

Bài thực hành số 7

Mục tiêu

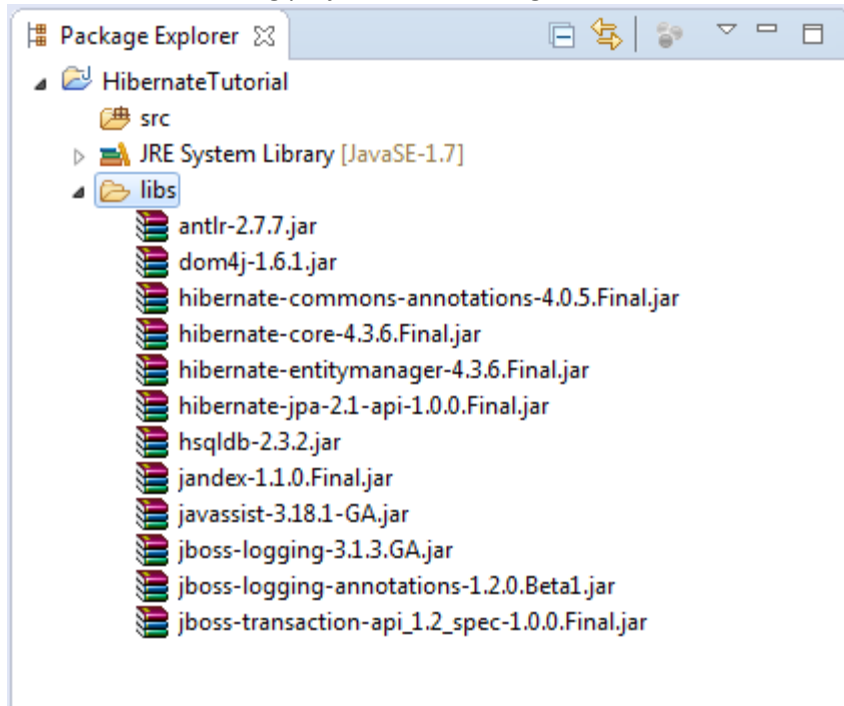
Hiểu cách sử dụng các thành phần cơ bản trong ngôn ngữ Hibernate

- **Hibernate Mapping: one to one**
- **Hibernate Mapping: Many to one**
- **Hibernate Mapping: one to many**
- **Hibernate Mapping: many to many**

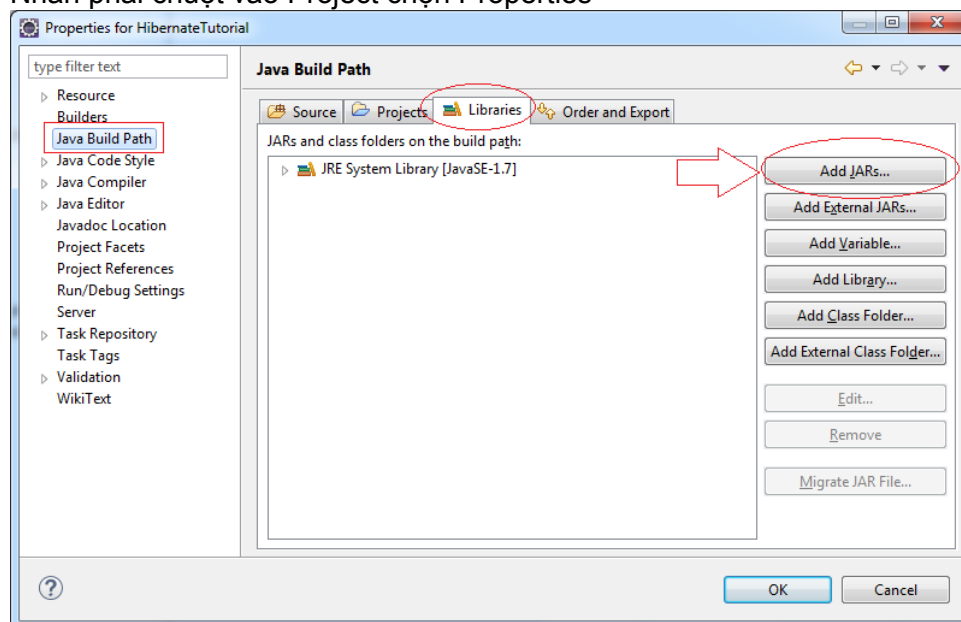
Sử dụng cơ sở dữ liệu tài nguyên simpleHr

Bài 1 Tạo Project và khai báo thư viện

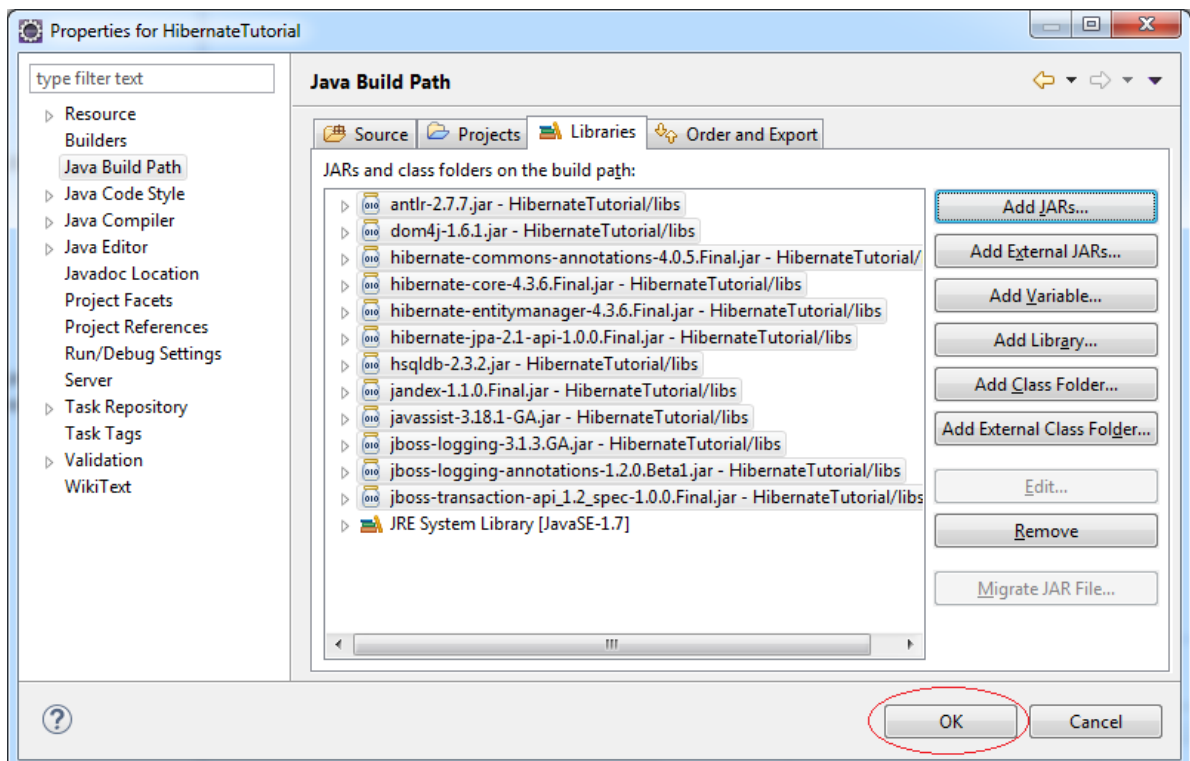
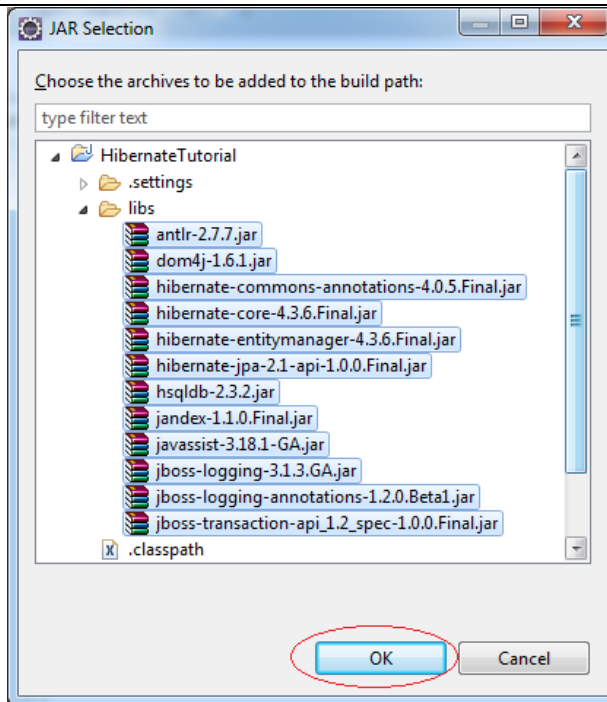
Tạo thư mục Libs trong project và add những file sau



Nhấn phải chuột vào Project chọn Properties

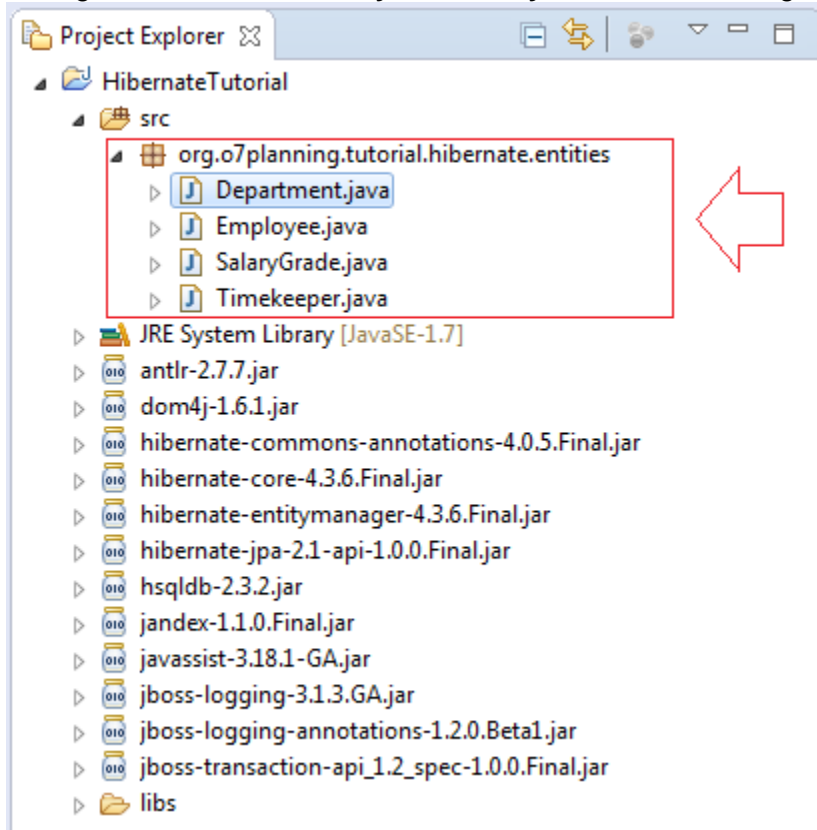


Add hết tất cả các thư viện có trong thư mục libs:



Bài 2 Tạo Class Entity

Chúng ta tạo các class **Entity**. Mỗi **Entity** sẽ mô tả một bảng trong **DB**.



1. **Department** - Phòng ban
2. **Employee** - Nhân viên
3. **SalaryGrade** - Bậc lương
4. **Timekeeper** - Máy chấm công, giờ ra vào của nhân viên.

- **Department.java**

?

```

1 package org.Fpoly.tutorial.hibernate.entities;
2
3 import java.util.HashSet;
4 import java.util.Set;
5
6 import javax.persistence.Column;
```

```

7 import javax.persistence.Entity;
8 import javax.persistence.FetchType;
9 import javax.persistence.Id;
10import javax.persistence.OneToOne;
11import javax.persistence.Table;
12import javax.persistence.UniqueConstraint;
13
14@Entity
15@Table(name = "DEPARTMENT",
16    uniqueConstraints = { @UniqueConstraint(columnNames = { "DEPT_NO"
17    }) })
18public class Department {
19
20    private Integer deptId;
21    private String deptNo;
22
23    private String deptName;
24    private String location;
25    private Set<Employee> employees = new HashSet<Employee>(0);
26
27    public Department() {
28    }
29
30    public Department(Integer deptId, String deptName, String location) {
31        this.deptId = deptId;
32        this.deptNo = "D" + this.deptId;
33        this.deptName = deptName;
34        this.location = location;
35    }
36
37    @Id
38    @Column(name = "DEPT_ID")
39    public Integer getDeptId() {
40        return deptId;
41    }
42
43    public void setDeptId(Integer deptId) {

```

```

44     this.deptId = deptId;
45 }
46
47 @Column(name = "DEPT_NO", length = 20, nullable = false)
48 public String getDeptNo() {
49     return deptNo;
50 }
51
52 public void setDeptNo(String deptNo) {
53     this.deptNo = deptNo;
54 }
55
56 @Column(name = "DEPT_NAME", nullable = false)
57 public String getDeptName() {
58     return deptName;
59 }
60
61 public void setDeptName(String deptName) {
62     this.deptName = deptName;
63 }
64
65 @Column(name = "LOCATION")
66 public String getLocation() {
67     return location;
68 }
69
70 public void setLocation(String location) {
71     this.location = location;
72 }
73
74 @OneToMany(fetch = FetchType.LAZY, mappedBy = "department")
75 public Set<Employee> getEmployees() {
76     return employees;
77 }
78
79 public void setEmployees(Set<Employee> employees) {
80     this.employees = employees;

```

```
81 }
    }
```

- **Employee.java**

?

```
1 package org.Fpoly.tutorial.hibernate.entities;
2
3 import java.util.Date;
4 import java.util.HashSet;
5 import java.util.Set;
6
7 import javax.persistence.Column;
8 import javax.persistence.Entity;
9 import javax.persistence.FetchType;
10 import javax.persistence.Id;
11 import javax.persistence.JoinColumn;
12 import javax.persistence.Lob;
13 import javax.persistence.ManyToOne;
14 import javax.persistence.OneToOne;
15 import javax.persistence.Table;
16 import javax.persistence.Temporal;
17 import javax.persistence.TemporalType;
18 import javax.persistence.UniqueConstraint;
19
20 @Entity
21 @Table(name = "EMPLOYEE",
22       uniqueConstraints = { @UniqueConstraint(columnNames = {
23 "EMP_NO" }) })
24 public class Employee {
25     private Long empId;
26     private String empNo;
27
28     private String empName;
29     private String job;
30     private Employee manager;
```

```

31  private Date hideDate;
32  private Float salary;
33  private byte[] image;
34
35  private Department department;
36  private Set<Employee> employees = new HashSet<Employee>(0);
37
38  public Employee() {
39  }
40
41  public Employee(Long empId, String empName, String job, Employee
42  manager,
43          Date hideDate, Float salary, Float comm, Department
44  department) {
45      this.empId = empId;
46      this.empNo = "E" + this.empId;
47      this.empName = empName;
48      this.job = job;
49      this.manager = manager;
50      this.hideDate = hideDate;
51      this.salary = salary;
52      this.department = department;
53  }
54
55  @Id
56  @Column(name = "EMP_ID")
57  public Long getEmpId() {
58      return empId;
59  }
60
61  public void setEmpId(Long empId) {
62      this.empId = empId;
63  }
64
65  @Column(name = "EMP_NO", length = 20, nullable = false)
66  public String getEmpNo() {
67      return empNo;

```



```

68  }
69
70  public void setEmpNo(String empNo) {
71      this.empNo = empNo;
72  }
73
74  @Column(name = "EMP_NAME", length = 50, nullable = false)
75  public String getEmpName() {
76      return empName;
77  }
78
79  public void setEmpName(String empName) {
80      this.empName = empName;
81  }
82
83  @Column(name = "JOB", length = 30, nullable = false)
84  public String getJob() {
85      return job;
86  }
87
88  public void setJob(String job) {
89      this.job = job;
90  }
91
92  @ManyToOne(fetch = FetchType.LAZY)
93  @JoinColumn(name = "MNG_ID")
94  public Employee getManager() {
95      return manager;
96  }
97
98  public void setManager(Employee manager) {
99      this.manager = manager;
100 }
101
102 @Column(name = "HIRE_DATE", nullable = false)
103 @Temporal(TemporalType.DATE)
104 public Date getHideDate() {

```

```

105     return hideDate;
106 }
107
108 public void setHideDate(Date hideDate) {
109     this.hideDate = hideDate;
110 }
111
112 @Column(name = "SALARY", nullable = false)
113 public Float getSalary() {
114     return salary;
115 }
116
117 public void setSalary(Float salary) {
118     this.salary = salary;
119 }
120
121 @Column(name = "IMAGE", length = 1111111, nullable = true)
122 @Lob
123 public byte[] getImage() {
124     return image;
125 }
126
127 public void setImage(byte[] image) {
128     this.image = image;
129 }
130
131 @ManyToOne(fetch = FetchType.LAZY)
132 @JoinColumn(name = "DEPT_ID", nullable = false)
133 public Department getDepartment() {
134     return department;
135 }
136
137 public void setDepartment(Department department) {
138     this.department = department;
139 }
140
141 @OneToMany(fetch = FetchType.LAZY, mappedBy = "empId")

```

```

142 public Set<Employee> getEmployees() {
143     return employees;
144 }
145
146 public void setEmployees(Set<Employee> employees) {
147     this.employees = employees;
148 }
149
150 }
```

- **SalaryGrade.java**

```

?
1 package org.Fpoly.tutorial.hibernate.entities;
2
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.Id;
6 import javax.persistence.Table;
7
8 @Entity
9 @Table(name = "SALARY_GRADE")
10 public class SalaryGrade {
11     private Integer grade;
12     private Float lowSalary;
13     private Float highSalary;
14
15     public SalaryGrade() {
16     }
17
18     public SalaryGrade(Integer grade, Float lowSalary, Float
19 highSalary) {
20         this.grade = grade;
21         this.lowSalary = lowSalary;
22         this.highSalary = highSalary;
23     }

```

```

24
25  @Id
26  @Column(name = "GRADE")
27  public Integer getGrade() {
28      return grade;
29  }
30
31  public void setGrade(Integer grade) {
32      this.grade = grade;
33  }
34
35  @Column(name = "LOW_SALARY", nullable = false)
36  public Float getLowSalary() {
37      return lowSalary;
38  }
39
40  public void setLowSalary(Float lowSalary) {
41      this.lowSalary = lowSalary;
42  }
43
44  @Column(name = "HIGH_SALARY", nullable = false)
45  public Float getHighSalary() {
46      return highSalary;
47  }
48
49  public void setHighSalary(Float highSalary) {
50      this.highSalary = highSalary;
51  }
    }

```

- **Timekeeper.java**

?

```

1 package org.Fpoly.tutorial.hibernate.entities;
2
3 import java.util.Date;

```

```

4
5 import javax.persistence.Column;
6 import javax.persistence.Entity;
7 import javax.persistence.FetchType;
8 import javax.persistence.GeneratedValue;
9 import javax.persistence.Id;
10import javax.persistence.JoinColumn;
11import javax.persistence.ManyToOne;
12import javax.persistence.Table;
13import javax.persistence.Temporal;
14import javax.persistence.TemporalType;
15
16import org.hibernate.annotations.GenericGenerator;
17
18@Entity
19@Table(name = "TIMEKEEPER")
20public class Timekeeper {
21    public static final char IN = 'I';
22    public static final char OUT = 'O';
23
24    private String timekeeperId;
25
26    private Date dateTime;
27
28    private Employee employee;
29
30    // 'I' or 'O'
31    private char inOut;
32
33    @Id
34    @GeneratedValue(generator = "uuid")
35    @GenericGenerator(name = "uuid", strategy = "uuid2")
36    @Column(name = "Timekeeper_Id", length = 36)
37    public String getTimekeeperId() {
38        return timekeeperId;
39    }
40

```

```

41     public void setTimekeeperId(String timekeeperId) {
42         this.timekeeperId = timekeeperId;
43     }
44
45     @Column(name = "Date_Time", nullable = false)
46     @Temporal(TemporalType.TIMESTAMP)
47     public Date getDateTime() {
48         return dateTime;
49     }
50
51     public void setDateTime(Date dateTime) {
52         this.dateTime = dateTime;
53     }
54
55     @ManyToOne(fetch = FetchType.LAZY)
56     @JoinColumn(name = "EMP_ID", nullable = false)
57     public Employee getEmployee() {
58         return employee;
59     }
60
61     public void setEmployee(Employee employee) {
62         this.employee = employee;
63     }
64
65     @Column(name = "In_Out", nullable = false, length = 1)
66     public char getInOut() {
67         return inOut;
68     }
69
70     public void setInOut(char inOut) {
71         this.inOut = inOut;
72     }
73
74 }

```

Bài 3

Giảng viên giao thêm bài cho sinh viên.

Yêu cầu nộp bài

Cuối giờ thực hành, sinh viên tạo thư mục theo tên <Tên đăng nhập SV>_Lab1, chứa tất cả sản phẩm của những bài lab trên, nén lại thành file zip và upload lên mục nộp bài tương ứng trên LMS.

Đánh giá bài lab

STT	Bài số	Điểm
1	Bài 1	
2	Bài 2	
3	Bài 3	