

CHUYÊN ĐỀ JAVA

# **HIBERNATE MAPPING**

## **ONE-TO-MANY**

Nguyễn Hoàng Anh

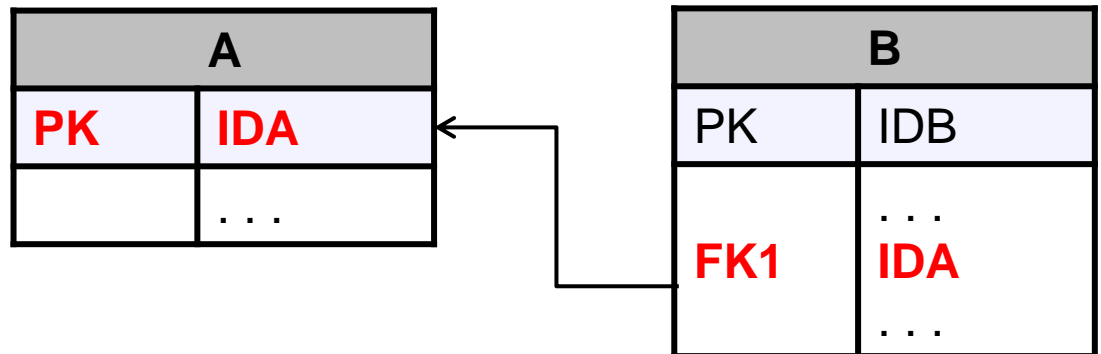
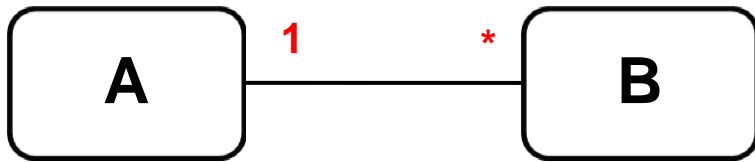
Email: [nhanh@fit.hcmus.edu.vn](mailto:nhanh@fit.hcmus.edu.vn)

ĐH KHTN, 2011

# Nội dung trình bày

- One to many
- Lazy Initialization
- Fetch
- Cascade

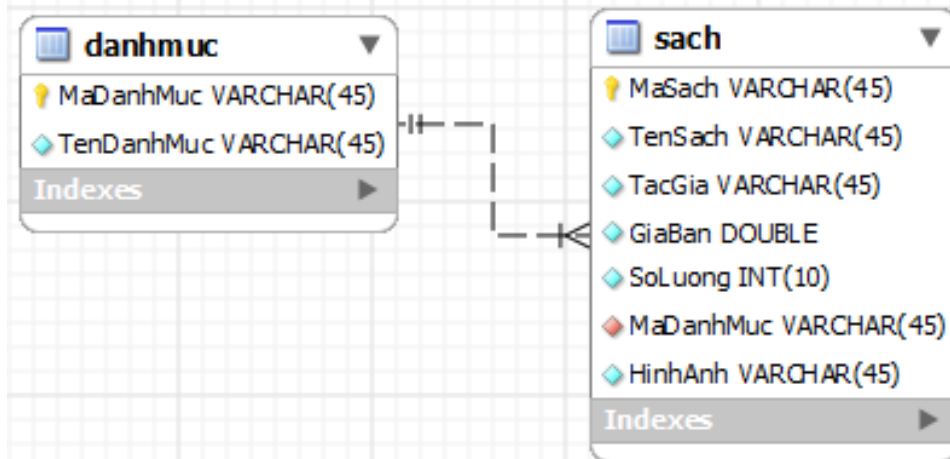
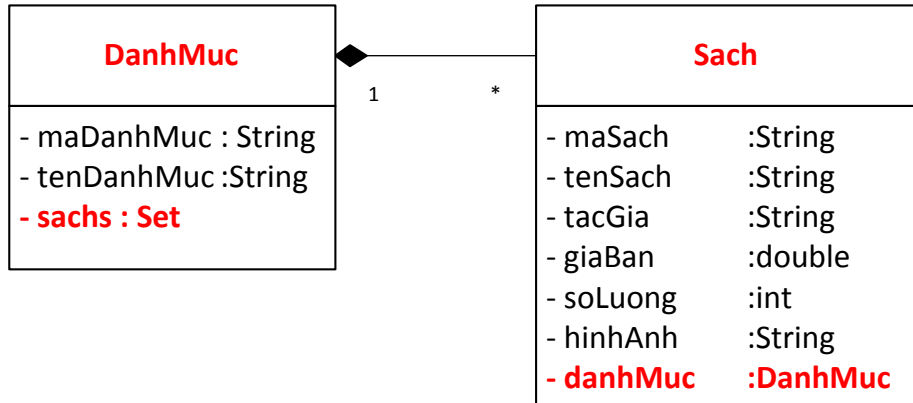
## Mối quan hệ một-nhiều (one-to-many)



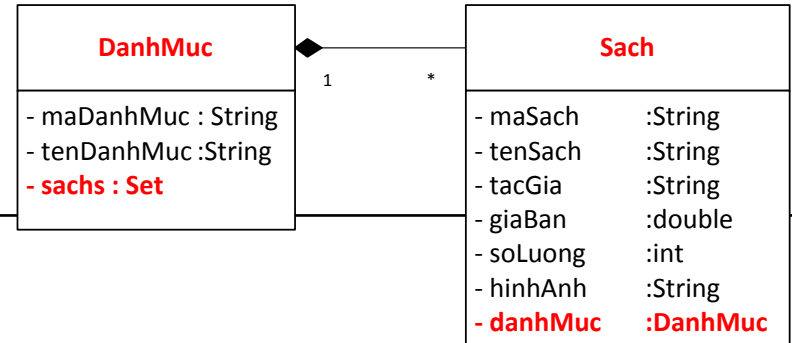
# Mối quan hệ **một nhiều (one-to-many)**

- Trong ứng dụng BookOnline
  - Mỗi đầu **sách** thuộc về một **danh mục**
  - Mỗi **danh mục** có thể có nhiều đầu **sách**
- Mỗi quan hệ hướng từ đầu **sách** đến **danh mục** được gọi là **mối quan hệ nhiều – một (many-to-one)**
- **Mối quan hệ hướng từ danh mục sang sách được gọi là mối quan hệ một-nhiều(one-to-many)**
- Nếu chỉ có mối quan hệ hướng từ sách đến danh mục hoặc ngược lại gọi là mối quan hệ một chiều (**unidirectional association**)
- Nếu mối quan hệ hướng cả từ sách đến danh mục và ngược lại gọi là mối quan hệ hai chiều (**bidirectional association**)

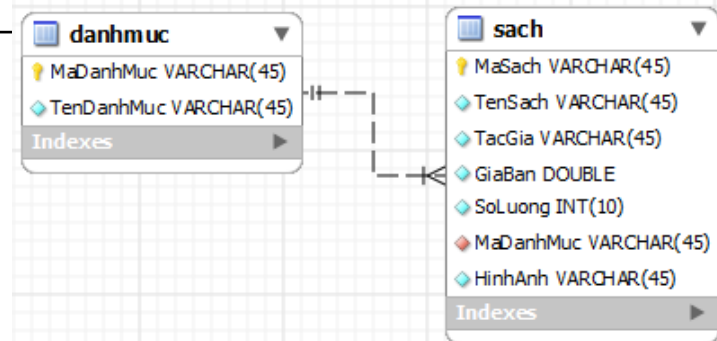
# BookOnline



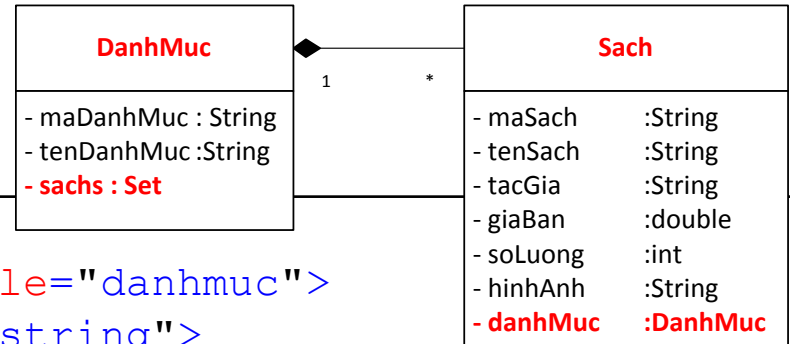
# DanhMuc POJO - SET



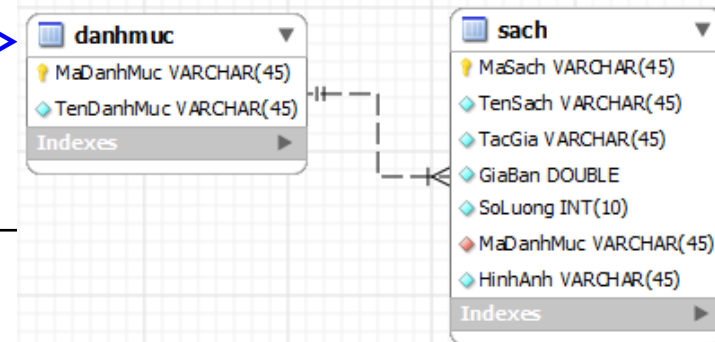
```
1 import java.util.HashSet;
2 import java.util.Set;
3
4 public class DanhMuc implements java.io.Serializable {
5
6     private String maDanhMuc;
7     private String tenDanhMuc;
8     private Set<Sach> sachs = new HashSet<Sach>(0);
9
10    //Các phương thức khởi tạo
11    . . .
12    //Getter & Setter
13    . . .
```



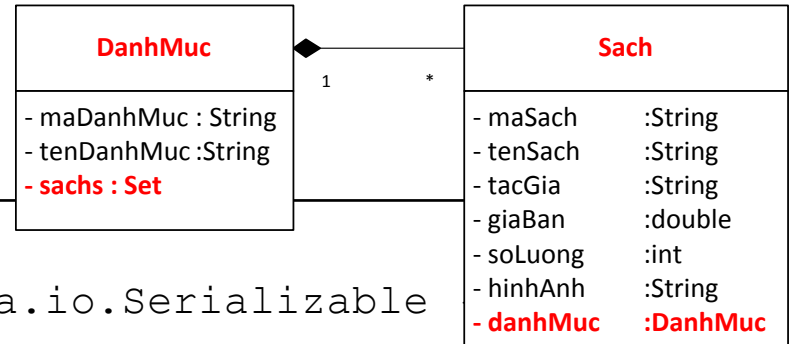
# DanhMuc.hbm.xml - Set



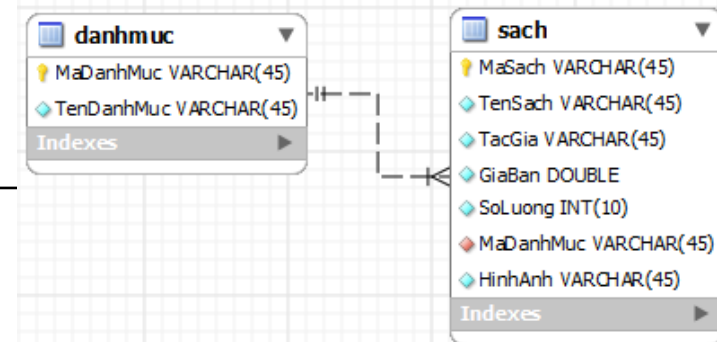
```
1 <hibernate-mapping>
2   <class name="pojo.DanhMuc" table="danhmuc">
3     <id name="maDanhMuc" type="string">
4       <column length="45" name="MaDanhMuc"/>
5       <generator class="assigned"/>
6     </id>
7     <property name="tenDanhMuc" type="string">
8       <column length="45" name="TenDanhMuc" not-null="true"/>
9     </property>
10    <set inverse="true" name="sachs" >
11      <key>
12        <column length="45" name="MaDanhMuc" not-null="true"/>
13      </key>
14      <one-to-many class="pojo.Sach"/>
15    </set>
16  </class>
17 </hibernate-mapping>
```



# Sach POJO



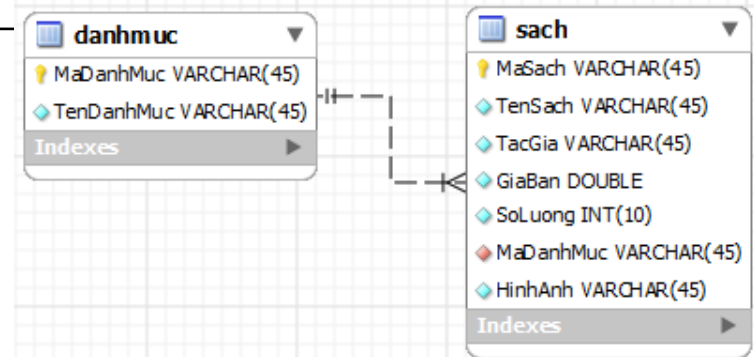
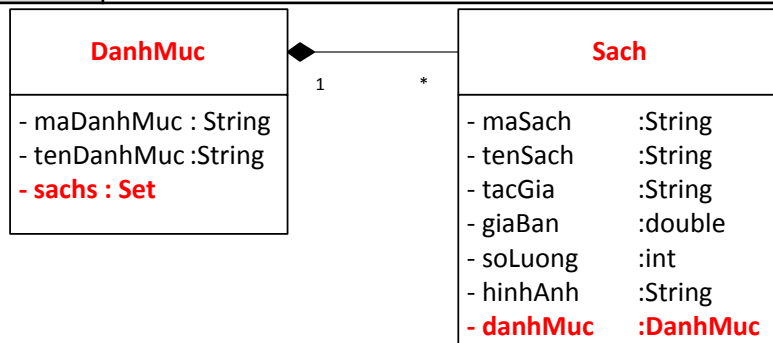
```
1 package pojo;
2 public class Sach implements java.io.Serializable
3     private String maSach;
4     private DanhMuc danhMuc;
5     private String tenSach;
6     private String tacGia;
7     private double giaBan;
8     private int soLuong;
9     private String hinhAnh;
10
11     //Constructors
12
13     //Getters & Setters
14
15
16
```





# Sach.hbm.xml

```
1 <hibernate-mapping>
2   <class catalog="bookonline" name="pojo.Sach" table="sach">
3     <id name="maSach" type="string">
4       <column length="45" name="MaSach"/>
5       <generator class="assigned"/>
6     </id>
7     . . .
8     <many-to-one class="pojo.DanhMuc" name="danhMuc"
9       fetch="select">
10      <column length="45" name="MaDanhMuc" not-null="true"/>
11    </many-to-one>
12  </class>
13 </hibernate-mapping>
```



# DanhMucDAO – lấy thông tin danh mục

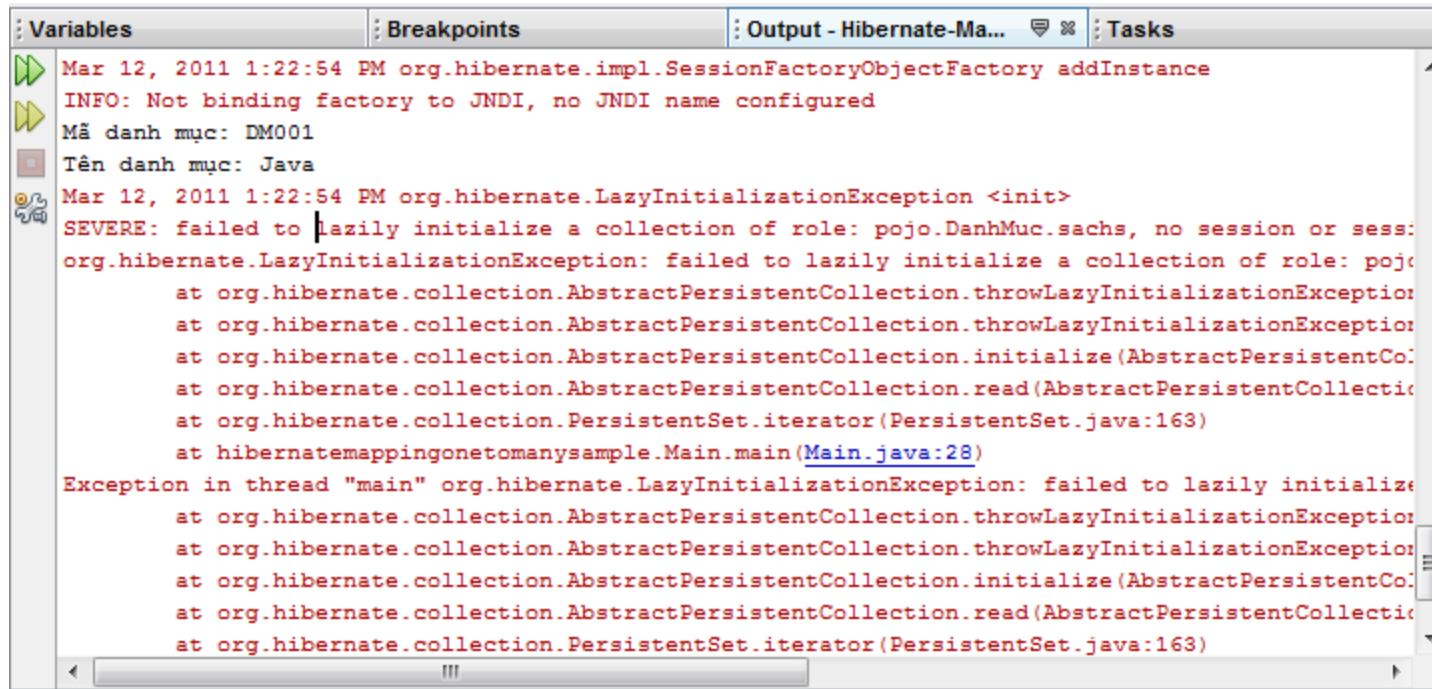
```
1 public class DanhMucDAO {
2     public static DanhMuc layThongTinDanhmuc(String maDanhmuc) {
3         DanhMuc dm = null;
4         Session session = HibernateUtil.getSessionFactory()
5                                 .openSession();
6         try {
7             dm = (DanhMuc)session.get(DanhMuc.class, maDanhmuc);
8         } catch (HibernateException ex) {
9             System.err.println(ex);
10        } finally {
11            session.close();
12        }
13        return dm;
14    }
15 }
```

# DanhMucDAO – lấy thông tin danh mục

```
1 public class Main {
2     public static void main(String[] args) {
3         DanhMuc dm=DanhMucDAO.layThongTinDanhmuc("DM001");
4         System.out.println("Mã danh mục: "+dm.getMaDanhMuc());
5         System.out.println("Tên danh mục: "+dm.getTenDanhMuc());
6         Iterator<Sach> sachs=dm.getSachs().iterator();
7         while(sachs.hasNext()){
8             Sach s=sachs.next();
9             System.out.println("Mã sách: "+ s.getMaSach());
10            System.out.println("Tên sách: "+s.getTenSach());
11            System.out.println("Tác giả: "+s.getTacGia());
12            System.out.println("Giá bán: "+s.getGiaBan());
13            System.out.println("Số lượng: "+s.getSoLuong());
14            System.out.println("Hình ảnh: "+s.getHinhAnh());
15        }
16    }
17 }
```

# DanhMucDAO – lấy thông tin danh mục

Lazy = true



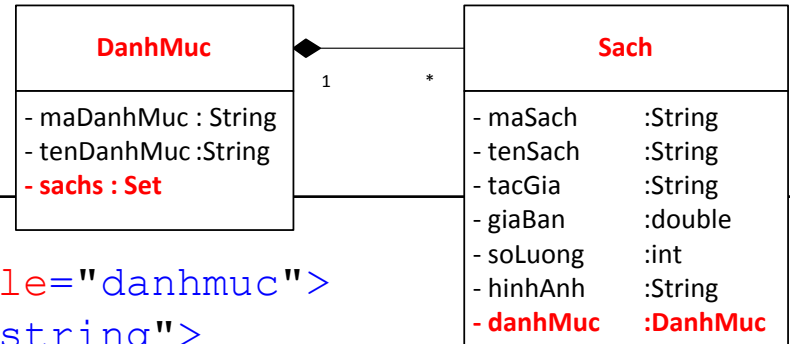
The screenshot shows the Output window of an IDE with the following content:

```
Variables | Breakpoints | Output - Hibernate-Ma... | Tasks
Mar 12, 2011 1:22:54 PM org.hibernate.impl.SessionFactoryObjectFactory addInstance
INFO: Not binding factory to JNDI, no JNDI name configured
Mã danh mục: DM001
Tên danh mục: Java
Mar 12, 2011 1:22:54 PM org.hibernate.LazyInitializationException <init>
SEVERE: failed to lazily initialize a collection of role:.pojo.DanhMuc.sachs, no session or sess:
org.hibernate.LazyInitializationException: failed to lazily initialize a collection of role:.pojo
    at org.hibernate.collection.AbstractPersistentCollection.throwLazyInitializationException
    at org.hibernate.collection.AbstractPersistentCollection.throwLazyInitializationException
    at org.hibernate.collection.AbstractPersistentCollection.initialize(AbstractPersistentCol
    at org.hibernate.collection.AbstractPersistentCollection.read(AbstractPersistentCollectio
    at org.hibernate.collection.PersistentSet.iterator(PersistentSet.java:163)
    at hibernatemappingonetomany.sample.Main.main(Main.java:28)
Exception in thread "main" org.hibernate.LazyInitializationException: failed to lazily initialize
    at org.hibernate.collection.AbstractPersistentCollection.throwLazyInitializationException
    at org.hibernate.collection.AbstractPersistentCollection.throwLazyInitializationException
    at org.hibernate.collection.AbstractPersistentCollection.initialize(AbstractPersistentCol
    at org.hibernate.collection.AbstractPersistentCollection.read(AbstractPersistentCollectio
    at org.hibernate.collection.PersistentSet.iterator(PersistentSet.java:163)
```

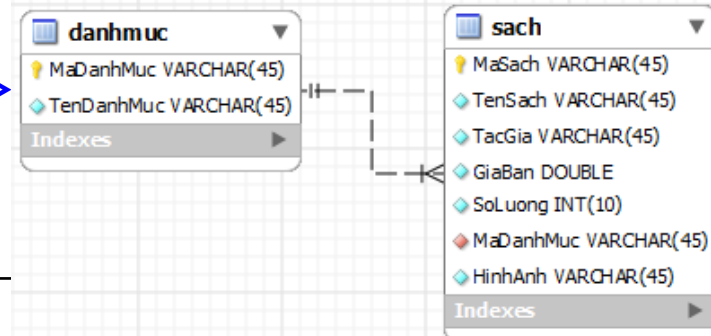
# Lazy Initialization

- Trong Hibernate, **Lazy Initialization** giúp
  - Tránh các câu truy vấn cơ sở dữ liệu không cần thiết
  - Gia tăng hiệu suất thực thi
  - Lazy mặc định có giá trị là **true**

# DanhMuc.hbm.xml - Set



```
1 <hibernate-mapping>
2   <class name="pojo.DanhMuc" table="danhmuc">
3     <id name="maDanhMuc" type="string">
4       <column length="45" name="MaDanhMuc"/>
5       <generator class="assigned"/>
6     </id>
7     <property name="tenDanhMuc" type="string">
8       <column length="45" name="TenDanhMuc" not-null="true"/>
9     </property>
10    <set inverse="true" name="sachs"
11      fetch="join" lazy="false" >
12    <key>
13      <column length="45" name="MaDanhMuc" not-null="true"/>
14    </key>
15    <one-to-many class="pojo.Sach"/>
16  </set>
17 </class>
18 </hibernate-mapping>
```

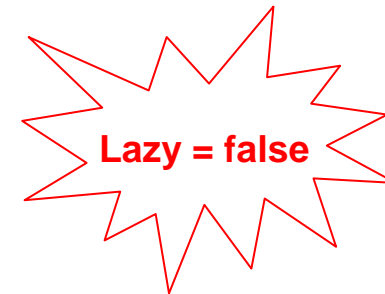


# DanhMucDAO – lấy thông tin danh mục

```
1  public class Main {
2      public static void main(String[] args) {
3          DanhMuc dm=DanhMucDAO.layThongTinDanhmuc("DM001");
4          System.out.println("Mã danh mục: "+dm.getMaDanhMuc());
5          System.out.println("Tên danh mục: "+dm.getTenDanhMuc());
6          Iterator<Sach> sachs=dm.getSachs().iterator();
7          while(sachs.hasNext()){
8              Sach s=sachs.next();
9              System.out.println("Mã sách: "+ s.getMaSach());
10             System.out.println("Tên sách: "+s.getTenSach());
11             System.out.println("Tác giả: "+s.getTacGia());
12             System.out.println("Giá bán: "+s.getGiaBan());
13             System.out.println("Số lượng: "+s.getSoLuong());
14             System.out.println("Hình ảnh: "+s.getHinhAnh());
15         }
16     }
17 }
```

# DanhMucDAO – lấy thông tin danh mục

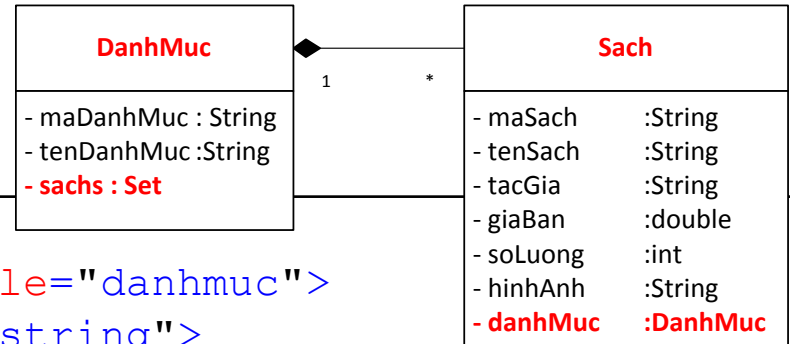
Variables		Breakpoints	
Name	Type	Value	
dm	DanhMuc	#1411	
maDanhMuc	String	"DM001"	
tenDanhMuc	String	"Java"	
sachs	PersistentSet	"size = 5"	
[0]	Sach	#1430	
[1]	Sach	#1431	
[2]	Sach	#1432	
[3]	Sach	#1433	
[4]	Sach	#1434	



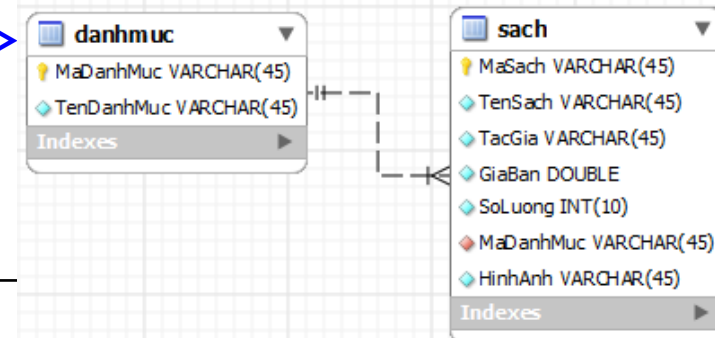
Variables		Breakpoints	
Mã danh mục:	DM001		
Tên danh mục:	Java		
Mã sách:	S002		
Tên sách:	Java 2		
Tác giả:	Nguyễn Văn Sơn		
Giá bán:	200000.0		
Số lượng:	300		
Hình ảnh:	images/2.jpg		
Mã sách:	S003		
Tên sách:	Java 3		
Tác giả:	Nguyễn Thị Buổi		
Giá bán:	150000.0		
Số lượng:	150		
Hình ảnh:	images/3.jpg		
Mã sách:	S005		
Tên sách:	Java 5		
Tác giả:	Nguyễn Thành Trung		
Giá bán:	220000.0		
Số lượng:	310		



# DanhMuc.hbm.xml - Set



```
1 <hibernate-mapping>
2   <class name="pojo.DanhMuc" table="danhmuc">
3     <id name="maDanhMuc" type="string">
4       <column length="45" name="MaDanhMuc"/>
5       <generator class="assigned"/>
6     </id>
7     <property name="tenDanhMuc" type="string">
8       <column length="45" name="TenDanhMuc" not-null="true"/>
9     </property>
10    <set inverse="true" name="sachs" >
11      <key>
12        <column length="45" name="MaDanhMuc" not-null="true"/>
13      </key>
14      <one-to-many class="pojo.Sach"/>
15    </set>
16  </class>
17 </hibernate-mapping>
```



# DanhMucDAO – lấy thông tin danh mục

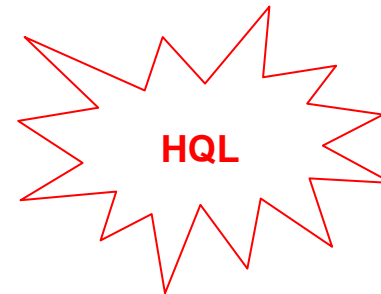
```
1 public class DanhMucDAO {
2     public static DanhMuc layThongTinDanhMuc(String maDanhMuc) {
3         DanhMuc dm = null;
4         Session session = HibernateUtil.getSessionFactory()
5                                     .openSession();
6         try {
7             String hql=" select dm ";
8             hql+= " from DanhMuc dm left join fetch dm.sachs";
9             hql+= " where dm.maDanhMuc=:maDanhMuc";
10            Query query=session.createQuery(hql);
11            query.setString("maDanhMuc", maDanhMuc);
12            dm=(DanhMuc)query.uniqueResult();
13        } catch (HibernateException ex) {
14            System.err.println(ex);
15        } finally {
16            session.close();
17        }
18        return dm;
19    }
```

# DanhMucDAO – lấy thông tin danh mục

```
1  public class Main {
2      public static void main(String[] args) {
3          DanhMuc dm=DanhMucDAO.layThongTinDanhmuc("DM001");
4          System.out.println("Mã danh mục: "+dm.getMaDanhMuc());
5          System.out.println("Tên danh mục: "+dm.getTenDanhMuc());
6          Iterator<Sach> sachs=dm.getSachs().iterator();
7          while(sachs.hasNext()){
8              Sach s=sachs.next();
9              System.out.println("Mã sách: "+ s.getMaSach());
10             System.out.println("Tên sách: "+s.getTenSach());
11             System.out.println("Tác giả: "+s.getTacGia());
12             System.out.println("Giá bán: "+s.getGiaBan());
13             System.out.println("Số lượng: "+s.getSoLuong());
14             System.out.println("Hình ảnh: "+s.getHinhAnh());
15         }
16     }
17 }
```

# DanhMucDAO – lấy thông tin danh mục

Variables		Breakpoints	
Name	Type	Value	
dm	DanhMuc	#1411	
maDanhMuc	String	"DM001"	
tenDanhMuc	String	"Java"	
sachs	PersistentSet	"size = 5"	
[0]	Sach	#1430	
[1]	Sach	#1431	
[2]	Sach	#1432	
[3]	Sach	#1433	
[4]	Sach	#1434	



Variables		Breakpoints	
Mã danh mục:	DM001		
Tên danh mục:	Java		
Mã sách:	S002		
Tên sách:	Java 2		
Tác giả:	Nguyễn Văn Sơn		
Giá bán:	200000.0		
Số lượng:	300		
Hình ảnh:	images/2.jpg		
Mã sách:	S003		
Tên sách:	Java 3		
Tác giả:	Nguyễn Thị Buổi		
Giá bán:	150000.0		
Số lượng:	150		
Hình ảnh:	images/3.jpg		
Mã sách:	S005		
Tên sách:	Java 5		
Tác giả:	Nguyễn Thành Trung		
Giá bán:	220000.0		
Số lượng:	310		

# Cascade

- save-update
- delete
- delete-orphan

## Tài liệu tham khảo

- Nguyễn Hoàng Anh, Tập bài giảng và video môn chuyên đề Java, 2010
- Gary Mak, Tập hướng dẫn từng bước Hibernate, 2006



# HỎI VÀ ĐÁP