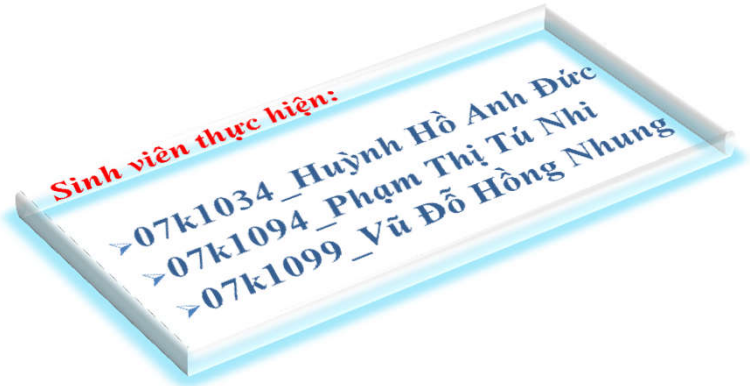
NHibernate


Sinh viên thực hiện:
07k1034_Huỳnh Hồ Anh Đức
07k1094_Phạm Thị Tú Nhi
07k1099_Vũ Đỗ Hồng Nhung

NHIBERNATE

I. Mô tả

II. Hoạt động

I. MÔ TẢ

I. MÔ TẢ

1. Khái Niệm
2. Lịch Sử
3. Chức năng

WHAT'S NHIBERNATE?

1 • Khái Niệm

NHibernate là một cảng của Hibernate Core cho Java vào Framework. NET và một số ứng dụng khác.

Nó kiên trì xử lý các đối tượng đồng bằng NET. Đến và từ một CSDL quan hệ cơ bản.

Với một mô tả XML của các thực thể và mối quan hệ.

NHibernate tự động tạo SQL cho tải và lưu trữ các đối tượng.

NHibernate không phải theo một mô hình lập trình hạn chế.

Các lớp học liên tục không cần phải thực hiện bất kỳ giao diện hay kế thừa từ một lớp cơ sở đặc biệt

Điều này làm cho nó có thể thiết kế logic kinh doanh bằng cách sử dụng đồng bằng NET (. CLR) các đối tượng và thành ngữ hướng đối tượng

NHibernate không phải theo một mô hình lập trình hạn chế

Các lớp học liên tục không cần phải thực hiện bất kỳ giao diện hay kế thừa từ một lớp cơ sở đặc biệt

Điều này làm cho nó có thể thiết kế logic kinh doanh bằng cách sử dụng đồng bằng NET (. CLR) các đối tượng và thành ngữ hướng đối tượng.

2 • Lịch sử hình thành

- NHibernate được bắt đầu bởi Tom Barrett, và sau đó được chỉnh sửa bởi Mike Doerfler và Peter Smulovics.
- Vào cuối năm 2005, JBoss , Inc (nay là một phần của Red Hat) thuê Sergey Koshcheyev, sau đó dẫn dắt và phát triển NHibernate, để làm việc toàn thời gian trên các phiên bản tương lai
- Vào cuối năm 2006 JBoss ngừng hỗ trợ để dự án này; ngày nay nó hoàn toàn phát triển và do cộng đồng.

CÁC PHIÊN BẢN CỦA NHIBERNATE

- Phiên bản 1.0 được nhân đôi các tính năng thiết lập của Hibernate 2.1, cũng như một số tính năng từ Hibernate 3.
- . NHibernate 1.2.1, phát hành vào tháng mười một năm 2017 , được giới thiệu thêm nhiều tính năng từ Hibernate 3 và hỗ trợ cho, NET. 2,0 stored procedures , generics, và các loại nullable.
- NHibernate 2,0 đã được phát hành ngày 23 Tháng Tám 2008 Nó được so sánh với Hibernate 3.2 về tính năng.Với phiên bản 2.0 phát hành, NHibernate bỏ hỗ trợ 1.1. NET
- NHibernate 2,1 đã được phát hành 17 tháng 7 năm 2009.
- NHibernate 3,0 sẽ là phiên bản đầu tiên sử dụng NET 3.5..

3

•Chức Năng

CÁC TÍNH NĂNG CHÍNH

- Tính năng chính của NHibernate là ánh xạ từ các loại NET. lớp học CSDL để bàn (và CLR từ dữ liệu SQL các loại dữ liệu)

- **Natural programming model - Mô hình lập trình tự nhiên** - NHibernate hỗ trợ thành ngữ OO tự nhiên; thừa kế, đa hình, thành phần, các bộ sưu tập, bao gồm cả các bộ sưu tập chung

- **Support for fine-grained object models .Hỗ trợ cho các mô hình đối tượng hạt tinh** - một loại phong phú của các ánh xạ cho các bộ sưu tập và phụ thuộc các đối tượng

CÁC TÍNH NĂNG CHÍNH

- **No build-time bytecode enhancement .Không có thời gian tăng cường bytecode-xây dựng** - không có mã số thế hệ phụ hoặc các bước chế biến bytecode trong thủ tục xây dựng

- **The query options -Các truy vấn lựa chọn** - NHibernate địa chỉ cả hai mặt của vấn đề; không chỉ làm thế nào để có được các đối tượng vào CSDL, mà còn làm thế nào để có được họ trở lại

- **Custom SQL - Custom SQL** - xác định chính xác rằng SQL NHibernate nên sử dụng để kéo dài các đối tượng của bạn. Stored procedures are supported on Microsoft SQL Server. thủ tục lưu trữ được hỗ trợ trên Microsoft SQL Server.

CÁC TÍNH NĂNG CHÍNH

- **Support for "conversations“- Hỗ trợ cho "cuộc hội thoại" -** NHibernate hỗ trợ hoàn cảnh sống bền bỉ, lâu dài, tháo / lắp lại của các đối tượng, và sẽ chăm sóc của khóa tự động lạc quan

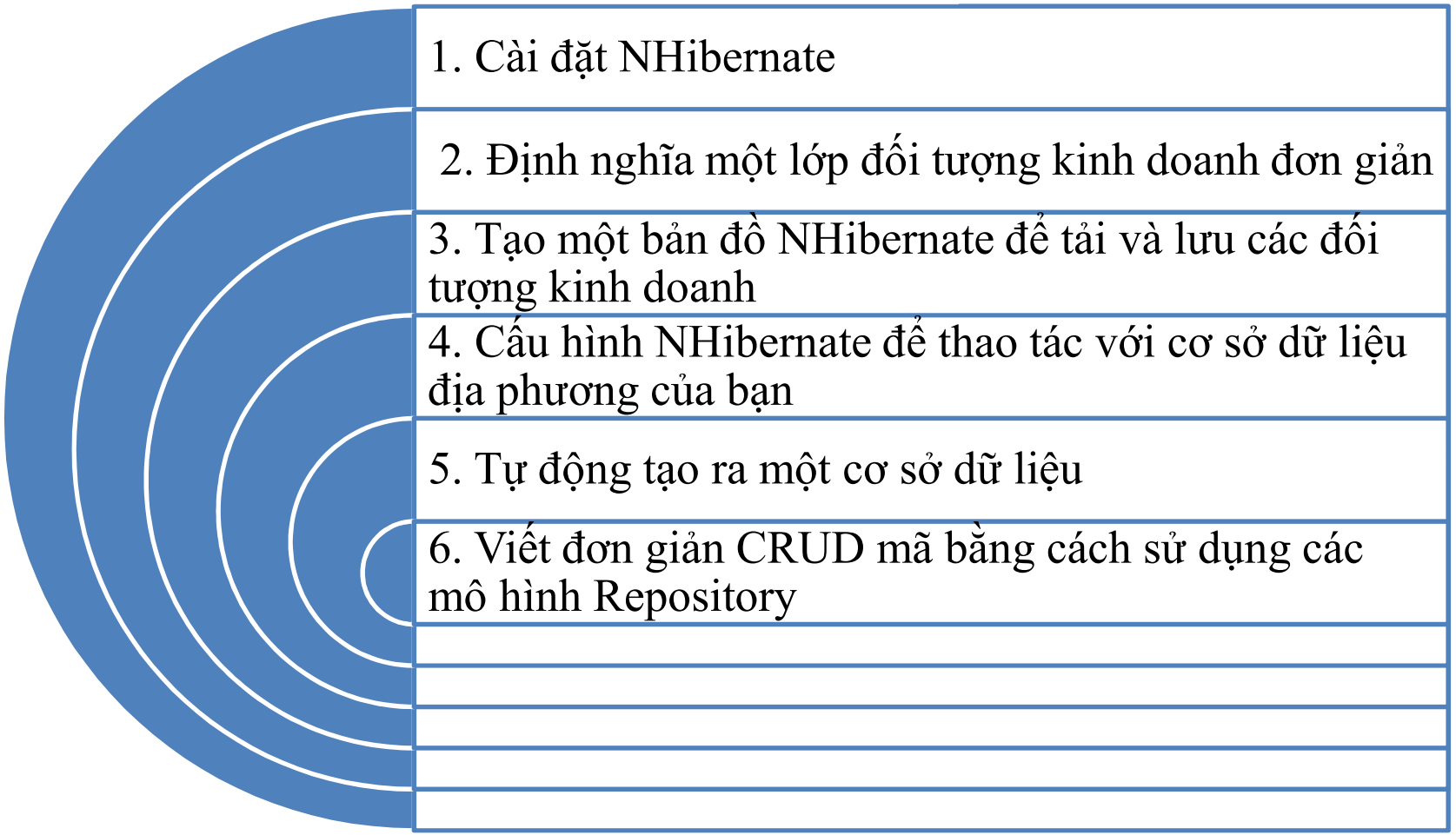
- NHibernate cũng cung cấp dữ liệu truy vấn và các phương tiện cứu

- NHibernate tạo ra các lệnh SQL và giúp các nhà phát triển từ dữ liệu hướng dẫn sử dụng bộ xử lý và chuyển đổi đối tượng, lưu giữ các ứng dụng di động cho hầu hết các CSDL SQL, với Portability CSDL trên không thực hiện giao tại rất ít

- **Free/open source - Miễn phí / mã nguồn mở -** NHibernate được cấp phép theo LGPL (Lesser GNU Public License)

II. HOẠT ĐỘNG

Tìm hiểu Nhibernate



1. CÀI ĐẶT NHIBERNATE

- Tải xuống các tập tin NHibernate-2.1.2.GA-bin

- tạo ra một thư mục có tên Nhibernate-Demo ngoài Desktop (C:\Users\AnhDuc\Desktop\Nhibernate-Demo)và giải nén tập tin vừa tải

TẠO DỰ ÁN

- Để bắt đầu xây dựng 1 dự án mới .Bạn thực hiện các bước sau

Bước 1: Tao ra 1 Folder con tên "UngDungNhibernate"(tên này do bạn đặt dùng để chứa tất cả Project của bạn

Bước 2: Mở VS 2005 (các phiên bản mới nhất của VS) tạo 1 Solution co tên "Nhibernate-Solution"

Bước 3: Add 1 ClassLibrary đặt tên "ThuVienNhibernate" Lớp thư viện này

Bước 4: Create 1 Folder co tên "UngDung-Nhiberbate" để chứa Ứng dụng của bạn

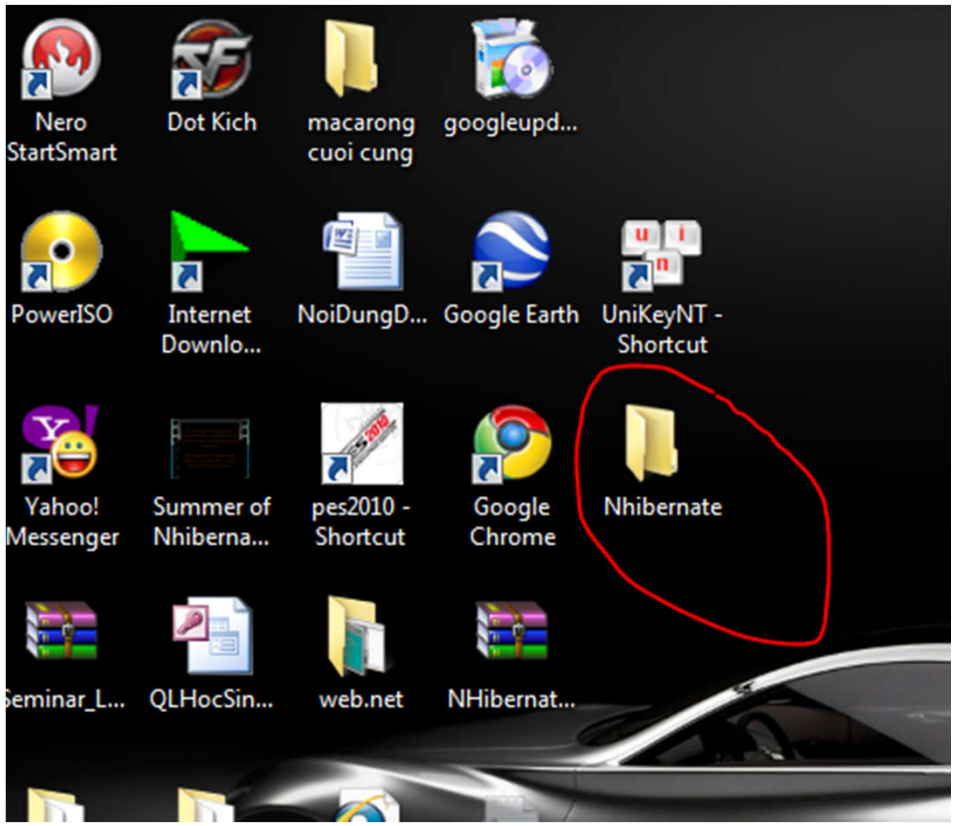
Sau đây là Demo tạo và Cài đặt Nhbernate

DEMO TẠO VÀ CÀI ĐẶT NHIBERNATE

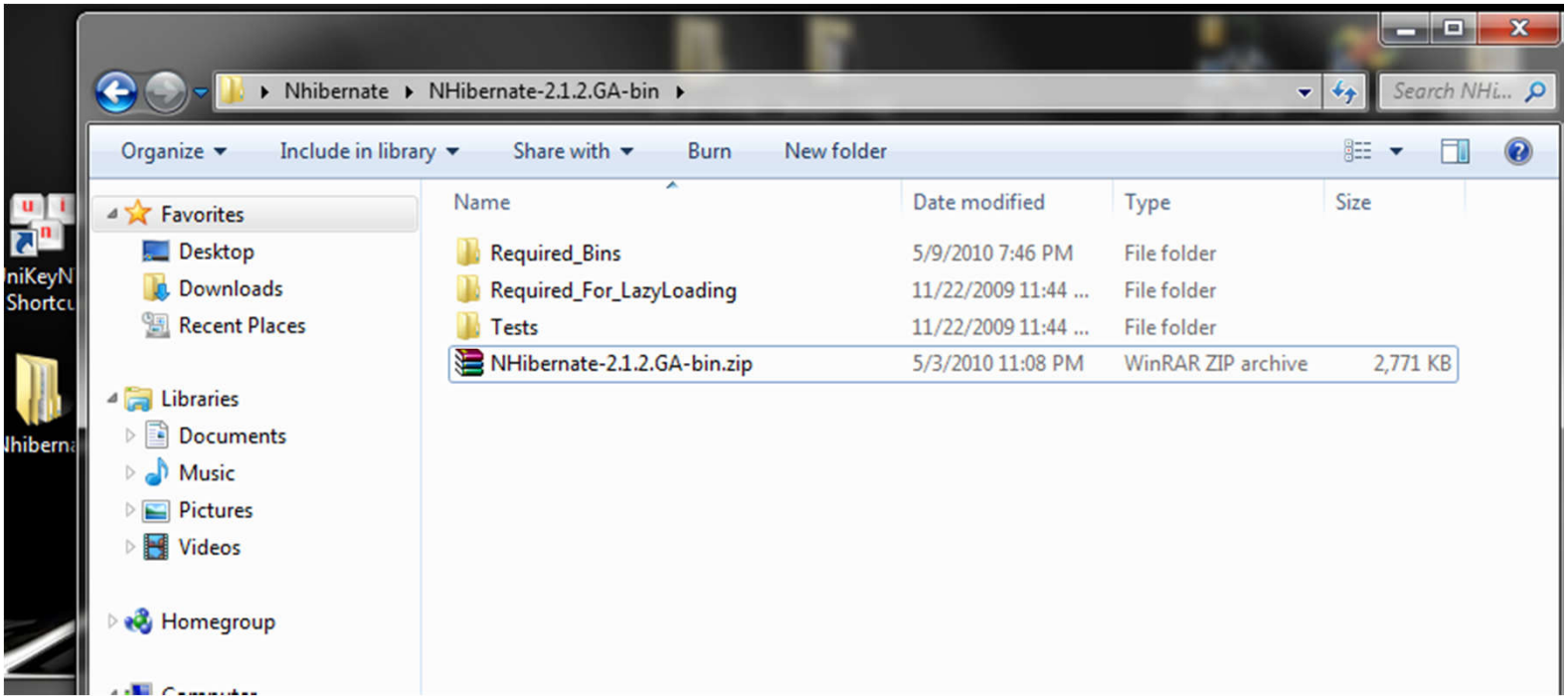

Recycle Bin
Nero StartSmart
Dot Kich
Microsoft Visual We...
googleupd...
SQL Server Manageme...
Microsoft Visual Stud...
thuyet trinh c4w
Adobe Reader 9
PowerISO
Internet Downlo...
NoiDungD...
Google Earth
UniKeyNT - Shortcut
INFINITI
Inspired Performance
Total Command...
sdfsdfs.txt
a
avast! Antivirus
Yahoo! Messenger
Microsoft Visual C# ...
pes2010 - Shortcut
Google Chrome
web
Seminar_L...
QLHocSin...
web.net
Capture.PNG
Virtual DJ
hoa moc lan
ong ngoai tuoi 30
edwincieza...
NHibernat...
07k1103_QL... - Shortcut
165139_AR...
vay lg
AjaxContro...
Kiem tra Toan 7 HKI...
New Text Docume...
settings - Shortcut
Nhibernate2
C4W.rar
infiniti.com/windows7

TẠO VÀ CÀI ĐẶT NHIBERNATE

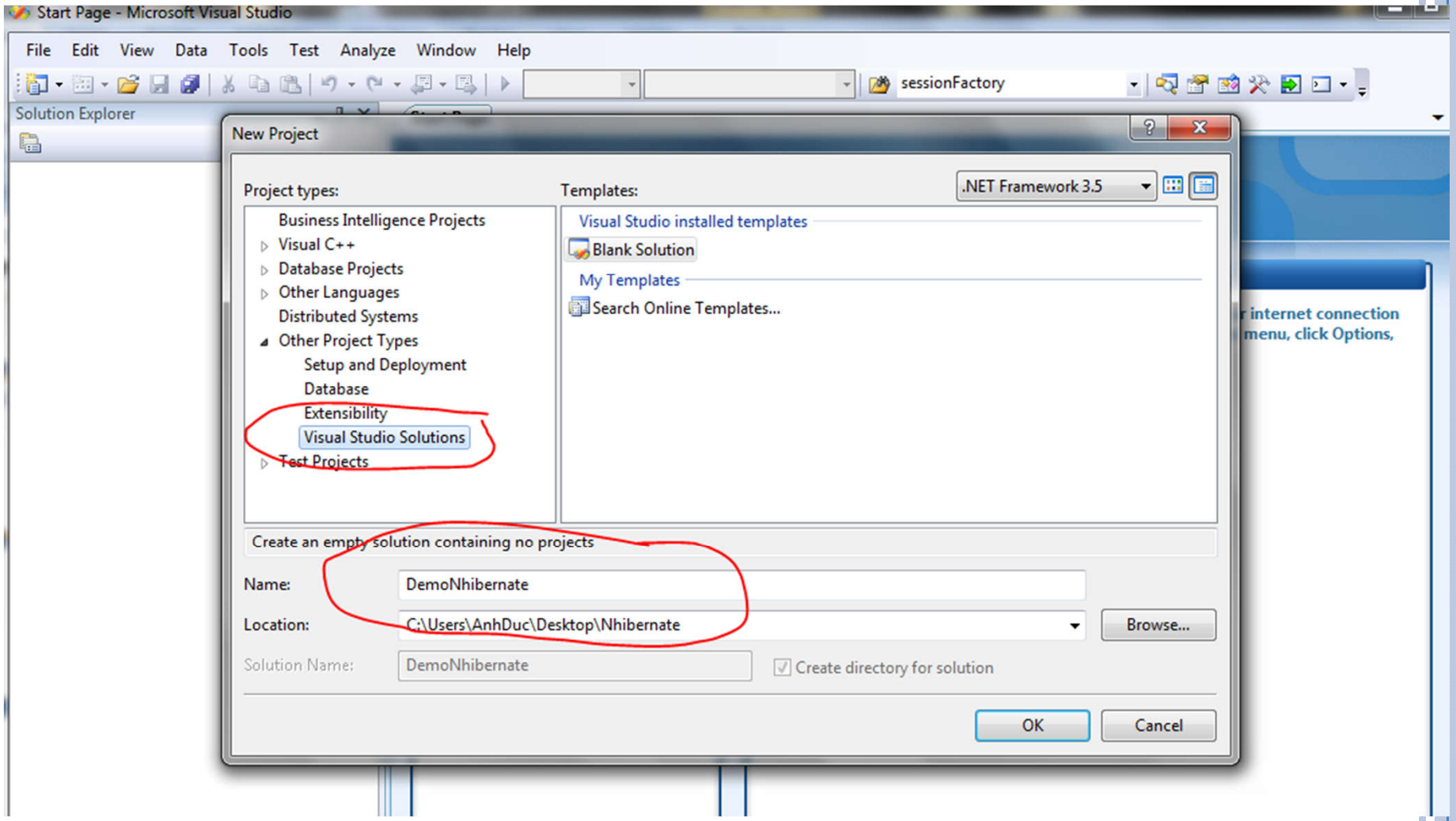
B1 :TAO 1 FOLDER Ở NƠI BẠN THÍCH(DESKTOP)



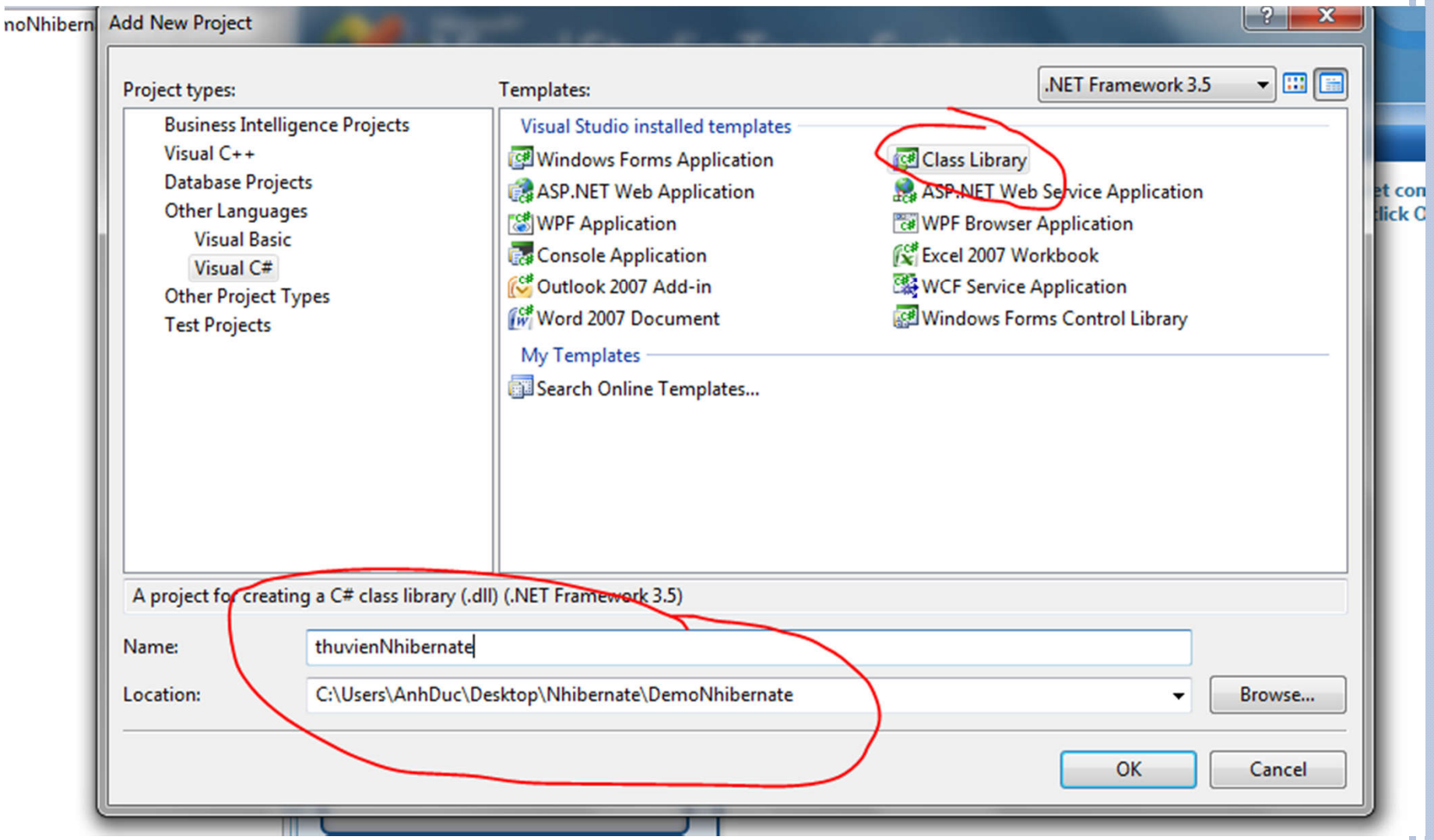
B2:TẢI NHIBERNATE-2.1.2.GA-BIN.ZIP (GOOGLE.COM)
LƯU VÀO THƯ MỤC VỪA TẠO GIẢI NÉN NÓ



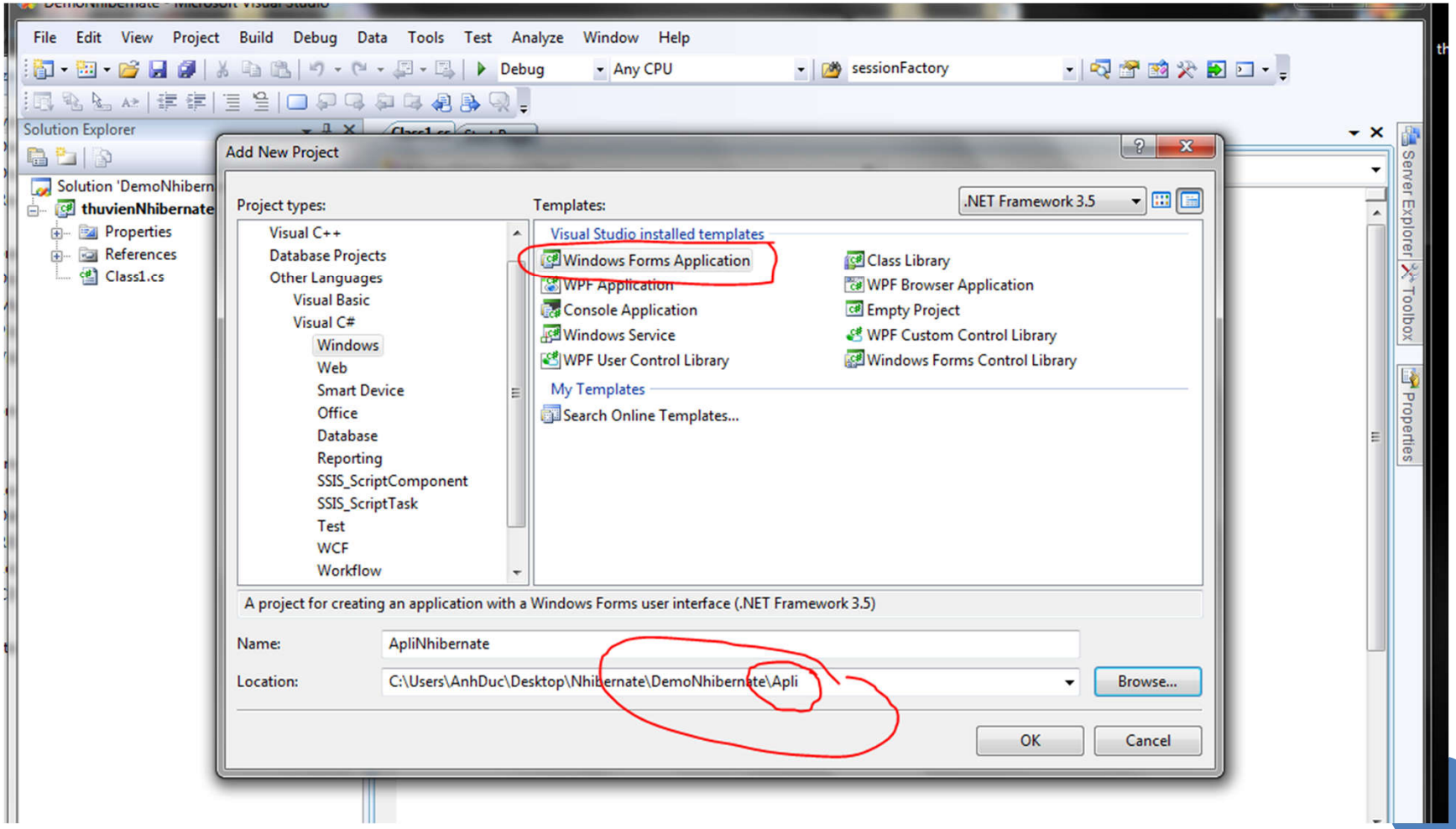
B3:Mở VS(phiển bản nào cũng được) tạo 1 solution đặt tên nào bạn thích(demoNhibernate)lưu nó vào folder bạn tạo



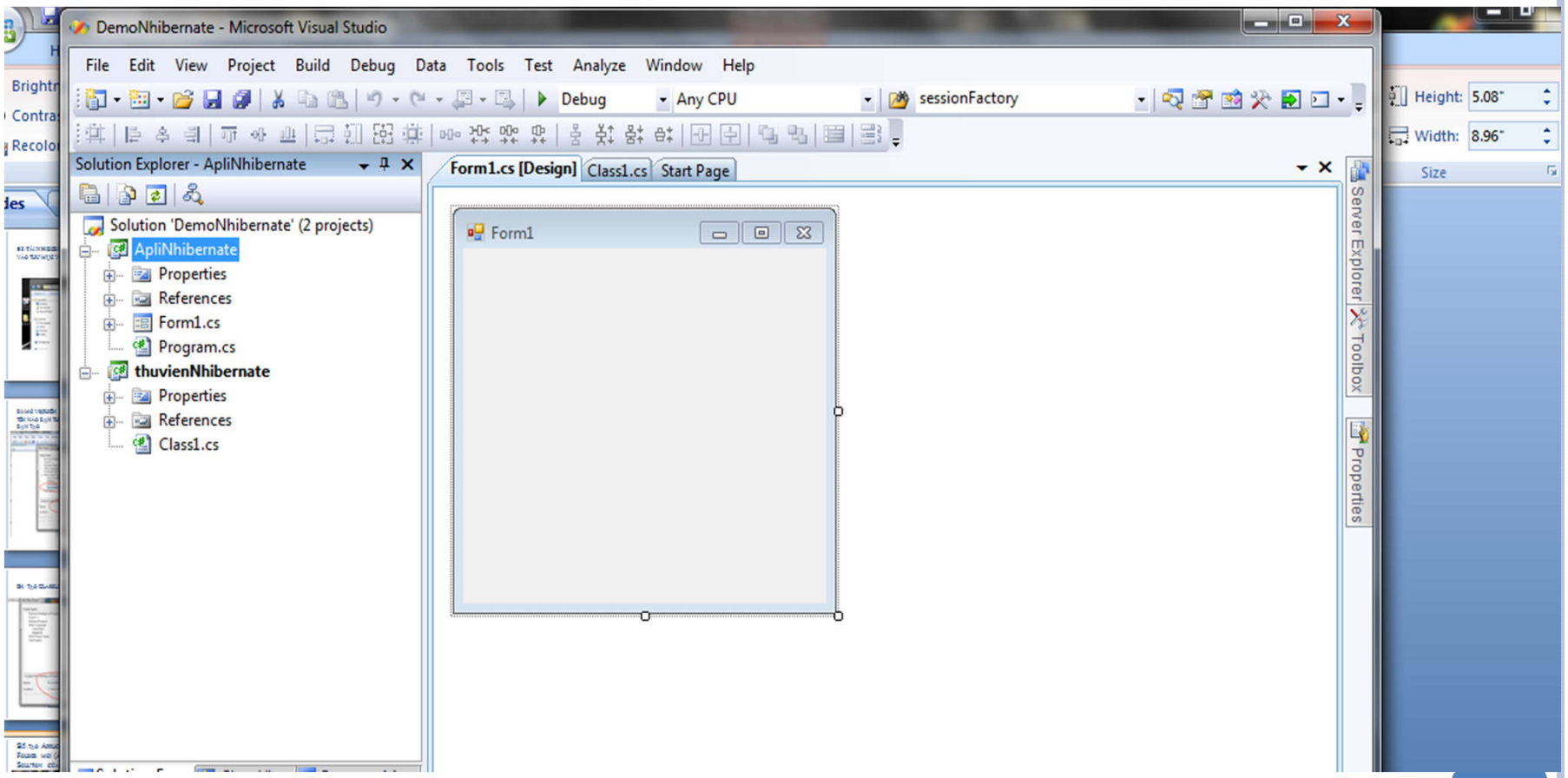
B4: TẠO CLASSLIBRARY (THUVIENNHIBERNATE)



B5.TẠO APPLICATION (APLINHIBERNATE) LƯU VÀO FOLDER MỚI (APLI) RỒI LƯU FOLDER NÀY VÀO SOLUTION CỦA BẠN

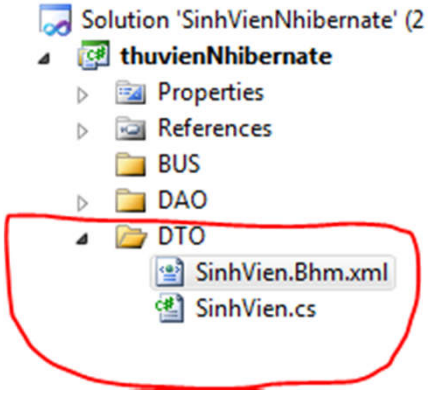


GIAO DIỆN CỦA SOLUTION CỦA BẠN SẼ NHƯ THẾ NÀY



2. XÁC ĐỊNH ĐỐI TƯỢNG KINH DOANH

- Cho phép bắt đầu bằng cách xác định một miền rất đơn giản:
- Tạo 1 Folder có tên DTO thuộc lớp thư viện thuvienNhibernate (lớp thư viện tạo trước đó)
- Tạo 2 file như hình dưới



Lớp đầu tiên đặt tên SinhVien.cs: có khai báo như sau :

```
namespace thuvienNhibernate.DTO
{
    public  class SinhVien
    {
    private int _Mssv, MaLop,;
    private string _HoTen,;
    private  string _GioiTinh;
    private string _DiaChi;
    public int Mssv
    {
            get { return _Mssv; }
            set { _Mssv = value; }
     }
    public string HoTen
    {
            get { return _HoTen; }
            set { _HoTen = value; }
     }
    public string GioiTinh
    {
            get { return _GioiTinh; }
            set { _GioiTinh = value; }
    }
    public string DiaChi
    {
            get { return _DiaChi; }
            set { _DiaChi = value; }
    }
    public int MaLop1
    {
            get { return MaLop; }
            set { MaLop = value; }
    }
        }
}
```

3. XÁC ĐỊNH BẢN ĐỒ

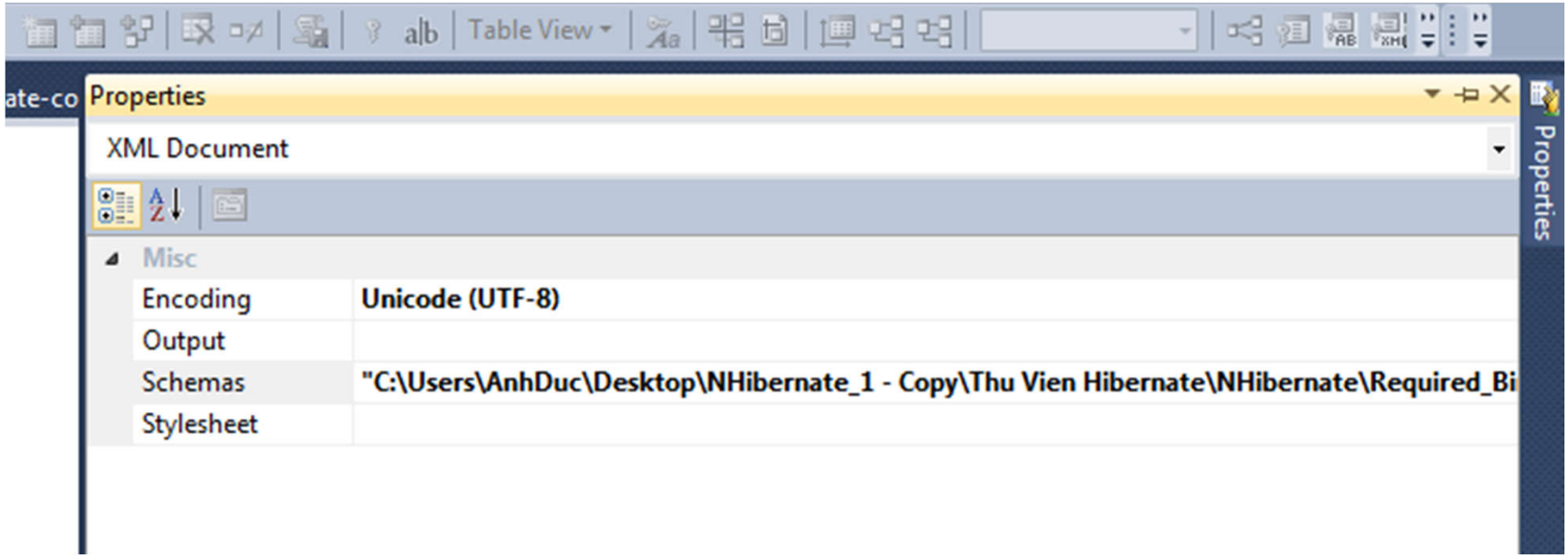
Để có thể tồn tại trường hợp của thực thể này trong một CSDL

NHibernate (Một thể hiện của một thực thể trong miền tương ứng với một **hàng trong một bảng** trong CSDL)

xác định một ánh xạ giữa các thực thể và bảng tương ứng trong CSDL

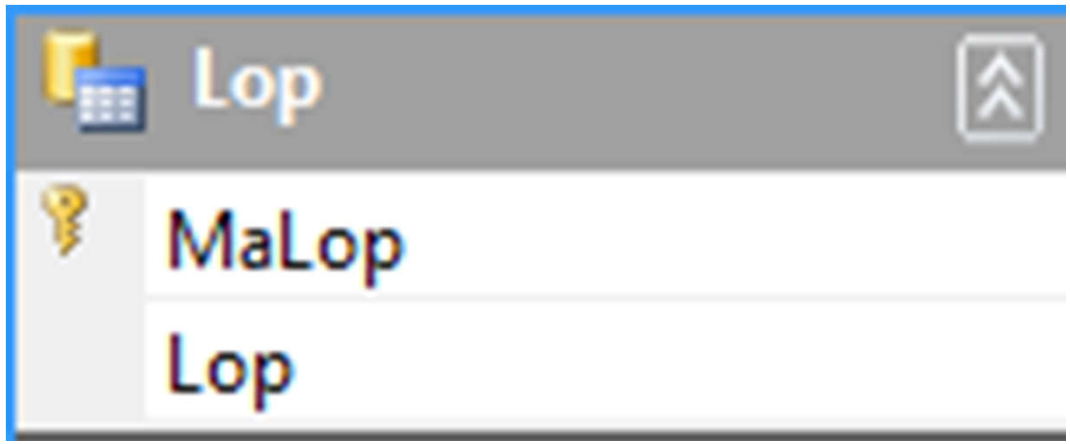
Lập bản đồ này có thể được thực hiện bằng cách xác định một tập Mapping (DTO) hoặc bằng cách trang trí các thực thể có thuộc tính

Properties thêm schema đến tập tin SinhVien.Bhm.xml



```
<?xml version="1.0" encoding="utf-8" ?>
<hibernate-mapping  xmlns="urn:nhibernate-mapping-2.2“
                    namespace="thuvienNhibernate“
                    assembly="thuvienNhibernate">
        <class name="SinhVien" table="SinhVien" lazy="false">
        <id name="Mssv" column="Mssv" unsaved-value="0">
        <generator class="native"/>  </id>
        <property name="HoTen"> <column name="HoTen" not-null="true"/></property>
        <property name="GioiTinh"> <column name="GioiTinh" not-null="true"/>
          </property>
        <property name="DiaChi"> <column name="DiaChi" not-null="true"/> </property>
        <property name="MaLop"> <column name="MaLop" not-null="true"/> </property>
        </class>
</hibernate-mapping>
```

- Tương tự tạo tiếp lớp thứ 2 :Lop.cs và cấu hình giống lớp SinhVien.cs



- Trong một tập tin bản đồ khi lớp tham chiếu đến một tên miền mà bạn luôn phải cung cấp tên đầy đủ của lớp

- Để thực hiện các xml có thể định nghĩa tên lắp ráp (trong đó các lớp miền được thực hiện và không gian tên của lớp học miền trong các thuộc assembly and namespace của nút gốc.

- Tương tự như báo cáo **sử dụng** trong C # lần đầu tiên xác định một **khóa chính** cho các đơn vị sản phẩm

- Về mặt kỹ thuật : có thể lấy **tên** tài sản của sản phẩm kể từ khi tài sản này phải được xác định và phải được duy nhất

4. CẤU HÌNH NHIBERNATE

NHibernate sở dữ liệu sản phẩm mà muốn sử dụng và cung cấp cho kết nối các chi tiết trong hình thức của một chuỗi kết nối

NHibernate hỗ trợ các sản phẩm CSDL nhiều

Thêm một file xml mới cho dự án thuvienNhibernate và gọi là hibernate.cfg.xml

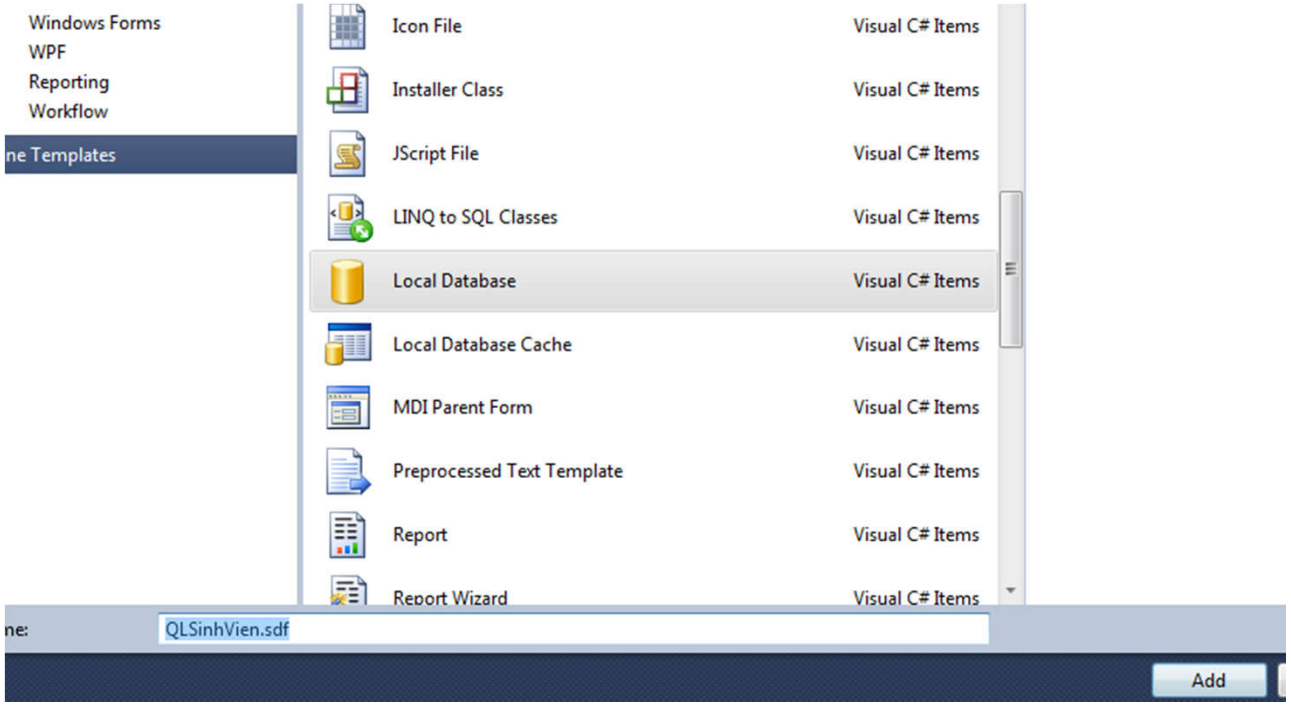
Set its property " Copy to Output " to " Copy always "

- Cấu hình như sau :

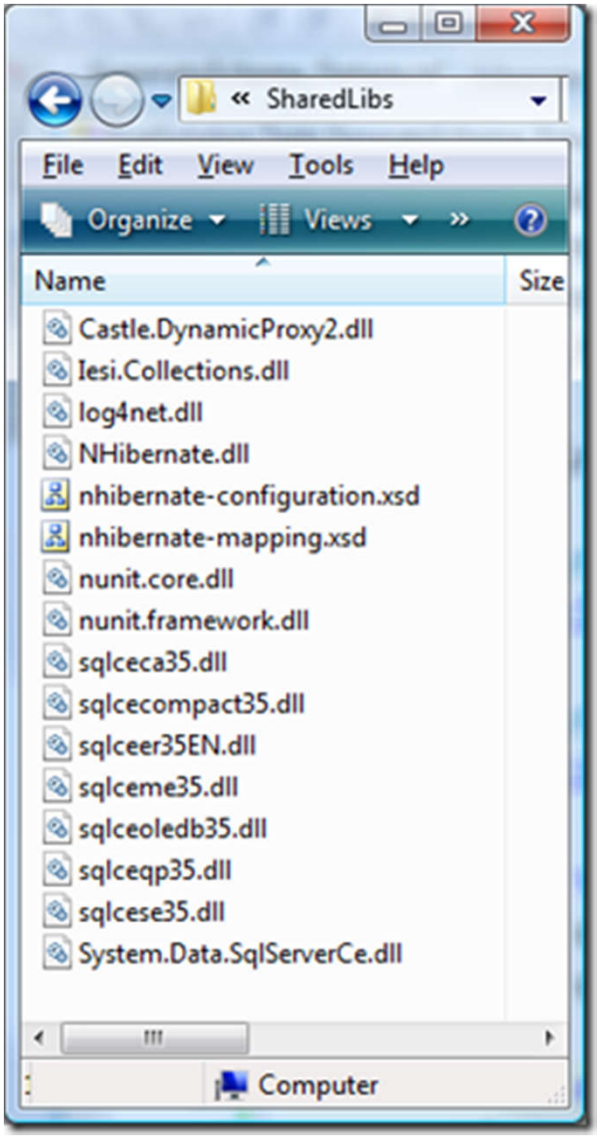
```
<?xml version="1.0" encoding="utf-8" ?>
<hibernate-configuration xmlns="urn:nhibernate-configuration-2.2">
<session-factory>
<property
    name="connection.provider">NHibernate.Connection.DriverConnectionProvider</prope
    rty>
<property name="dialect">NHibernate.Dialect.MsSqlCeDialect</property>
<property
    name="connection.driver_class">NHibernate.Driver.SqlServerCeDriver</property>
<property name="connection.connection_string">Data
    Source=C:\Users\AnhDuc\Desktop\NHibernate_1\SinhVienNhibernate\thuvienNhibernat
    e\QLSinhVien.sdf;Max Database Size=2047</property>
<property name="show_sql">True</property>
</session-factory>
</hibernate-configuration>
```

5. CẤU HÌNH NHIBERNATE

Thêm một CSDL trống rỗng, gọi là **QLSinhVien.sdf** cho dự án thuvienNhibernate (chọn **CSDL địa phương** như bản mẫu)

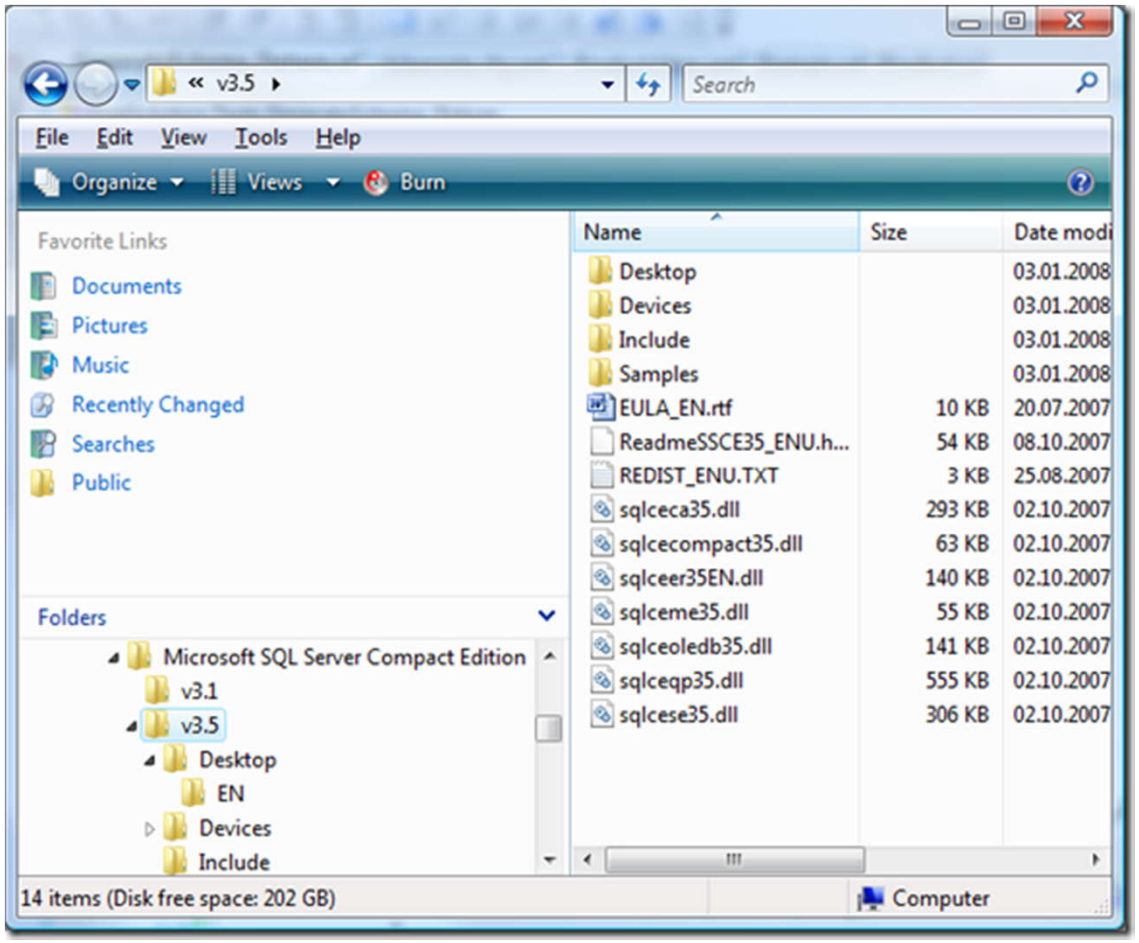


KIỂM TRA CÁC THIẾT LẬP



- Đầu tiên xác minh có những file sau trong thư mục.

  Nhibernate tạo ở đầu file

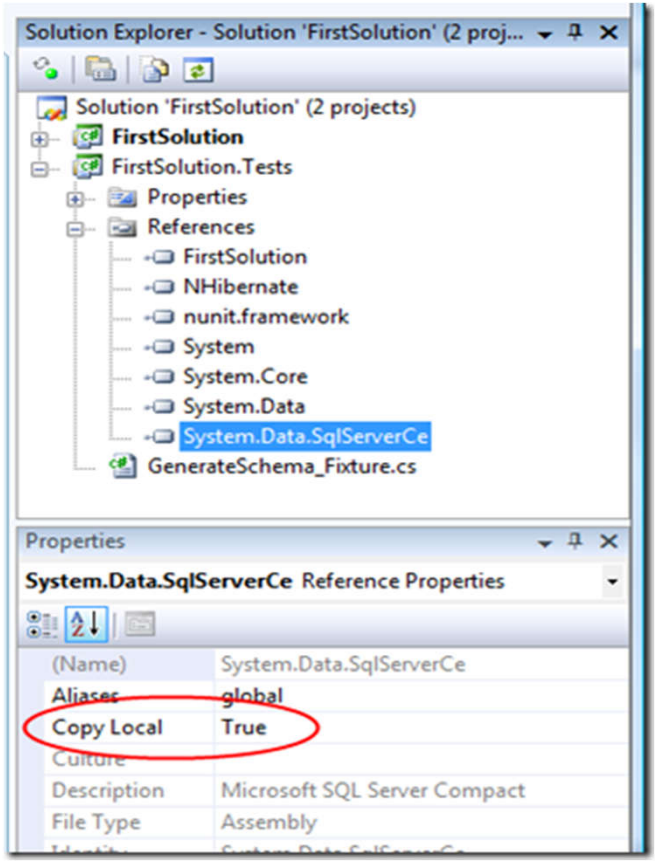


- Lưu ý: các System.Data.SqlServerCe.dll nằm ở thư mục Desktop-sub
- Tất cả các tập tin khác có thể tìm thấy trong thư mục NHibernate

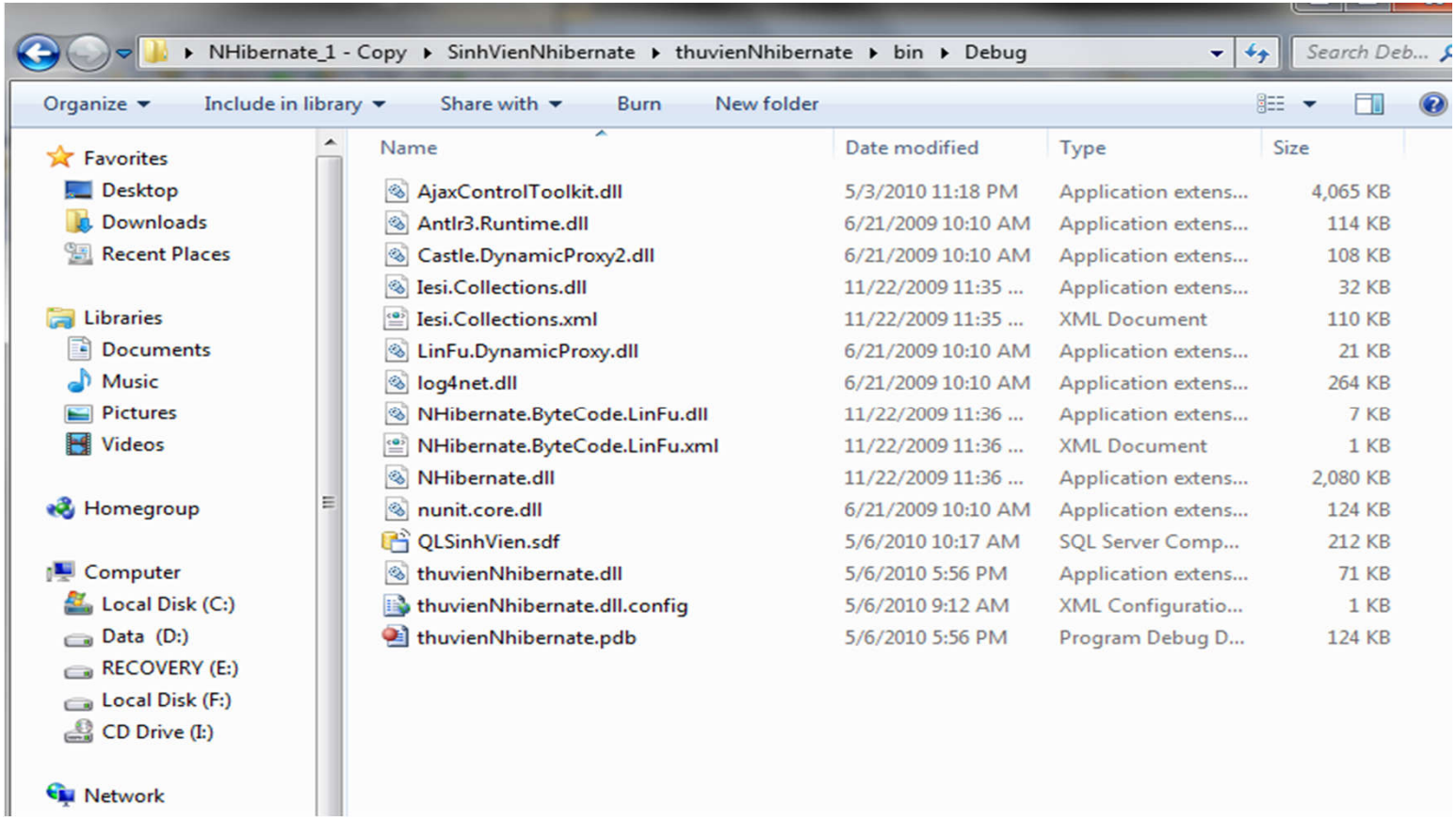
- Thêm một tham chiếu đến các dự án thuvienNhibernate trong dự án thử nghiệm

- Thêm một bản sao của hibernate.cfg.xml vào thư mục gốc của dự án này kiểm tra đơn vị. Trực tiếp hành động với NHibernate trong dự án NUnit nhu cầu truy cập vào các file này

- Ngoài ra thêm tài liệu tham khảo để NHibernate.dll, nunit.framework.dll và Systm.Data.SqlServerCe.dll



- Copy tất cả các file "*. dll " nằm trong NHibernate-2.1.2.GA-bin cho vao phan Debug của Ứng Dụng



TA CÓ CODE :

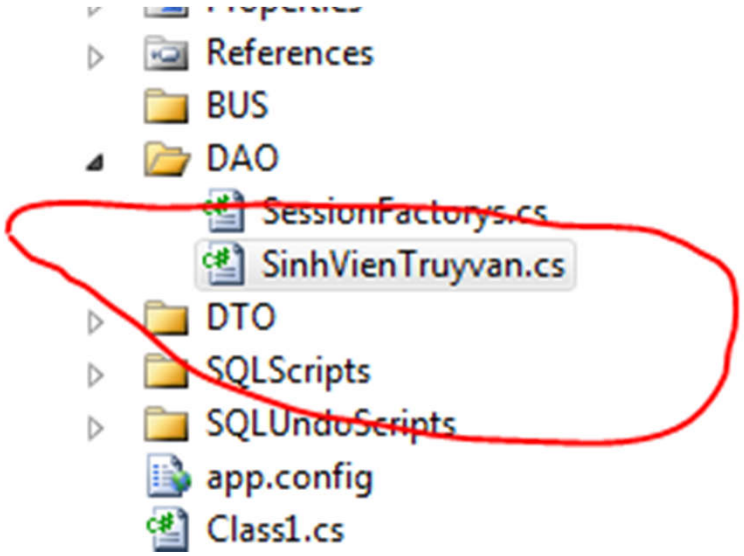
```
using System;
Using System.Collections.Generic;
using System.Linq;
using System.Text;
using thuvienNhibernate.DTO;
using thuvienNhibernate;
using System.Configuration;
namespace thuvienNhibernate.DAO
{
    public class SessionFactorys
    {
            private static NHibernate.ISessionFactory _SessionFactory;
            private static void Init()
            {
                    NHibernate.Cfg.Configuration config = new NHibernate.Cfg.Configuration();
                    config.Configure();
                    config.AddAssembly("thuvienNhibernate");
                    _SessionFactory = config.BuildSessionFactory();
     }
```

```
        public static NHibernate.ISessionFactory GetSessionFactory()
        {
                if (_SessionFactory == null)
                 {
                        Init();
                }
                 return _SessionFactory;

        }
        public static NHibernate.ISession GetnewSession()
        {
                return GetSessionFactory().OpenSession();
        }
    }
}
```

THIẾT LẬP TRUY VẤN ĐẾN CSDL

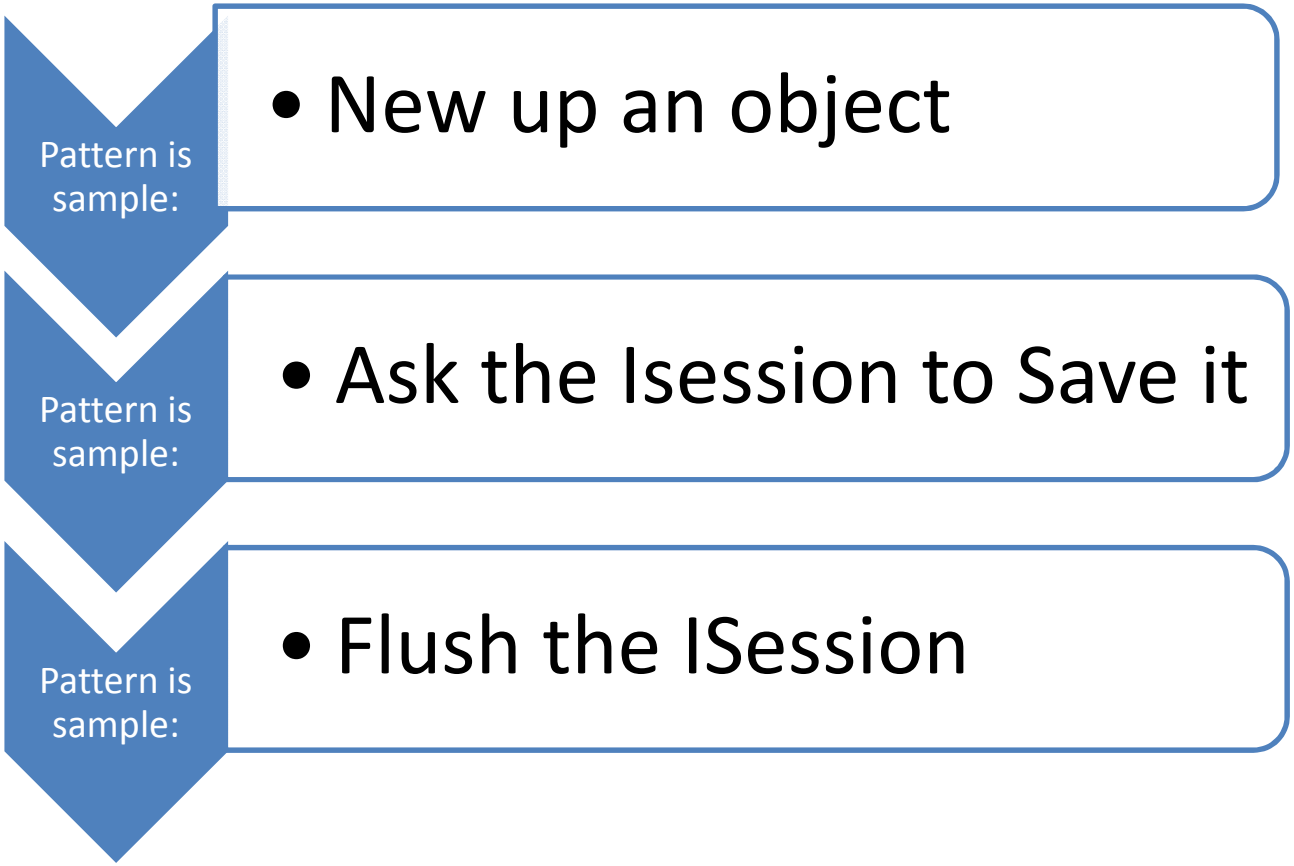
- Tạo class SinhVienTruyvan.cs .Với lớp này tạo các hàm Isert ,Delete,Update,Load CSDL



SELECT

```
public static DataTable GetTable()
{
        DataTable dt = new DataTable();
      OleDbConnection cn;
      cn = DataProvider.ConnectionData();
      string strSQL;
      strSQL = "Select * From Sach";
      OleDbDataAdapter da;
      da = new OleDbDataAdapter(strSQL, cn);
      da.Fill(dt);
      cn.Close();
      return dt;
}
```

Examining INSERTs with NHibernate



INSERT

```
public static void Add(SinhVien sv)
{
        ISession session =
   NHibernateHelper.GetCurrentSession();
        ITransaction transaction = session.BeginTransaction();
        session.Save(sv);
        session.Flush();
        transaction.Commit();
        NHibernateHelper.CloseSession();
}
```

Examining DELETEs with NHibernate

Pattern is sample:

- Have an object

Pattern is sample:

- Ask the Isession to Delete it

Pattern is sample:

- Flush the ISession

DELETE

```
public static void Add(SinhVien sv)
{
        ISession session =
   NHibernateHelper.GetCurrentSession();
        ITransaction transaction = session.BeginTransaction();
        session.Delete(sv);
        session.Flush();
        transaction.Commit();
        NHibernateHelper.CloseSession();
}
```

Examining UPDATEs with NHibernate

Pattern is sample:
- Have an object

Pattern is sample:
- Change a property

Pattern is sample:
- Ask the Isession to Update it

Pattern is sample:
- Flush the ISession

UPDATE

```
public static void Add(SinhVien sv)
{
        ISession session =
   NHibernateHelper.GetCurrentSession();
        ITransaction transaction = session.BeginTransaction();
        session.Update(sv);
        session.Flush();
        transaction.Commit();
        NHibernateHelper.CloseSession();
}
```

INDENTIFYING ANNOYANCES IN THESE USAGE PATTERN

- Allows call like session.SaveOrUpdate(sv);
- Nhibernate will …
  - Check if the object is in the sesion
  - If its not there,call Implicit Save(sv);
  - If its there ,check to see it the object is “dirty”(changed)
  - If its dirty,call implicit Update(sv);
- SaveOrUpdate() relieves us of having to keep track of “dirty” state ourselves

6. CRUD HOẠT ĐỘNG

- Hệ thống đã sẵn sàng để bắt đầu. Đã thực hiện thành công tên miền, quy định các tập tin Mapping và NHibernate Config. Cuối cùng đã sử dụng NHibernate để tự động tạo ra các lược đồ CSDL từ tên miền(và các tập tin lập bản đồ)

- Giao diện repository là một phần của tên miền

- Thêm một thư mục BUS thuộc dự án SinhVienNHibernate.
- Thêm một giao diện class có tên SinhVienBUS. Xác định các giao diện sau :

```
using System;
using System.Collections.Generic;
namespace ThuVienNHibernate.BUS
 { (
        public class SinhVienTruyvanBUS
        { (
                public static void  AddSv(SinhVien sv);
                public static void  RemoveSv(SinhVien sv);
                public static void  UpdateSv(SinhVien sv);
         } )
 } )
```

- Dòng _SessionFactory = config.BuildSessionFactory()

  Đây là một quá trình tốn kém và do đó nên được thực hiện chỉ một lần. Đó là lý do tại sao đặt nó vào phương pháp này là chỉ thực hiện một lần trong một chu kỳ kiểm tra

- Để giữ hiệu lực thử nghiệm các phương pháp, lại tạo ra CSDL schema trước khi thực hiện mỗi phương pháp thử nghiệm.

- Đó là bởi vì NHibernate là theo mặc định cấu hình để sử dụng tải lazy cho tất cả các thực thể. Đó là cách tiếp cận đề nghị và tôi khuyên bạn nên nồng nhiệt không phải để thay đổi nó cho tối đa là tính linh hoạt

- Bây giờ đã sẵn sàng để thực hiện các phương pháp khác cũng của **Repository**

- Đối với thử nghiệm này, thay vì sẽ có một kho lưu trữ (có nghĩa là CSDL bảng) đã có chứa một số sản phẩm. Chỉ cần thêm một phương pháp để CreateInitialData lớp kiểm tra

Cảm ơn mọi người đã quan tâm theo dõi!!!

Chúc thành công!!!