

Bài 6: Giới thiệu Hibernate Framework

Nội dung bài học

- Hibernate Framework là gì?
- Hibernate vs JDBC
- Vòng đời của Hibernate
- Các bước sử dụng Hibernate
- Cấu hình Hibernate

Ý tưởng

- Cơ sở dữ liệu thường được thiết kế và lưu trữ theo hướng quan hệ.
- Tuy nhiên phần mềm thường được xây dựng theo hướng đối tượng.
- Đối với lập trình viên khi xây dựng phần mềm thường muốn làm việc với các đối tượng và không phải nhớ đến các dòng, các cột trong các bảng của cơ sở dữ liệu
- JDBC xử lý các CSDL lớn rất chậm (từ 1000 bảng trở lên), đặc biệt nếu ứng dụng có khả năng kết nối với nhiều hệ quản trị khác nhau, câu truy vấn trong mỗi hệ quản trị có thể có cú pháp khác
- → Khó vận hành, bảo trì và xử lý sự cố với các CSDL lớn

Hibernate Framework là gì?

- Hibernate được phát triển bởi Gavin King từ năm 2001, là một ORM framework thuần Java
- Hibernate là một phần mềm miễn phí
- Hibernate giúp lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh, cho phép bạn truy vấn dữ liệu bằng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc bằng SQL thuần.

Hibernate Framework là gì?

- **Các thành phần của Hibernate project** (các thành phần thư viện Hibernate).
 - Hibernate Core: Cung cấp các chức năng cơ bản của persistent layer cho các ứng dụng java với các APIs và hỗ trợ XML Mapping metadata.
 - Hibernate Annotations: các map class với JDK 5.0 Annotations, bao gồm Hibernate Validator.
 - Hibernate EntityManager : sử dụng EJB 3.0 API trong JSE hoặc với bất kỳ JEE nào.

Hibernate Framework là gì?

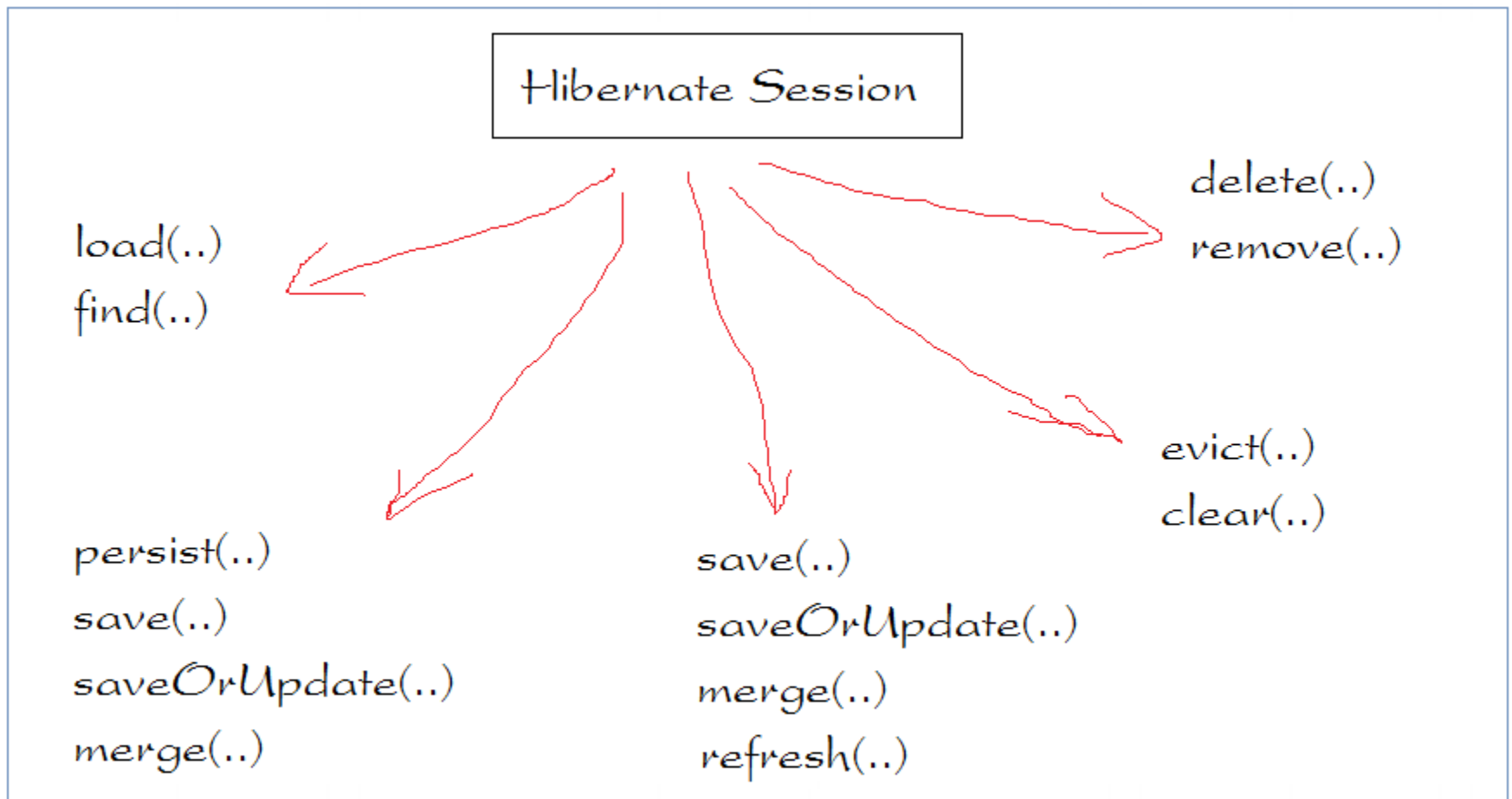
- Hibernate Tools : các tool tích hợp với Eclipse và Ant giúp tạo ra các persistent object từ 1 schema có sẵn trong database(reverse-engineering) và từ các file hbm sinh ra các class java thực hiện các persistent object rồi Hibernate tự tạo tác object trong database (forward-engineering).
- NHibernate : Hibernate cho .NET Framework.
- Jboss Seam : Một Java EE 5.0 framework cho phát triển các ứng dụng JSF, Ajax và EJB 3.0 với sự hỗ trợ của Hibernate. Seam hiện rất mới và tỏ ra rất mạnh để phát triển các ứng dụng Web 2.0. Nó tích hợp đầy đủ tất cả các công nghệ "hot" nhất hiện nay.

So sánh 2 giải pháp JDBC và Hibernate

JDBC	Hibernate
Với JDBC, lập trình viên phải viết mã để lập bản đồ dữ liệu của một mô hình đối tượng sang dữ liệu của một mô hình quan hệ và lược đồ cơ sở dữ liệu tương ứng của nó	Hibernate là giải pháp ORM linh hoạt và mạnh mẽ để lập bản đồ các lớp Java vào các bảng cơ sở dữ liệu. Hibernate tự quản lý bản đồ này bằng cách sử dụng các tập tin XML, vì vậy lập trình viên không cần phải viết mã cho điều này.
Với JDBC, các bản đồ của các đối tượng Java với các bảng cơ sở dữ liệu và ngược lại được quản lý bởi các lập trình viên bằng tay	Hibernate cung cấp một hệ thống transparent persistence.
JDBC chỉ hỗ trợ native Structured Query Language (SQL). Lập trình viên phải tự tìm ra cách phù hợp để truy cập cơ sở dữ liệu	Hibernate cung cấp HQL mở rộng (không phụ thuộc vào kiểu cơ sở dữ liệu) đồng thời vẫn hỗ trợ SQL
Ứng dụng sử dụng JDBC để xử lý các dữ liệu liên tục Nếu bảng cơ sở dữ liệu thay đổi thì lập trình viên phải viết lại mã	Hibernate cung cấp bản đồ tự động. Nếu có sự thay đổi trong cơ sở dữ liệu thì chỉ cần sửa file XML
Với JDBC, bộ đệm được quản lý bằng mã viết tay	Hibernate, với Transparent Persistence, bộ đệm được đặt vào không gian làm việc của ứng dụng. Điều này nâng cao hiệu năng của ứng dụng nếu ứng dụng đọc cùng một dữ liệu nhiều lần

Vòng đời của Hibernate

- Class **Session** của **Hibernate** có một số các method quan trọng. Chúng được chia ra các nhóm như hình minh họa dưới đây.



Vòng đời của Hibernate

- Một đối tượng trong **Hibernate** có 1 trong 4 trạng thái:
Transient (Tạm thời)
- **Persistent** (Bền vững)
- **Removed** (Đã bị xóa - dưới DB)
- **Detached** (Đã bị tách riêng ra - so với session hiện tại)

Vòng đời của Hibernate

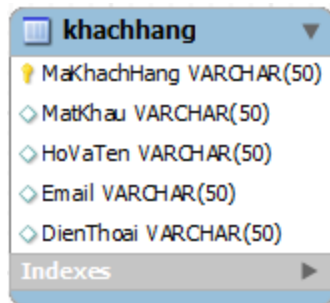
- Trường hợp bạn tạo mới một đối tượng java từ một Entity, đối tượng đó có tình trạng là Transient. Hibernate không biết về sự tồn tại của nó. Nó nằm ngoài sự quản lý của Hibernate.
- Trường hợp bạn lấy ra đối tượng Entity bằng method get, load hoặc find, bạn có được một đối tượng nó tương ứng với 1 record dưới database. Đối tượng này có trạng thái Persistent. Nó được quản lý bởi Hibernate.
- Session gọi một trong các method save, saveOrUpdate, persist, merge sẽ đẩy đối tượng Transient vào sự quản lý của Hibernate và đối tượng này chuyển sang trạng thái Persistent. Tùy tình huống nó sẽ insert hoặc update dữ liệu vào DB.
- Session gọi evict(..) hoặc clear() để đuổi các đối tượng có trạng thái persistent (bền vững) ra khỏi sự quản lý của Hibernate, giờ các đối tượng này sẽ có trạng thái mới là Detached (Bị tách ra). Nếu nó không được đính (Attached) trở lại, nó sẽ bị bộ gom rác của Java quét đi theo cơ chế thông thường.
- Sử dụng update(..), saveOrUpdate(..), merge(..) sẽ đính trở lại các đối tượng Detached vào lại. Tùy tình huống nó sẽ tạo ra dưới DB câu lệnh update hoặc insert. Các đối tượng sẽ trở về trạng thái Persistent (bền vững).
- Session gọi method remove(..), delete(..) để xóa một bản ghi, đối tượng persistent giờ chuyển sang trạng thái Removed (Đã bị xóa).

Các bước sử dụng hibernate

- Bước 1: Tạo cơ sở dữ liệu
- Bước 2: Tạo các POJO
- Bước 3: Tạo file cấu hình hibernate.cfg.xml
- Bước 4: Tạo các file mapping <POJO>.hbm.xml
- Bước 5: Khai báo các file mapping vào hibernate.cfg.xml
- Bước 6: Xây dựng lớp HibernateUtil
- Bước 7: Xây dựng các DAO & Sử dụng

Ví Dụ Minh Họa

Chuẩn bị cơ sở dữ liệu



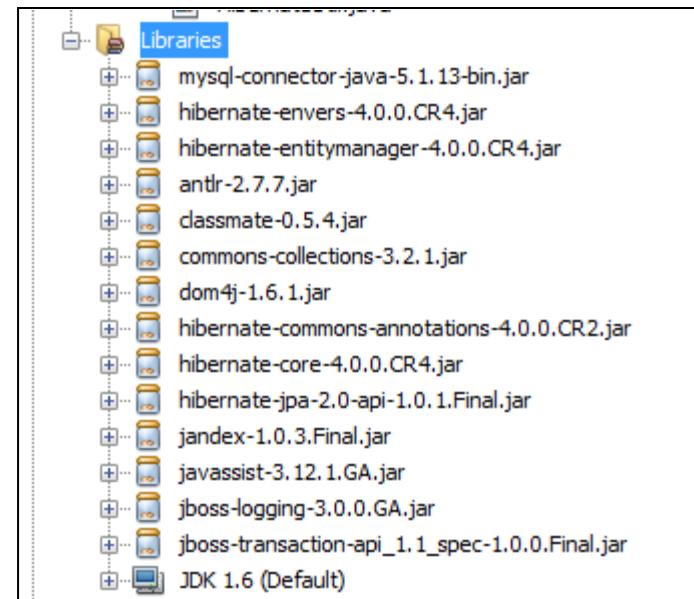
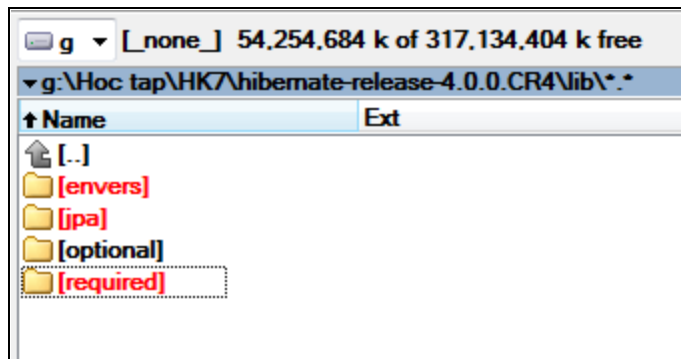
khachhang	
MaKhachHang	VARCHAR(50)
MatKhai	VARCHAR(50)
HoVaTen	VARCHAR(50)
Email	VARCHAR(50)
DienThoai	VARCHAR(50)
Indexes	

Table Khachhang

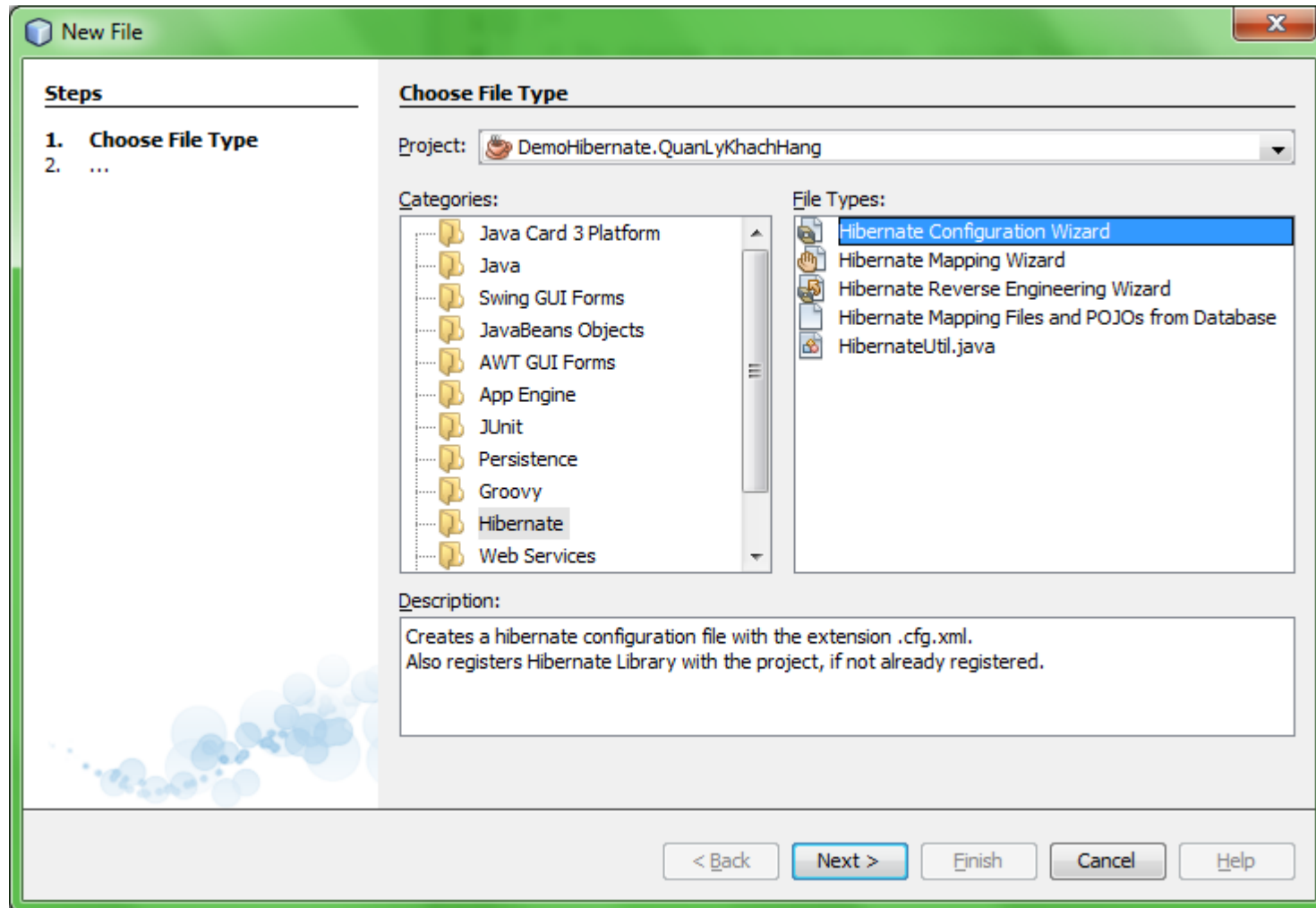
	MaKhachHang	MatKhai	HoVaTen	Email	DienThoai
▶	KH01	123456	Hồ Văn Tấn	tanhv90@gmail.com	0974447031
	KH02	123123	Sử Bá Thuần	thuanbsf@gmail.com	0973560943
	KH03	123123	Nguyễn Văn Cù	ngv@gmail.com	0973560112
*	NULL	NULL	NULL	NULL	NULL

Bước 1: Add thư viện hibernate

- Link download thư viện:
<http://nchc.dl.sourceforge.net/project/hibernate/hibernate4/4.0.0.CR4/hibernate-release-4.0.0.CR4.zip>
- Các thư mục chứa file thư viện cần add vào project:
envers, **jpa**, **required** (thư mục optional chứa các file thư viện tùy chọn)



Bước 2: Tạo file cấu hình hibernate.cfg.xml



Bước 2: Tạo file cấu hình hibernate.cfg.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
3 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
4 <hibernate-configuration>
5   <session-factory>
6     <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
7     <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
8     <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/quanlysach</property>
9     <property name="hibernate.connection.username">root</property>
10    <property name="hibernate.connection.password">password</property>
11    <mapping resource="entity/Khachhang.hbm.xml"/>
12  </session-factory>
13 </hibernate-configuration>
```


Bước 3: Tạo POJO KháchHang.

```
1 public class Khachhang implements java.io.Serializable {
2
3     private String maKhachHang;
4     private String matKhai;
5     private String hoVaTen;
6     private String email;
7     private String dienThoai;
8
9     //Các phương thức khởi tạo, get, set.
10 }
11
12
13
14
15
16
17
```

Bước 4: Tạo file mapping Khachhang.hbm.xml

```
1  <?xml version="1.0"?>
2  <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
3  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
4  <!-- Generated Oct 14, 2011 4:33:03 PM by Hibernate Tools 3.2.1.GA -->
5  <hibernate-mapping>
6      <class name="entity.Khachhang" table="khachhang" catalog="quanlysach">
7          <id name="maKhachHang" type="string">
8              <column name="MaKhachHang" length="50" />
9              <generator class="assigned" />
10         </id>
11         <property name="matKhau" type="string">
12             <column name="MatKhau" length="50" />
13         </property>
14         <property name="hoVaTen" type="string">
15             <column name="HoVaTen" length="50" />
16         </property>
17         <property name="email" type="string">
18             <column name="Email" length="50" />
19         </property>
20         <property name="dienThoai" type="string">
21             <column name="DienThoai" length="50" />
22         </property>
23     </class>
24 </hibernate-mapping>
```

Bước 4: Tạo file mapping Khachhang.hbm.xml

Tên class trong POJO: `name="entity.Khachhang"`

Tên bảng trong database: `table="khachhang"`

Tên database `catalog="quanlysach"`

Tên thuộc tính trong POJO: `<property name="matKhau" type="string">`

Tên cột trong Database: `<column name="MatKhau" length="50" />`

Chú ý

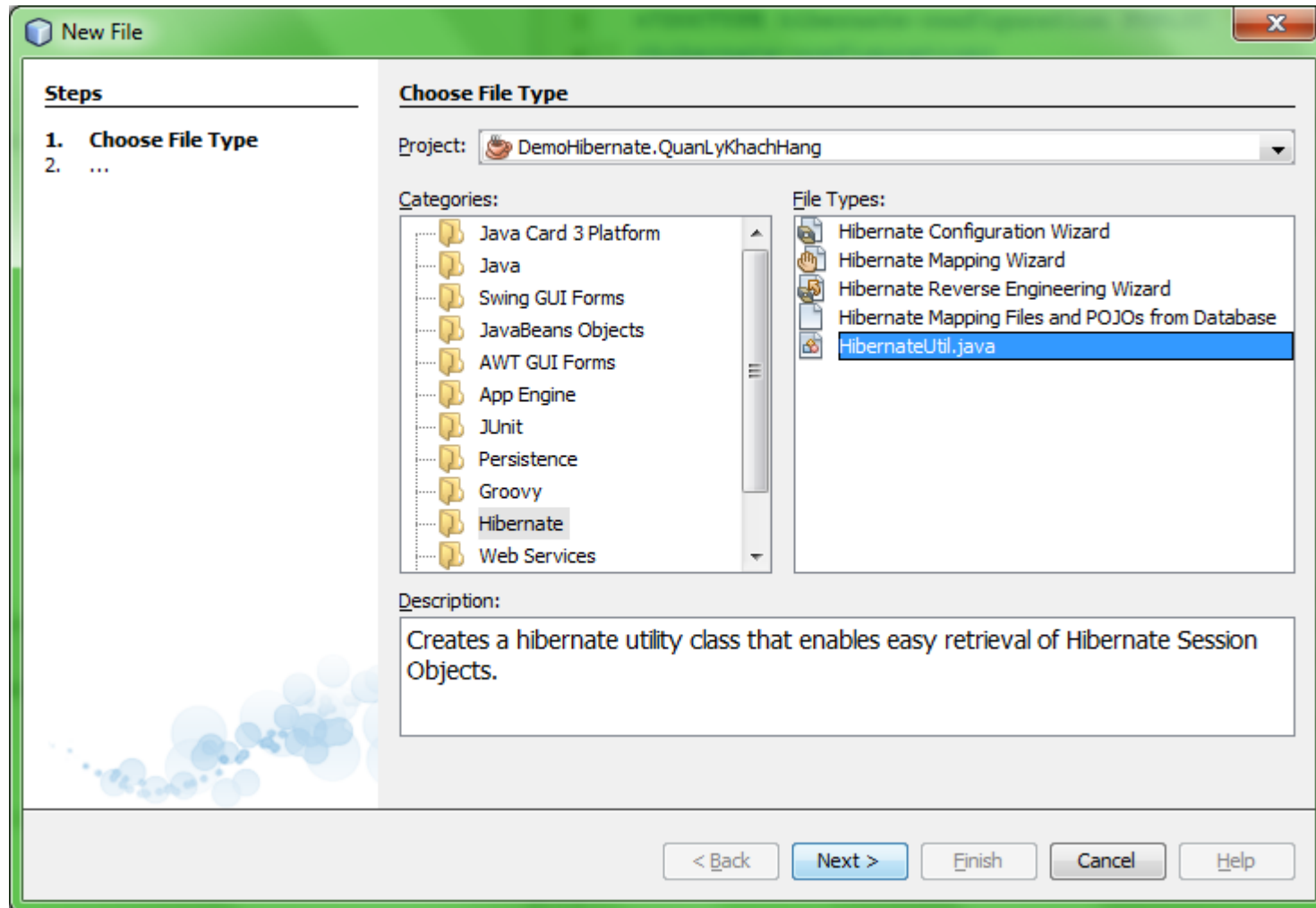
- Tên class, thuộc tính: phải đúng theo tên class, thuộc tính trong POJO, có phân biệt hoa thường.
- Tên table, column: đúng theo tên table, cột trong cơ sở dữ liệu, không phân biệt hoa thường.
- Có thể lược bỏ tên table, column → tên table, column sẽ được lấy từ tên class, thuộc tính.
- Có thể lược bỏ length.
- Có thể lược bỏ type nhưng sẽ làm giảm hiệu suất → không nên dùng.

Bước 5: Khai báo file mapping vào hibernate.cfg.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
3 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
4 <hibernate-configuration>
5   <session-factory>
6     <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
7     <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
8     <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/quanlysach</property>
9     <property name="hibernate.connection.username">root</property>
10    <property name="hibernate.connection.password">password</property>
11    <property name="hibernate.current_session_context_class">thread</property>
12    <mapping resource="entity/Khachhang.hbm.xml"/>
13  </session-factory>
</hibernate-configuration>
```

Chú ý: “**entity/Khachhang.hbm.xml**” đường dẫn là dấu /
Không được dùng dấu chấm “**Entity.Khachhang.hbm.xml**”

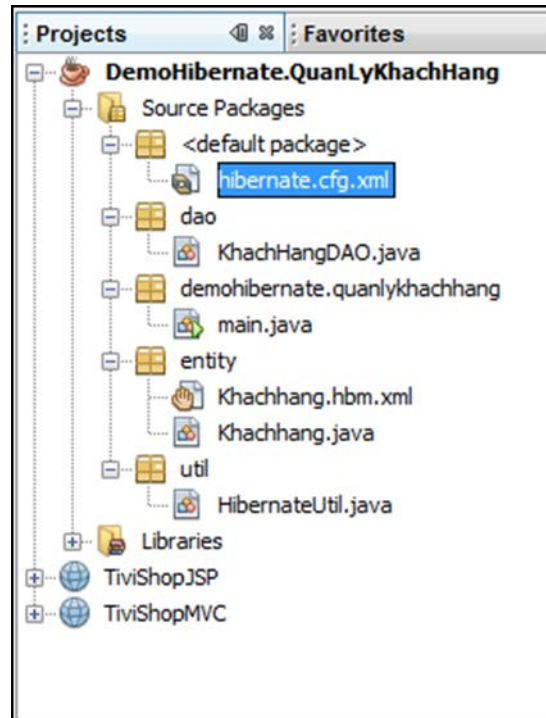
Bước 6. Xây dựng lớp HibernateUtil



Bước 6. Xây dựng lớp HibernateUtil

```
1 package util;
2
3 import org.hibernate.cfg.AnnotationConfiguration;
4 import org.hibernate.SessionFactory;
5
6 public class HibernateUtil {
7
8     private static final SessionFactory sessionFactory;
9
10    static {
11        try {
12            sessionFactory = new AnnotationConfiguration()
13                               .configure().buildSessionFactory();
14        } catch (Throwable ex) {
15            System.err.println("Initial SessionFactory creation failed." + ex);
16            throw new ExceptionInInitializerError(ex);
17        }
18    }
19    public static SessionFactory getSessionFactory() {
20        return sessionFactory;
21    }
22 }
23
24
25
```

Cấu trúc thư mục đã tạo



Bước 7. Xây dựng lớp Khách Hàng DAO & sử dụng

- Lấy danh sách khách hàng
- Lấy thông tin khách hàng
- Thêm thông tin khách hàng
- Cập nhật thông tin khách hàng
- Xóa thông tin khách hàng

DAO: Lấy danh sách khách hàng

```
1 import java.util.List;
2 import org.hibernate.Query;
3 import org.hibernate.Session;
4 import util.HibernateUtil;
5
6 public class KhachHangDAO {
7     public static List<Khachhang> layDanhSachKhachHang()
8     {
9         List<Khachhang> dsKhachHang = null;
10        Session session = HibernateUtil.getSessionFactory().getCurrentSession();
11        session.beginTransaction();
12        String hql = "from Khachhang";
13        Query query = session.createQuery(hql);
14        dsKhachHang = query.list();
15        return dsKhachHang;
16    }
17 }
```

DAO: Lấy danh sách khách hàng

Chú ý:

- Sử dụng `getCurrentSession()` → không cần phải `open session` và `close session` sau khi sử dụng xong. Tuy nhiên phải khai báo trong file config:

`<property name="hibernate.current_session_context_class">thread</property>`

```
1 <session-factory>
2     <property
3 name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
4     <property
5 name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
6     <property name="hibernate.connection.url">
7 jdbc:mysql://localhost:3306/quanlysach?useUnicode=true&characterEncoding=UTF-8
8     </property>
9     <property name="hibernate.connection.username">root</property>
10    <property name="hibernate.connection.password">password</property>
11    <property name="hibernate.current_session_context_class">thread</property>
12    <mapping resource="entity/Khachhang.hbm.xml"/>
13    </session-factory>
```

- Sử dụng `HibernateUtil.getSessionFactory().openSession();`
→ phải close session sau khi sử dụng xong. `Session.Close();`

Hiển thị danh sách khách hàng.

```
1 public class main {  
2     public static void main(String[] args) {  
3         List<Khachhang> dsKhachHang = KhachHangDAO.layDanhSachKhachHang();  
4         for (int i = 0; i < dsKhachHang.size(); i++) {  
5             System.out.println("Họ tên: " + dsKhachHang.get(i).getHoVaTen()  
6                 + " - Email: " + dsKhachHang.get(i).getEmail()  
7                 + " - Số điện thoại: " + dsKhachHang.get(i).getDienThoai());  
8         }  
9     }  
10 }
```

Kết quả

```
Họ tên: Hồ Văn Tấn - Email: tanhv90@gmail.com - Số điện thoại: 0974447031  
Họ tên: Sử Bá Thuận - Email: thuanbsf@gmail.com - Số điện thoại: 0973560943  
Họ tên: Nguyễn Văn Cù - Email: ngv@gmail.com - Số điện thoại: 0973560112  
BUILD SUCCESSFUL (total time: 2 seconds)
```

DAO: Lấy thông tin khách hàng

```
1 public static Khachhang layThongTinKhachHang(String maKhachHang) {  
2     Session session = HibernateUtil.getSessionFactory().openSession();  
3     session.beginTransaction();  
4  
5     Khachhang kh = (Khachhang) session.get(Khachhang.class, maKhachHang);  
6  
7     session.close();  
8     return kh;  
9 }  
10
```

- Sử dụng `HibernateUtil.getSessionFactory().openSession();`
→ close session sau khi sử dụng xong. `Session.Close();`

Hiển thị thông tin khách hàng

```
1 public class main {  
2  
3     public static void main(String[] args) {  
4  
5         Khachhang kh = KhachHangDAO.layThongTinKhachHang("KH01");  
6         System.out.println("Họ tên: " + kh.getHoVaTen());  
7         System.out.println("Email: " + kh.getEmail());  
8         System.out.println("Số Báo Thuần: " + kh.getDienThoai());  
9     }  
10 }
```

Kết quả

```
Họ tên: HỒ VĂN TÂN  
Email: tanhv90@gmail.com  
Số điện thoại: 0974447031  
BUILD SUCCESSFUL (total time: 2 seconds)
```

→ DAO: Thêm khách hàng.

```
1 public static boolean themKhachHang(Khachhang kh) {
2     if(KhachHangDAO.layThongTinKhachHang(kh.getMaKhachHang()) != null)
3         return false;
4     Session session = HibernateUtil.getSessionFactory().openSession();
5     try {
6         session.beginTransaction();
7         session.save(kh);
8         session.getTransaction().commit();
9         return true;
10    } catch (Exception e) {
        session.getTransaction().rollback();
        System.out.println(e);
        return false;
    } finally {
        session.close();
    }
}
```

→ Thêm khách hàng.

```
1 public class main {  
2  
3     public static void main(String[] args) {  
4  
5         Khachhang kh = new Khachhang("KH05", "123123", "Trần Trọng Vĩnh",  
6                                     "ttv@gmail.com", "097654321");  
7         if(KhachHangDAO.themKhachHang(kh) == true)  
8             System.out.println("Thêm thành công!");  
9         else  
10            System.out.println("Thêm không thành công!");  
  
    }  
}
```

Kết quả

	MaKhachHang	MatKau	HoVaTen	Email	DienThoai
	KH01	123456	Hồ Văn Tấn	tanhv90@gmail.com	0974447031
	KH02	123123	Sử Bá Thuận	thuanbsf@gmail.com	0973560943
	KH03	123123	Nguyễn Văn Cừ	ngv@gmail.com	0973560112
	KH04	123123	Lê Văn Tám	lvt@gmail.com	097654321
▶	KH05	123123	Trần Trọng Vĩnh	ttv@gmail.com	097654321
*	NULL	NULL	NULL	NULL	NULL

→ Không lưu được tiếng việt

→ Thêm đoạn sau vào phần cấu hình CSDL trong file config Hibernate
`?useUnicode=true&characterEncoding=UTF-8`

```
1 <session-factory>
2     <property
3 name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
4     <property
5 name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
6     <property name="hibernate.connection.url">
7
8 jdbc:mysql://localhost:3306/quanlysach?useUnicode=true&characterEncoding=UTF-8
9     </property>
10    <property name="hibernate.connection.username">root</property>
13    <property name="hibernate.connection.password">password</property>
14    <property name="hibernate.current_session_context_class">thread</property>
15    <mapping resource="entity/Khachhang.hbm.xml"/>
16 </session-factory>
```


→ DAO: Cập nhật thông tin khách hàng.

```
1 public static boolean capNhatThongTinKhachHang(Khachhang kh)
2     {
3         if(KhachHangDAO.layThongTinKhachHang(kh.getMaKhachHang()) == null)
4             return false;
5         Session session=HibernateUtil.getSessionFactory().getCurrentSession();
6         try {
7             session.getTransaction().begin();
8             session.update(kh);
9             session.getTransaction().commit();
10            return true;
11        } catch (Exception e) {
12            session.getTransaction().rollback();
13            System.out.print(e);
14            return false;
15        }
16    }
```

→ Cập nhật thông tin khách hàng.

```
1 public class main {  
2  
3     public static void main(String[] args) {  
4         Khachhang kh = new Khachhang("KH05", "111111", "Lê Hồng Phong",  
5             "ttv@gmail.com", "097654321");  
6         if (KhachHangDAO.capNhatThongTinKhachHang(kh) == true) {  
7             System.out.println("Cập nhật thành công!");  
8         } else {  
9             System.out.println("Cập nhật thất bại!");  
10        }  
11    }  
12 }
```

Kết quả

```
Cập nhật thành công!  
BUILD SUCCESSFUL (total time: 2 seconds)
```

	KH02	123123	Sử Bá Thuận	thuanbsf@gmail.com	0973560943
	KH03	123123	Nguyễn Văn Cừ	ngv@gmail.com	0973560112
	KH04	123123	Lê Văn Tám	lvt@gmail.com	097654321
▶	KH05	111111	Lê Hồng Phong	ttv@gmail.com	097654321

→ DAO: Xóa một khách hàng.

```
1 public static boolean xoaKhachHang(String maKhachHang)
2 {
3     Khachhang kh = KhachHangDAO.layThongTinKhachHang(maKhachHang);
4     if(kh == null)
5         return false;
6
7     Session session=HibernateUtil.getSessionFactory().getCurrentSession();
8     try {
9         session.getTransaction().begin();
10        session.delete(kh);
11        session.getTransaction().commit();
12        return true;
13    } catch (Exception e) {
14        session.getTransaction().rollback();
15        System.out.print(e);
16        return false;
17    }
18 }
19
20
```

→ Xóa một khách hàng.

```
1 public class main {  
2  
3     public static void main(String[] args) {  
4         if (KhachHangDAO.xoaKhachHang("KH05") == true) {  
5             System.out.println("Xóa thành công!");  
6         } else {  
7             System.out.println("Xóa thất bại!");  
8         }  
9     }  
10 }
```

Kết quả

Xóa thành công!

BUILD SUCCESSFUL (total time: 2 seconds)

	MaKhachHang	MatKhai	HoVaTen	Email	DienThoai
▶	KH01	123456	Hồ Văn Tấn	tanhv90@gmail.com	0974447031
	KH02	123123	Sử Bá Thuận	thuanbsf@gmail.com	0973560943
	KH03	123123	Nguyễn Văn Cừ	ngv@gmail.com	0973560112
	KH04	123123	Lê Văn Tám	lvt@gmail.com	097654321
*	NULL	NULL	NULL	NULL	NULL

Giới thiệu

- Hibernate được thiết kế để hoạt động với nhiều môi trường khác nhau → vì thế nó cần được cấu hình khi sử dụng.

Cấu Hình Cơ Bản

- Cấu hình cơ sở dữ liệu.
- Khai báo các file mapping .hbm.xml

Cấu hình cơ sở dữ liệu

```
1 <hibernate-configuration>
2     <session-factory>
3         .....
4     </session-factory>
5 </hibernate-configuration>
```

Cấu hình cơ sở dữ liệu

```
1 <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
2 <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
3 <property name="hibernate.connection.url">
4 '    jdbc:mysql://localhost:3306/quanlysach?useUnicode=true&characterEncoding=UTF-8
5 </property>
6 <property name="hibernate.connection.username">root</property>
7 <property name="hibernate.connection.password">password</property>
```

- **hibernate.dialect** : loại cơ sở dữ liệu được sử dụng.
- **hibernate.connection.driver_class**: driver được sử dụng.
- **Hibernate.connection.url**: url cơ sở dữ liệu.

Sử dụng **?useUnicode=true&characterEncoding=UTF-8** để có thể lưu dữ liệu unicode xuống cơ sở dữ liệu.

- **hibernate.connection.username**: username.
- **hibernate.connection.password**: password.
- **hibernate.connection.pool_size**: số lượng kết nối tối đa tới CSDL tại một thời điểm.

Cấu hình cơ sở dữ liệu

hibernate.dialect : loại cơ sở dữ liệu được sử dụng.

Ví dụ: `org.hibernate.dialect.MySQLDialect` ⇔ MySQL

Tên cơ sở dữ liệu	Tên property
DB2	<code>org.hibernate.dialect.DB2Dialect</code>
HypersonicSQL	<code>org.hibernate.dialect.HSQLDialect</code>
Infomix	<code>org.hibernate.dialect.InformixDialect</code>
Ingres	<code>org.hibernate.dialect.IngresDialect</code>
Interbase	<code>org.hibernate.dialect.InterbaseDialect</code>
MySQL	<code>org.hibernate.dialect.MySQLDialect</code>
Oracle (any version)	<code>org.hibernate.dialect.OracleDialect</code>
Oracle 9	<code>org.hibernate.dialect.Oracle9Dialect</code>

Khai báo file mapping

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
3 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
4 <hibernate-configuration>
5     <session-factory>
6         <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
7         <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
8         <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/quanlysach</property>
9         <property name="hibernate.connection.username">root</property>
10        <property name="hibernate.connection.password">password</property>
11        <mapping resource="entity/Khachhang.hbm.xml"/>
12    </session-factory>
13 </hibernate-configuration>
```

Chú ý: “**entity/Khachhang.hbm.xml**” đường dẫn là dấu /
Không được dùng dấu chấm “**Entity.Khachhang.hbm.xml**”

Các Cấu Hình Tùy Chọn Khác

Ngoài các cấu hình cơ bản, việc cấu hình hibernate còn rất nhiều tùy chọn khác để điều khiển cách thức hoạt động của hibernate. Thông thường các thuộc tính này nếu không được khai báo sẽ có giá trị mặc định.

- Các thuộc tính về Configuration
- JDBC và các thuộc tính về Connection.
- Các thuộc tính về Transaction
- Các thuộc tính về khác.

Các thuộc tính về Configuration.

Tên thuộc tính	Ý nghĩa	Giá trị
hibernate.show_sql	In tất cả các câu truy vấn SQL đã dùng ra console (phục vụ debug)	True, False
hibernate.format_sql	In các câu SQL đã dùng ra file log và console (phục vụ debug)	True, False
hibernate.default_schema	Tên table mặc định, nếu trong câu SQL không chỉ rõ table đang dùng.	Tên CSDL. VD: table HocSinh
hibernate.default_catalog	Tên cơ sở dữ liệu mặc định, nếu trong câu SQL không chỉ rõ cơ sở dữ liệu đang dùng.	Tên CSDL. VD: quanlyhocsinh
hibernate.max_fetch_depth	Độ sâu join fetch (tự động join các bảng trong quá trình mapping)	Chỉ nên từ 0 - 3
hibernate.order_updates	Thay đổi mặc định hibernate sẽ update khóa chính của một đối tượng trước, sau đó mới tới các cột còn lại.	True, false
hibernate.generate_statistics		True, false

Các thuộc tính về Transaction.

Tên thuộc tính	Ý nghĩa	Giá trị
hibernate.transaction.flush_before_completion	Nếu là true thì session sẽ tự động flush trước khi hoàn thành.	True, False
hibernate.transaction.auto_close_session	Tự động đóng session sau khi hoàn thành.	True, False

Các thuộc tính khác.

Tên thuộc tính	Ý nghĩa	Giá trị
hibernate.current_session_context_class	Các tùy chỉnh cho session hiện tại	Jta, thread, managed, custom.Class
hibernate.query.factory_class	Chọn phương pháp phân tích cú pháp HQL, mặc định là true.	ASTQueryTranslatorFactory ClassicQueryTranslatorFactory

XIN CẢM ƠN!

