

Lab 3: Demo HibernateUtils, bảng dữ liệu Department.

Giao diện chính chương trình

Giao diện chính:

```
DemoCRUDHibernate (1) [Java Application]
Picked up _JAVA_OPTIONS: -Djava.net.preferIPv4Stack=true
-----MÔI BẠN CHỌN CHỨC NĂNG-----
+-----+
|                                     |
|                               Choose please                               |
|-----+-----+
| 1. Lấy danh sách tất cả các phòng ban trên hệ thống                    |
| 2. Tìm phòng ban theo ID                                                |
| 3. Tìm phòng ban theo Name                                              |
| 4. Tạo mới 1 phòng                                                      |
|-----+-----+
+-----+
```

Lấy danh sách tất cả phòng ban:

```
Console
DemoCRUDHibernate (1) [Java Application]
from
testingsystem1.department department0
+-----+
| ID | | Name | |
+-----+
| 15 | | 123 | |
| 16 | | abc | |
| 6 | | Accounting | |
| 3 | | Business Development | |
| 1 | | Engineering | |
| 10 | | Human Resources | |
| 4 | | Legal | |
| 12 | | Marketing | |
| 2 | | Product Management | |
| 11 | | Research and Development | |
| 8 | | Sale | |
| 7 | | Services | |
| 5 | | Support | |
| 13 | | test | |
| 9 | | Training | |
+-----+
-----MÔI BẠN CHỌN CHỨC NĂNG-----
```

Tạo khung chương trình:

Tạo mới 1 project: VTI_Lab3_DemoHibernateUtil.

Tạo mới các package và class như dưới:

Package: com.vti.entity, com.vti.frontend, com.vti.repository, com.vti.utils

Class: DepartmentRepository, Department, DemoCRUDHibernate, HibernateUtils

- ▼ VTI_Lab3_DemoHibernateUtil
 - ▼ src/main/java
 - ▼ com.vti
 - ▼ entity
 - > Department.java
 - ▼ frontend
 - > DemoCRUDHibernate.java
 - ▼ repository
 - > DepartmentRepository.java
 - ▼ utils
 - > **HibernateUtils.java**
 - > src/test/java
 - > JRE System Library [J2SE-1.5]
 - > Maven Dependencies
 - ▼ src
 - ▼ main
 - > java
 - ▼ resources
 - hibernate.cfg.xml
 - > test
 - > target
 - pom.xml

Thêm các cấu hình cho file: pom.xml

```
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.4.17.Final</version>
</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.20</version>
</dependency>
```

```

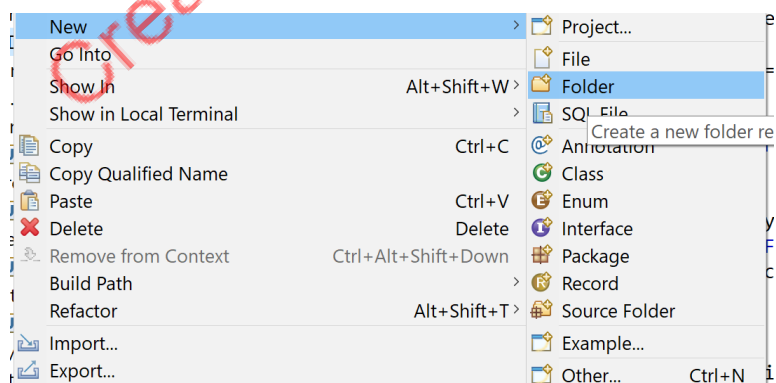
9      <packaging>jar</packaging>
10
11      <name>DemoCRUD</name>
12      <url>http://maven.apache.org</url>
13
14      <properties>
15          <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
16      </properties>
17
18      <dependencies>
19          <dependency>
20              <groupId>junit</groupId>
21              <artifactId>junit</artifactId>
22              <version>3.8.1</version>
23              <scope>test</scope>
24          </dependency>
25
26          <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
27          <dependency>
28              <groupId>org.hibernate</groupId>
29              <artifactId>hibernate-core</artifactId>
30              <version>5.4.17.Final</version>
31          </dependency>
32
33          <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
34          <dependency>
35              <groupId>mysql</groupId>
36              <artifactId>mysql-connector-java</artifactId>
37              <version>8.0.20</version>
38          </dependency>
39      </dependencies>
40  </project>
41
42

```



pom.xml

Tạo folder mới để lưu file hibernate.cfg.xml: src/main/resources



Thêm các cấu hình cho file: hibernate.cfg.xml trong resources

```

<?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>
        <!-- Database connection settings -->
        <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/testingsystem1</property>
        <property name="connection.username">root</property>

```

```

        <property name="connection.password">root</property>

        <!-- format code SQL -->
        <property name="show_sql">true</property>
        <property name="hibernate.format_sql">true</property>
        <property name="connection.pool_size">10</property>

        <!-- other -->
        <property name="hibernate.connection.characterEncoding">utf8</property>

    </session-factory>
</hibernate-configuration>

```



hibernate.cfg.xml

- ▼ DemoCRUDHibernate
 - ▼ src/main/java
 - ▼ com.vti
 - ▼ entity
 - > Department.java
 - ▼ frontend
 - > DemoCRUDHibernate.java
 - ▼ repository
 - > DepartmentRepository.java
 - > src/test/java
 - ▼ src/main/resources
 - ▣ hibernate.cfg.xml
 - > JRE System Library [J2SE-1.5]
 - > Maven Dependencies
 - > src
 - > target
 - ▣ pom.xml

Tạo Class HibernateUtils trong package com.vti.utils:

```

package com.vti.utils;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

import com.vti.entity.Department;

public class HibernateUtils {

    private static HibernateUtils instance;

    private Configuration configuration;
    private SessionFactory sessionFactory;

    public static HibernateUtils getInstance() {
        if (null == instance) {
            instance = new HibernateUtils();
        }
        return instance;
    }

    private HibernateUtils() {
        configure();
    }

    private void configure() {
        // load configuration
    }
}

```

```

        configuration = new Configuration();
        configuration.configure("hibernate.cfg.xml");

        // add entity
        configuration.addAnnotatedClass(Department.class);
    }

    private SessionFactory buildSessionFactory() {
        if (null == sessionFactory || sessionFactory.isClosed()) {
            ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder()
                .applySettings(configuration.getProperties()).build();

            sessionFactory = configuration.buildSessionFactory(serviceRegistry);
        }

        return sessionFactory;
    }

    public void closeFactory() {
        if (null != sessionFactory && sessionFactory.isOpen()) {
            sessionFactory.close();
        }
    }

    public Session openSession() {
        buildSessionFactory();
        return sessionFactory.openSession();
    }
}

```



HibernateUtils.java

Tạo Class Department trong package com.vti.entity:

```

package com.vti.entity;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "department", catalog = "testingsystem1")
public class Department implements Serializable {
    @Column(name = "DepartmentID")
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private short id;
    @Column(name = "DepartmentName", nullable = false, unique = true)
    private String name;

    /**
     * @return the id
     */
    public short getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(byte id) {
        this.id = id;
    }
}

```

```

    }

    /**
     * @return the name
     */
    public String getName() {
        return name;
    }

    /**
     * @param name the name to set
     */
    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Department [id=" + id + ", name=" + name + "]";
    }
}

```



Department.java

Tạo Class DepartmentRepository trong package com.vti.repository:

```

package com.vti.repository;

import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

import com.vti.entity.Department;
import com.vti.utils.HibernateUtils;

public class DepartmentRepository {
    private HibernateUtils hibernateUtils;

    public DepartmentRepository() {
        hibernateUtils = HibernateUtils.getInstance();
    }

    @SuppressWarnings("deprecation")
    public List<Department> getAllDepartments() {

        Session session = null;

        try {
            session = hibernateUtils.openSession();
            Query<Department> query = session.createQuery("FROM Department");

            return query.list();

        } finally {
            if (session != null) {
                session.close();
            }
        }
    }

    public Department getDepartmentByID(short id) {
        Session session = null;
        try {
            session = hibernateUtils.openSession();
            Department department = session.get(Department.class, id);

```

```

        return department;
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

public Department getDepartmentByName(String name) {
    Session session = null;
    try {
        // get session
        session = hibernateUtils.openSession();

        // create hql query
        Query<Department> query = session.createQuery("FROM Department WHERE name = :nameParameter");

        // set parameter
        query.setParameter("nameParameter", name);

        // get result
        Department department = query.uniqueResult();
        return department;
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

public void createDepartment(Department department) {
    Session session = null;
    try {
        // get session
        session = hibernateUtils.openSession();
        session.beginTransaction();

        // create
        session.save(department);

        session.getTransaction().commit();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

public boolean isDepartmentExistsByID(short id) {
    // get department
    Department department = getDepartmentByID(id);

    // return result
    if (department == null) {
        return false;
    }

    return true;
}

public boolean isDepartmentExistsByName(String name) {
    Department department = getDepartmentByName(name);

    if (department == null) {
        return false;
    }

    return true;
}

```

```
}
```



DepartmentRepository.java

Tạo Class DemoCRUDHibernate trong com.vti.frontend:

```
package com.vti.frontend;

import java.util.List;
import java.util.Scanner;

import com.vti.entity.Department;
import com.vti.repository.DepartmentRepository;

public class DemoCRUDHibernate {
    public static void main(String[] args) {
        while (true) {
            Scanner scanner = new Scanner(System.in);
            int choose = menu();
            switch (choose) {
                case 1:
                    showAllDepartmet();
                    break;
                case 2:
                    System.out.println("Nhập vào ID của phòng ban");
                    short id = scanner.nextShort();
                    showDepartmentByID(id);
                    break;
                case 3:
                    System.out.println("Nhập vào tên phòng ban cần tìm kiếm");
                    String nameDep = scanner.nextLine();
                    showDepartmentByName(nameDep);
                    break;
                case 4:
                    System.out.println("Tạo mới phòng ban");
                    System.out.println("Nhập tên phòng cần tạo:");
                    String nameDepCreate = scanner.nextLine();
                    Department dep = new Department();
                    dep.setName(nameDepCreate);
                    createDep(dep);
                    showAllDepartmet();
                    break;
                default:
                    System.out.println("Nhập lại:");
                    break;
            }
        }
    }

    private static void createDep(Department dep) {
        DepartmentRepository depRepository = new DepartmentRepository();
        depRepository.createDepartment(dep);
    }

    private static void showDepartmentByName(String nameDep) {
        DepartmentRepository depRepository = new DepartmentRepository();
        Department depByName = depRepository.getDepartmentByName(nameDep);
        if (depByName == null) {
            System.out.println("Không có phòng ban này trên hệ thống");
        } else {
            String leftAlignFormat = "| %-5s | %-25s |%n";
            System.out.format("+-----+-----+%n");
            System.out.format("| ID      | Name                                     |%n");
            System.out.format("+-----+-----+%n");
        }
    }
}
```



```

        System.out.format(leftAlignFormat, depByName.getId(), depByName.getName());
    }

}

private static void showDepartmentByID(Short id) {
    DepartmentRepository depRepository = new DepartmentRepository();
    Department dep = depRepository.getDepartmentByID(id);
    String leftAlignFormat = "| %-5s | %-25s |%n";
    System.out.format("+-----+-----+%n");
    System.out.format("| ID      | Name                               |%n");
    System.out.format("+-----+-----+%n");

    System.out.format(leftAlignFormat, dep.getId(), dep.getName());
}

private static void showAllDepartmet() {
    System.out.println("Danh sách phòng ban trên hệ thống");
    DepartmentRepository depRepository = new DepartmentRepository();
    List<Department> listDep = depRepository.getAllDepartments();
    String leftAlignFormat = "| %-5s | %-25s |%n";
    System.out.format("+-----+-----+%n");
    System.out.format("| ID      | Name                               |%n");
    System.out.format("+-----+-----+%n");

    for (Department department : listDep) {
        System.out.format(leftAlignFormat, department.getId(),
department.getName());
    }
}

@SuppressWarnings("resource")
private static int menu() {
    while (true) {
        System.out.println("-----MỜI BẠN CHỌN CHỨC NĂNG-----");
        String leftAlignFormat = "| %-72s |%n";
        System.out.format("+-----+");
        System.out.format("|                               Choose please");
        System.out.format("+-----+");
        System.out.format(leftAlignFormat, "1. Lấy danh sách tất cả các phòng ban");
        System.out.format(leftAlignFormat, "2. Tìm phòng ban theo ID");
        System.out.format(leftAlignFormat, "3. Tìm phòng ban theo Name");
        System.out.format(leftAlignFormat, "4. Tạo mới 1 phòng");
        System.out.format("+-----+");
        Scanner sc = new Scanner(System.in);
        if (sc.hasNextInt()) {
            int i = sc.nextInt();
            if ((i == 1) || (i == 2) || (i == 3) || (i == 4)) {
                return i;
            } else {
                System.out.println("Nhập lại");
            }
        } else {
            System.out.println("Nhập lại");
        }
    }
}
}

```



DemoCRUDHibernate.java