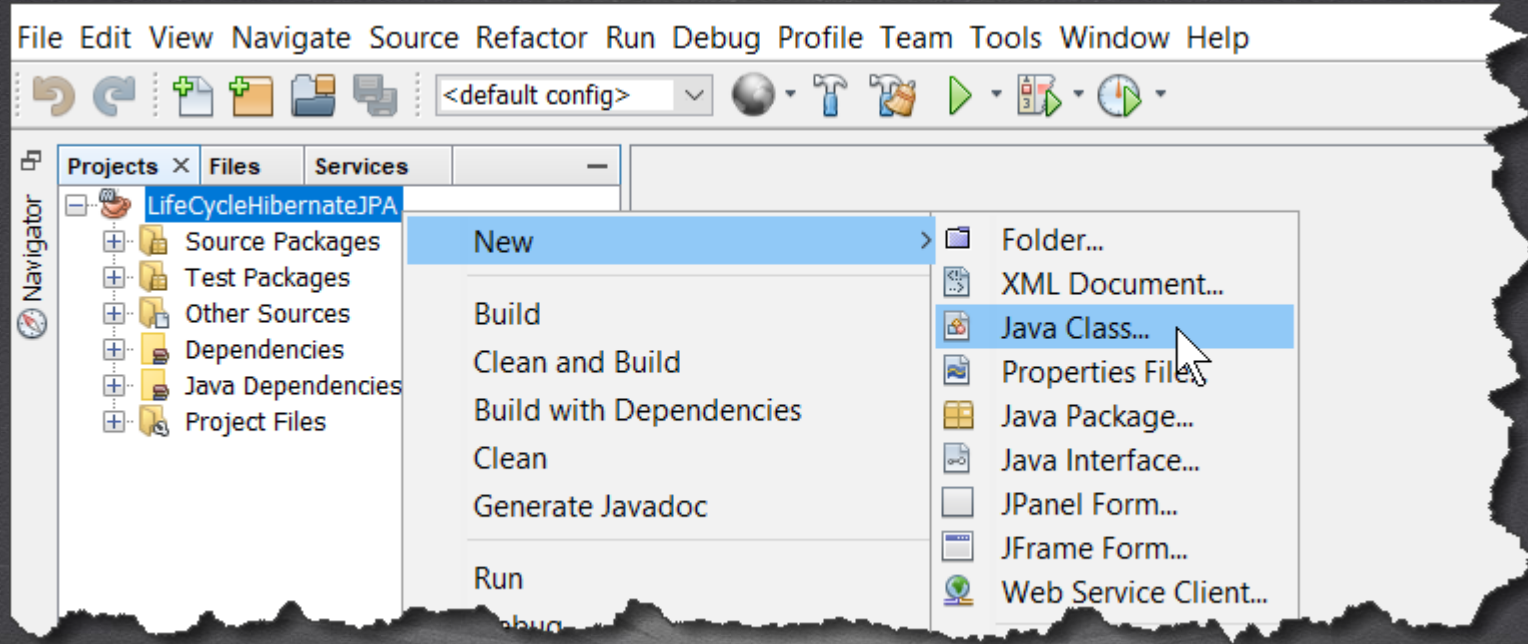
We observe the result of executing our class. The idAddress to search may not exist, so the value must be adapted to an existing value.



```
Building LifeCycleHibernateJPA 1
-----
--- exec-maven-plugin:1.2.1:exec (default-cli) @ LifeCycleHibernateJPA ---
11:00:56 [main] INFO org.hibernate.jpa.internal.util.LogHelper - HHH000204: Processing PersistenceUnitInfo [
    name: HibernateJpaPU
    ...]
11:00:56 [main] INFO org.hibernate.Version - HHH000412: Hibernate Core {5.3.6.Final}
11:00:56 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
11:00:57 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations {5.0.4.Final}
11:00:57 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
11:00:57 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sms_db?useSSL=false]
11:00:57 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
11:00:57 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
11:00:57 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size: 20 (min=1)
11:00:57 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
11:00:58 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
11:00:58 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addrel_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_
Hibernate: select address0_.id_address as id_addrel_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_
11:00:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [2]
Address retrieved with find:Address{idAddress=2, streetName=Star Avenue, streetNumber=51, country=USA, version=0, deleted=0}
Address retrieved with getReference:Address{idAddress=2, streetName=Star Avenue, streetNumber=51, country=USA, version=0, deleted=0}
```


10. CREATE A NEW JAVA CLASS

Create a new class:



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

10. CREATE A JAVA CLASS

Create a Java Class:

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

11. MODIFY THE CODE

[State3ModifyPersistentObject.java:](#)

Click to download

```
package test.lifecycle;

import javax.persistence.*;
import model.Address;

public class State3ModifyPersistentObject {

    public static void main(String[] args) {
        EntityManagerFactory factory = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = factory.createEntityManager();

        /**
         * Objective: Modify a persistent object (reattaching) Move from a
         * detached to persistent state
         */
        //We recover a persistent object
        Address address = null;
        try {
            em.getTransaction().begin();
            //Identifier to recover, you can also use merge to retrieve an object
            address = (Address) em.find(Address.class, 1);
            em.getTransaction().commit(); //hacemos flush
        } catch (Exception e) {
            em.getTransaction().rollback();
            e.printStackTrace(System.out);
        }
    }
}
```


11. MODIFY THE CODE

[State3ModifyPersistentObject.java:](#)

Click to download

```
//We print the values of the recovered object
//state detached
System.out.println("Address recovered:" + address);

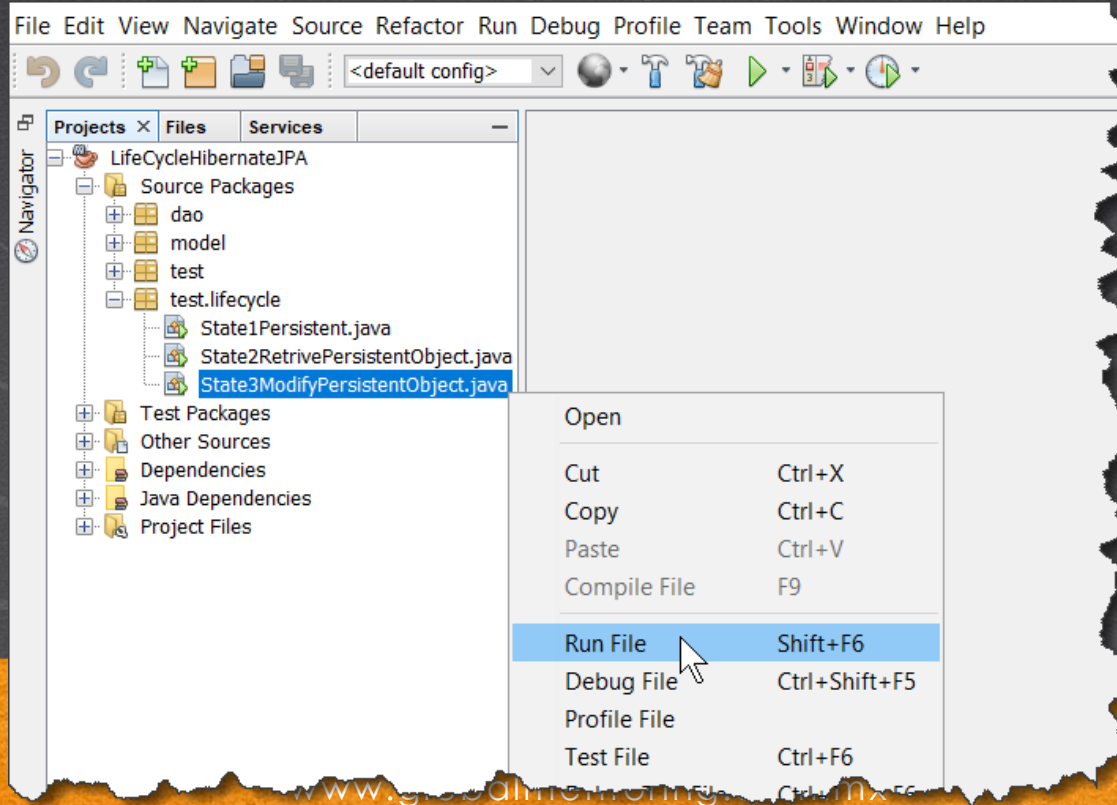
//We modify the object outside of a transaction
if (address != null) {
    address.setStreetName("new street");
}

//We return to save the object in a new transaction
//We change the state from detached to persistent (reattaching)
try {
    em.getTransaction().begin();
    em.merge(address);
    em.getTransaction().commit(); //we make flush
} catch (Exception e) {
    em.getTransaction().rollback();
    e.printStackTrace(System.out);
} finally {
    if (em != null) {
        em.close();
    }
}

System.out.println("Modified address object:" + address);
//The address object changes to detached status when closing session
}
```

12. EXECUTE THE CLASS

Execute the Java class:



12. EXECUTE THE JAVA CLASS

Execute the class:

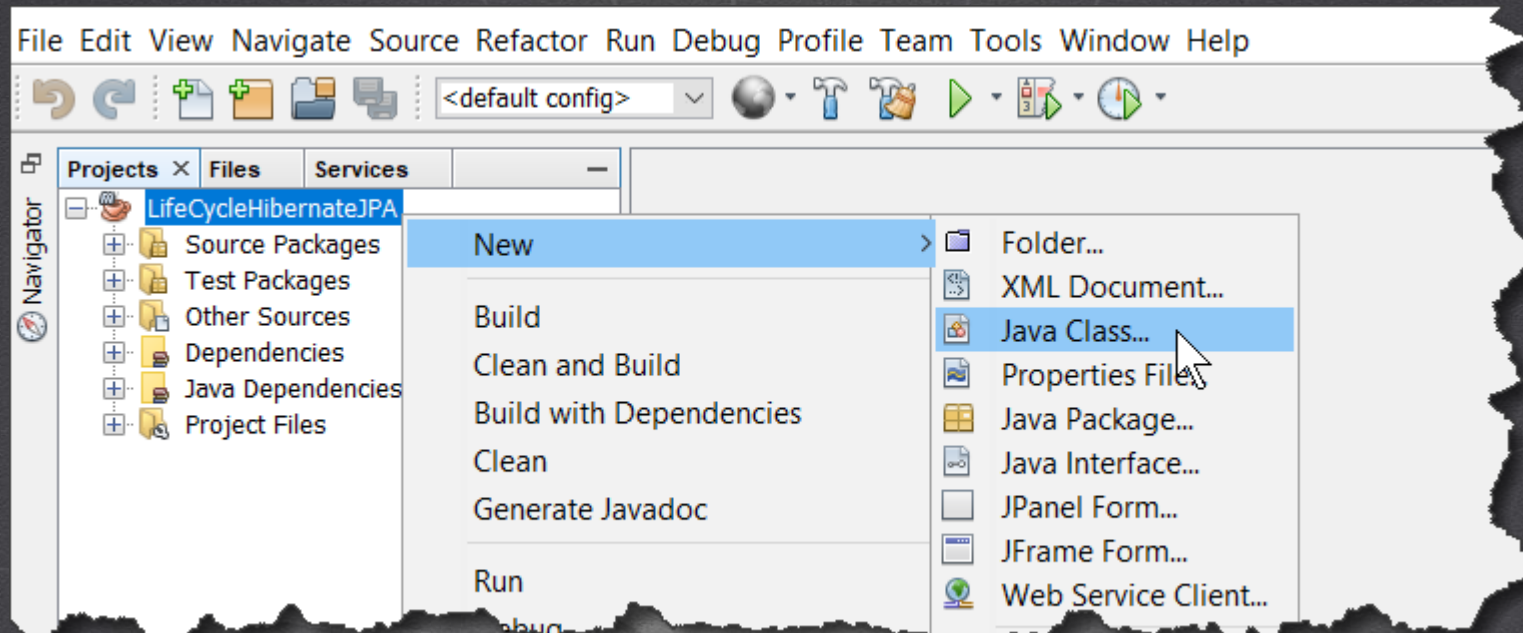
```
Output - Run (State3ModifyPersistentObject) x
11:51:53 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
11:51:53 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations {5.0.4.Final}
11:51:53 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
11:51:53 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sms_db?useSSL=false]
11:51:53 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
11:51:53 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
11:51:53 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size: 20 (min=1)
11:51:53 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
11:51:54 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
11:51:54 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addrel_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_ from address0_ where address0_.id_address=?
Hibernate: select address0_.id_address as id_addrel_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_ from address0_ where address0_.id_address=?
11:51:54 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
Address recovered:Address{idAddress=1, streetName=Estrella, streetNumber=109, country=Mexico, version=0, deleted=0}
11:51:54 [main] DEBUG org.hibernate.SQL - update address set country=?, deleted=?, street_name=?, street_number=?, version=? where id_address=?
Hibernate: update address set country=?, deleted=?, street_name=?, street_number=?, version=? where id_address=?
11:51:54 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [Mexico]
11:51:54 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [0]
11:51:54 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [VARCHAR] - [new street]
11:51:54 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [109]
11:51:54 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [INTEGER] - [0]
11:51:54 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [INTEGER] - [1]
Modified address object:Address{idAddress=1, streetName=new street, streetNumber=109, country=Mexico, version=0, deleted=0}
```

HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

13. CREATE A NEW JAVA CLASS

Create a new Java class:



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

13. CREATE A JAVA CLASS

Create a Java class:

New Java Class

Steps

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

Name and Location

Class Name: State4ModifyObjectLongSession

Project: LifeCycleHibernateJPA

Location: Source Packages

Package: test.lifecycle

Created File: C:\Courses\Hibernate\Lesson05\LifeCycleHibernateJPA\src\main\java\test\lifecycle\State4ModifyObjectLongSession.java

< Back Next > **Finish** Cancel Help

14. MODIFY THE CODE

[State4ModifyObjectLongSession.java:](#)

Click to download

```
package test.lifecycle;

import javax.persistence.*;
import model.Address;

public class State4ModifyObjectLongSession {

    public static void main(String[] args) {
        /*We use the JPA Persistence Unit */
        EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = fabrica.createEntityManager();

        //We recover a persistent object
        Address address = null;
```


14. MODIFY THE CODE

[State4ModifyObjectLongSession.java:](#)

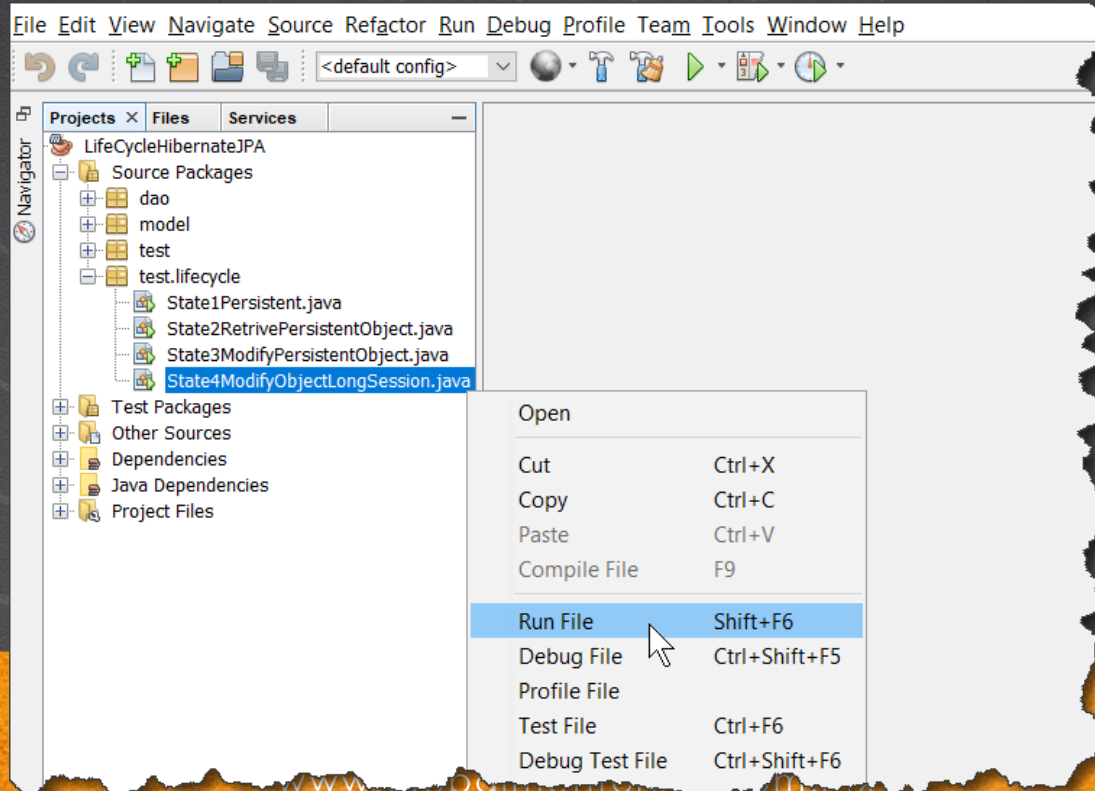
Click to download

```
try {
    em.getTransaction().begin();
    //Identifier to recover, you can also use merge to retrieve an object
    address = (Address) em.find(Address.class, 1);
    System.out.println("Address unmodified:" + address);
    //We modify the object in the same transaction
    address.setStreetNumber("555");
    em.getTransaction().commit(); //hacemos flush
} catch (Exception e) {
    em.getTransaction().rollback();
    e.printStackTrace(System.out);
} finally {
    if (em != null) {
        em.close();
    }
}

//We print the values (status detached)
System.out.println("Address recovered:" + address);
}
```

15. EXECUTE THE JAVA CLASS

Execute the Java class:



15. EXECUTE THE JAVA CLASS

Execute the Java class:

```
Output - Run (State4ModifyObjectLongSession) X
12:21:57 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
12:21:57 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations {5.0.4.Final}
12:21:57 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
12:21:57 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sm
12:21:57 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
12:21:57 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
12:21:57 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size:
12:21:57 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
12:21:58 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
12:21:58 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addr1_0_0_, address0_.country as country2_0_0_, address0_.deleted as dele
Hibernate: select address0_.id_address as id_addr1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_nam
12:21:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
Address unmodified:Address{idAddress=1, streetName=new street, streetNumber=109, country=Mexico, version=0, deleted=0}
12:21:58 [main] DEBUG org.hibernate.SQL - update address set country=?, deleted=?, street_name=?, street_number=?, version=? where id_address=?
Hibernate: update address set country=?, deleted=?, street_name=?, street_number=?, version=? where id_address=?
12:21:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [Mexico]
12:21:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [0]
12:21:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [VARCHAR] - [new street]
12:21:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [555]
12:21:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [INTEGER] - [0]
12:21:58 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [INTEGER] - [1]
Address recovered:Address{idAddress=1, streetName=new street, streetNumber=555, country=Mexico, version=0, deleted=0}
```

HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

EXERCISE CONCLUSION

- With this exercise we have seen the life cycle of entity objects supported by the JPA API.
- The life cycle of Entity objects in Hibernate / JPA is very important to know, since it depends on that we can correctly perform the operations on our Entity objects and at the end on our records in the respective database table.
- With this we conclude the theme of life cycle in Hibernate / JPA.

ONLINE COURSE

HIBERNATE & JPA

By: Eng. Ubaldo Acosta



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx