**Steps to Create Hibernate Application Using Eclipse IDE:**

Step-1: Create Java project in Eclipse IDE.

Step-2: Create user-defined library with hibernate jars and ojdbc6.jar using eclipse IDE.

Step-3: Define Domain class.

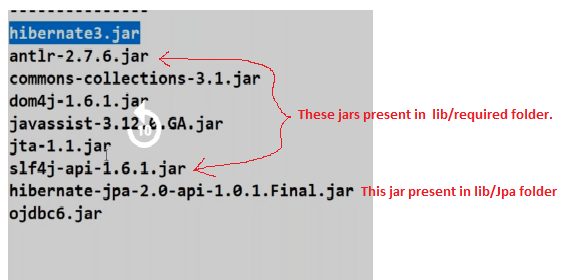
Step-4: Create Hibernate Configuration File.

Step-5: Create Hibernate Mapping File.

Step-6: Create Client Application.

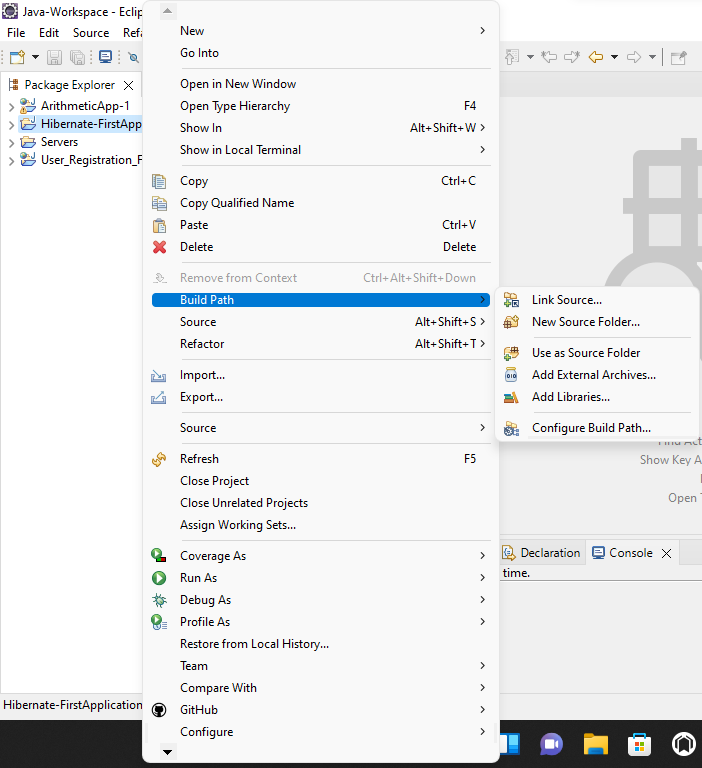
Step-7: Run Client Application.

**Step2:-Create user-defined library with hibernate jars and ojdbc6.jar using eclipse IDE:**

With following jar files, we have to create user-defined library.

a.Right click on project Name.

Buildpath 🡪 configure build path.



b. click on Add library.

A screenshot of a computer

Description automatically generated

c. select “User library” and click next button.

A screenshot of a computer program

Description automatically generated

d. click on user libraries button.

A screenshot of a computer

Description automatically generated

e. click on “new” button . Enter the user-library name in text box and press “ok button”.

A screenshot of a computer

Description automatically generated

f. press the “Add external jars” and add above specified jars to our library.

A screenshot of a computer

Description automatically generated

g. press apply and close 🡪 finish -🡪 apply and close buttons.

Step3: Define a Domain class.

**package** com.hib.domain;

**import** java.io.Serializable;

**public** **class** Emp **implements** Serializable {

**private** **int** eid;

**private** String ename;

**private** **float** sal;

**public** Emp() {

}

**public** **int** getEid() {

**return** eid;

}

**public** **void** setEid(**int** eid) {

**this**.eid = eid;

}

**public** String getEname() {

**return** ename;

}

**public** **void** setEname(String ename) {

**this**.ename = ename;

}

**public** **float** getSal() {

**return** sal;

}

**public** **void** setSal(**float** sal) {

**this**.sal = sal;

}

}

Step4: create Hibernate Mapping File.(Emp.hbm.xml)

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name=*"com.hib.domain.Emp"* table=*"emp"*>

<id name=*"eid"* column=*"eid"* type=*"int"*/>

<property name=*"ename"* column=*"ename"* type=*"string"*/>

<property name=*"sal"* column=*"sal"* type=*"float"*/>

</class>

</hibernate-mapping>

Step5: create hibernate Configuration File(hibernate.cfg.xml)

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>SYSTEM</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.Oracle10gDialect</property>

<mapping resource=*"Emp.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

Step6: create client application(Test.java)

**package** com.hib.domain;

**import** java.util.Scanner;

**import** org.hibernate.HibernateException;

**import** org.hibernate.MappingException;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**try** {

Configuration cfg=**new** Configuration();

cfg.configure("hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session sv=sf.openSession();

Scanner s1=**new** Scanner(System.***in***);

Transaction t=sv.beginTransaction();

Emp e1=**new** Emp();

System.***out***.print("Enter the eno: ");

e1.setEid(s1.nextInt());

s1.nextLine();

System.***out***.print("Enter the ename: ");

e1.setEname(s1.nextLine());

System.***out***.print("Enter the sal:");

e1.setSal(s1.nextFloat());

**int** pk=(Integer)sv.save(e1);

**if**(pk==e1.getEid()) {

System.***out***.println("record Inserted");

}

**else**

{

System.***out***.println("record not Inserted");

}

Emp e2=**new** Emp();

e2=(Emp)sv.get(Emp.**class**,e1.getEid());

System.***out***.println("eid"+"\t"+"ename"+"\t"+"salary");

System.***out***.println("-------------------------------");

System.***out***.println(e2.getEid()+"\t"+e2.getEname()+"\t"+e2.getSal());

System.***out***.print("Enter the pk value of record to Delete:");

Emp e3=**new** Emp();

e3.setEid(s1.nextInt());

sv.delete(e3);

System.***out***.println("Record Deleted");

t.commit();

sv.close();

sf.close();

}

**catch**(MappingException e) {

System.***out***.println(e.getMessage());

}

**catch**(HibernateException e) {

System.***out***.println(e.getMessage());

}

}

}

Project Explorer

-------------------

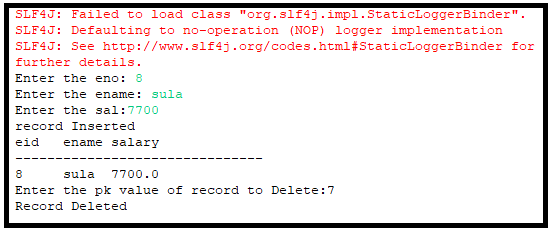
A screenshot of a computer program

Description automatically generated

**Output:-**

**A black screen with white text

Description automatically generated**



**A screen shot of a computer

Description automatically generated**