

第八章 分层开发

课前回顾

1. JDBC 操作步骤有哪些

```
//引入驱动包
//加载驱动
Class.forName("com.mysql.cj.jdbc.Driver");
//获取连接：从驱动管理器中获取连接
Connection conn = DriverManager.getConnection(url, username, password);
//创建SQL语句执行器
PreparedStatement ps = conn.prepareStatement(sql);
//设置参数
ps.setString(1, "");
//执行查询或者更新
ResultSet rs = ps.executeQuery();
//遍历结果集
while(rs.next()){
    rs.getString("name");
    rs.getInt(2);
}

// int affectedRows = ps.executeUpdate();
//关闭资源
rs.close();
ps.close();
conn.close();
```

2. Class类的常用方法有哪些

```
Constructor c = clazz.getDeclaredConstructor();//获取类中定义的非参构造
c.setAccessible(true); //设置访问权限
Object o = c.newInstance();
Field[] fields = clazz.getDeclaredFields();
Method m = clazz.getDeclaredMethod("", 参数类型列表);
m.invoke(o, 值);
```

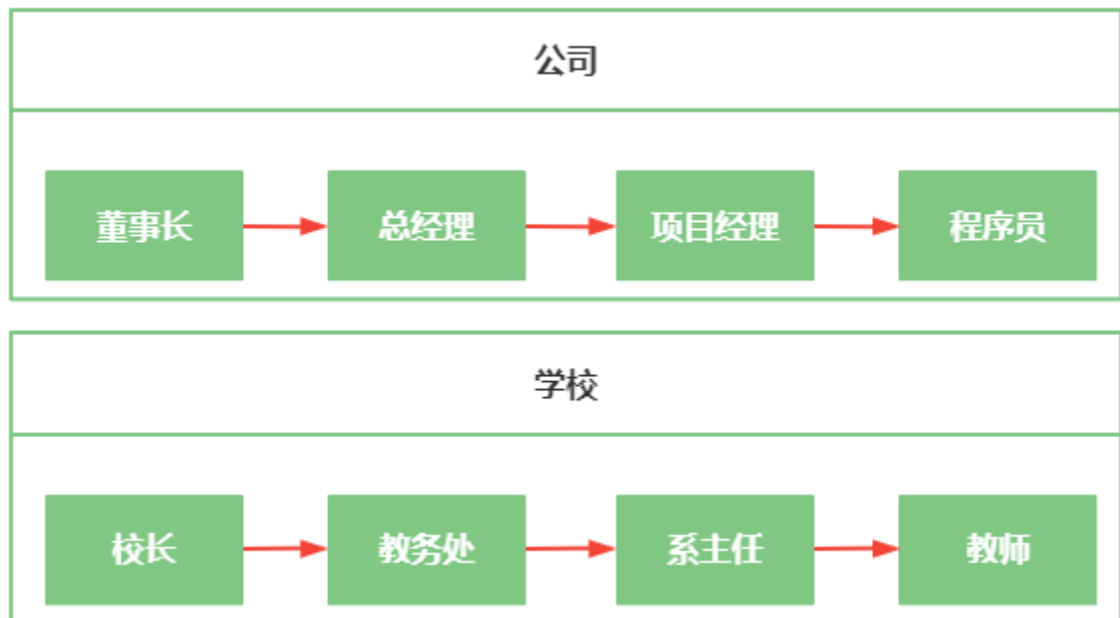
章节内容

- 分层开发 **重点**

章节目标

- 掌握分层开发

第一节 生活中的分层



由上图可以看出，公司和学校对职工都进行了分层，为什么要进行这样的分层呢？

- 分层后每一层只专注于自己所做的事情，能够提高作业质量
- 便于分工协作，提高作业效率
- 便于业务拓展
- 方便问题排查

这样的分层有什么特点呢？

- 上层制定任务，下层接受任务，上层安排下层做事，但下层不能安排上层做事
- 下层只需要汇报做事的结果，不需要汇报做事的过程

第二节 分层开发

1. 三层结构

生活中的分层也可以应用于软件开发中，软件开发分层主要分为三层：

- 界面层，又称控制层（controller）
与用户进行交互，主要负责数据采集和展示
- 业务逻辑层（service）
负责处理功能模块的业务逻辑，以及界面层和数据访问层的数据流转
- 数据访问层（data access object => dao）
只负责与数据库进行交互

2. 分层原则

软件分层开发也具有生活中分层的特点，这些特点被称之为分层原则：

- 封装性原则
每层只向外公开接口，但隐藏了内部实现细节
- 顺序访问原则
下层为上层服务，但下层不能使用上层服务
- 开闭原则
对扩展开放，对修改关闭

3. 分层的优点

软件分层开发的好处：

- 各层专注于自己所做的事情，便于提高开发质量
- 便于分工协作，提高开发效率
- 便于程序扩展
- 便于代码复用
- 易于维护

4. 分层开发案例

使用分层开发完成用户注册与登录功能。