Neuedu

东软睿道内部公开

文件编号： D000-

Spring框架技术

版本：1.1.0-0.0.0

2017-5-8

Neuedu

文件修改控制

修改编号	版本	修改条款及内容	修改日期
1	1.0.0-0.0.0	创建	2012/9/1
2	1.1.0-0.0.0	修改课件模板为新模板	2017/5/8
3	1.1.0-0.0.0	修改了课件内容和案例源码，以Spring3.2版本为例，进行讲解	2017/5/8
4	1.1.0-0.0.0	修改课件内容，增加注解编程	2017/5/8

www.neuedu.com

Neuedu

Spring框架技术

—— Spring 入门

V1.1

课程目标

Neuedu

- 掌握Spring IoC、DI的概念；
- 理解Spring AOP的概念和技术原理；
- 能够使用“注解”方式和“静态配置文件”方式实现IOC和AOP。

课程目录

Neuedu

内容	参考课时（H）
第一章 Spring入门	1
第二章 容器和Bean的基本原理	2
第三章 依赖注入的实现（IOC）	4
第四章 面向切面编程（AOP）	5
合计	12

本章内容

Neuedu

节	知识点	掌握程度	难易程度
Spring简介	Spring是什么	了解	
	Spring框架组成	了解	
	Spring的历史及目标	了解	
	Spring的优点	了解	
第一个Spring程序	环境搭建	掌握	
	编写接口和实现类	掌握	
	编写配置文件	掌握	
	编写测试类	掌握	

Spring是什么

Neuedu

- 轻量级框架
  - 发布仅单一jar包；额外的消耗可忽略不计
  - 非侵入性
- 容器
  - 管理对象的生命周期，控制对象的创建方式

- Ioc/DI(Inversion of Control/ Dependency Injection)
  - 控制反转/依赖注入
  - 配置及管理对象的依赖关系
- AOP(Aspect-oriented programming)
  - 面向方面编程
  - OOP及IoC的补充

- 通用解决方案及最佳实践
  - 当前流行开源产品的封装

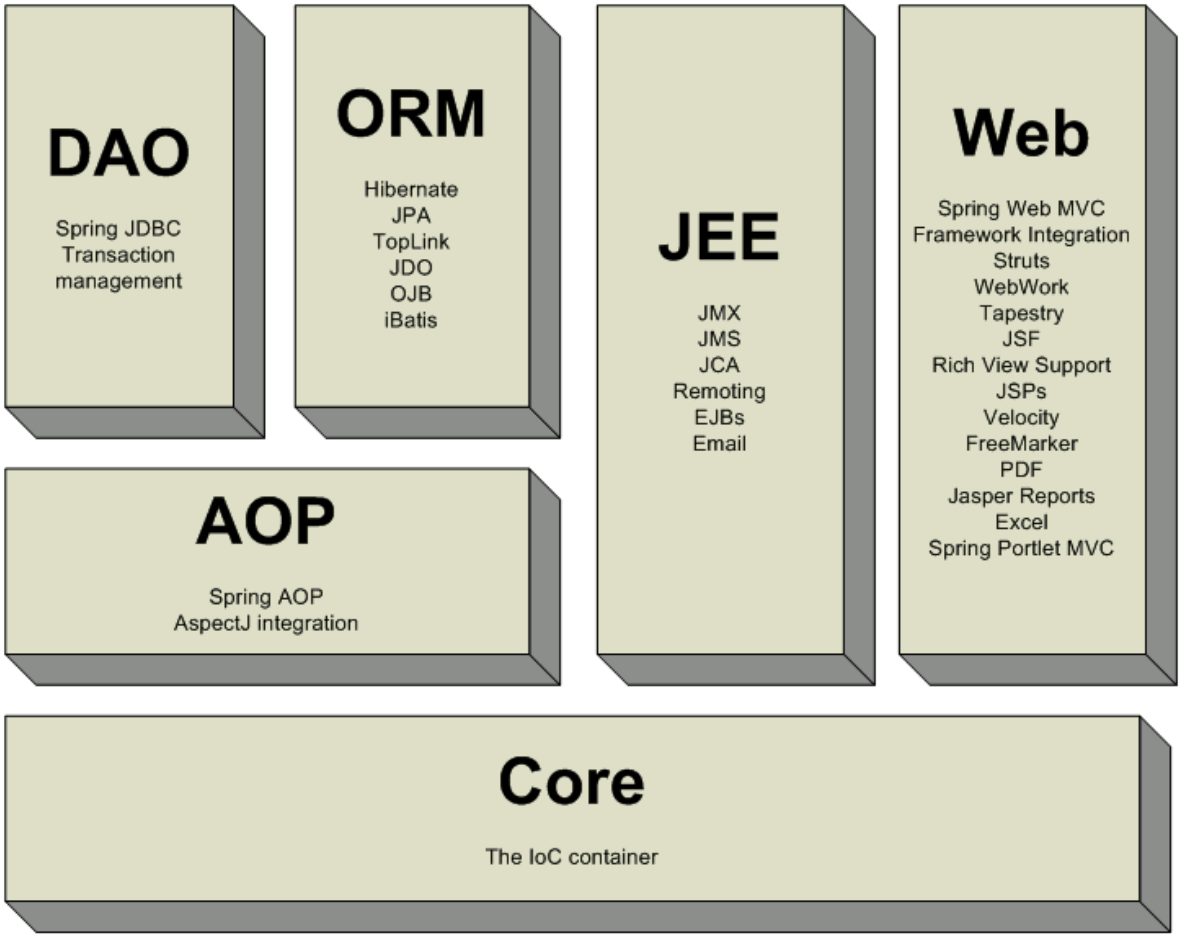
Spring是什么

Neuedu

- Spring是一个开源的JavaEE框架。
  - 它作为一个优秀的轻量级的企业应用开发框架，可以大大简化企业应用开发的复杂性；
  - 能够创建出松耦合、易测试、易扩展、易维护的Java应用系统；

Spring框架组成

Neuedu



Spring框架组成（核心容器）

- 核心容器：
  - 核心容器提供Spring框架的基本功能，为Spring提供了基础服务支持，核心容器的主要组件就是BeanFactory
  - BeanFactory是所有基于Spring框架系统的核心，通过BeanFactory，Spring使用工厂模式来实现IoC，将应用程序的配置和依赖关系与实际的应用程序分离开来。
  - Spring之所以称为容器，就是由于BeanFactory的自动装备和注入。

Spring框架组成（Application Context）

- Application Context（上下文）
  - 由一个配置文件，向 Spring 框架提供上下文信息。
  - BeanFactory使spring成为容器，上下文模块使Spring成为框架。
  - 这个模块对BeanFactory进行了扩展，添加了对I18N，系统生命周期事件以及验证的支持。 这个模块提供了许多企业级服务，例如：邮件服务，JNDI访问，EJB集成，远程调用以及定时服务，并且支持与模板框架的集成。

Spring框架组成（AOP）

- Spring AOP：
  - AOP面向切面编程。该模块将AOP 编程功能集成到了Spring 框架中。这样，凡是 Spring 框架管理的任何对象都可以很容易地支持 AOP。该模块为应用程序中基于Spring 管理的对象提供了事务管理服务。
  - 这个模块由于使用了 AOP Alliance的API，所以，可以和其他AOP框架互通，
  - AOP Alliance是一个开源项目，目的是促进AOP的使用，并且通过定义一套通用的接口和组件来确保不同的AOP之间达到互通性。

Spring框架组成（DAO）

Neuedu

- Spring DAO：
  - 这个模块封装了数据库连接的创建、语句对象生成、结果集处理、连接关闭等操作，而且重构了所有数据库系统的异常信息，用户不再需要处理数据库异常了。

  - 在这个模块中，利用了Spring的AOP模块完成了为系统中对象提供事务管理的服务。

Spring框架组成（ORM）

- Spring ORM:
  - Spring 没有实现自己的ORM方案，而是为当前主流的ORM框架预留了整合接口，**包括hibernate、JDO等。**所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。
  - Spring的事物管理支持所有这些ORM框架以及JDBC。

Spring框架组成（Web）

- Spring Web 模块：
  - Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。
  - Spring 框架支持与 Jakarta Struts 的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。
  - MVC框架是一个全功能的构建 Web 应用程序的 MVC 实现。
  - 通过策略接口，MVC 框架变成为高度可配置的。MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI等。

Spring的历史及目标

Neuedu

- Spring的核心代码均来自于真实的项目，Rod Johnson是这个产品的创造者，是从商业项目开发实践中逐步提炼出的一种架构基调。

- 从2003年正式启动，整个项目的开发始终贯彻着如下的核心架构理念，具有概念上的完整性和一致性：
  - 降低开发成本，方便使用，促进良好的编程习惯
  - 整合各类框架，遵守不重新发明轮子的原则
  - 易于选择，方便测试
  - 统一配置，灵活可扩展
  - 非侵入性
  - 提供最好的IOC解决方案
  - 提供最好的AOP解决方案

Spring的优点

Neuedu

- 开源框架，开放性较高
- 有效地组织中间层对象
- 多种可选的事务处理方式
- 多种可选的持久层策略
- 多种可选的Web MVC框架策略

- 高度可扩展的安全解决方案
- 有效的消除单例、工厂等模式的使用
- 将面向接口编程做到实处
- 使单元测试变得简单
- 使EJB的使用成为一个选择
- 提供了一致的数据访问框架
- 只选择你需要的

Hello Spring

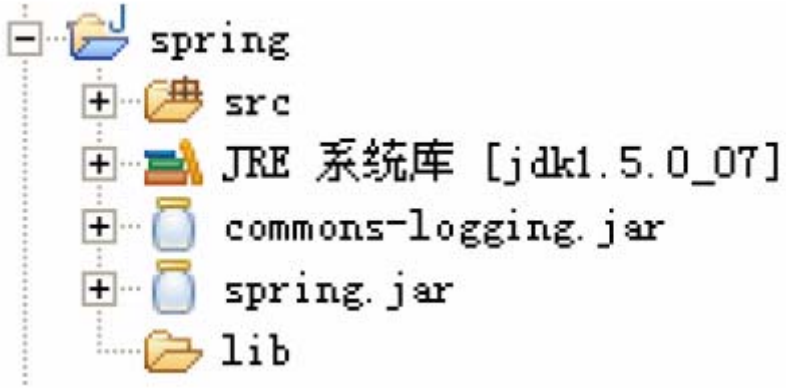
Neuedu

- 环境搭建
  - 进入spring官网https://spring.io/下载spring开发包
  - 建立java工程或者java web工程
  - 将spring包导入工程
- 创建接口
- 创建接口实现类
- 编写配置文件
- 编写测试类

Hello Spring

Neuedu

- 环境搭建



Hello Spring

Neuedu

- 创建接口和实现类

```
public interface Formater {
    public String execute(String s);
}
```

```
public class UpperFormater implements Formater {
    private String title;

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String execute(String s) {
        return (title+" : "+s).toUpperCase();
    }
}
```

```
public class LowerFormater implements Formater {
    private String title;

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String execute(String s) {
        return (title+" : "+s).toLowerCase();
    }
}
```

Hello Spring

Neuedu

- 编写配置文件

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

        xsi:schemaLocation="http://www.springframework.org/schema/bea
   ns http://www.springframework.org/schema/beans/spring-beans-
   2.5.xsd">

   <bean id="formaterBean" class="example.LowerFormater">
        <property name="title">
                <value>Hello Spring</value>
        </property>
   </bean>
</beans>
```

Hello Spring

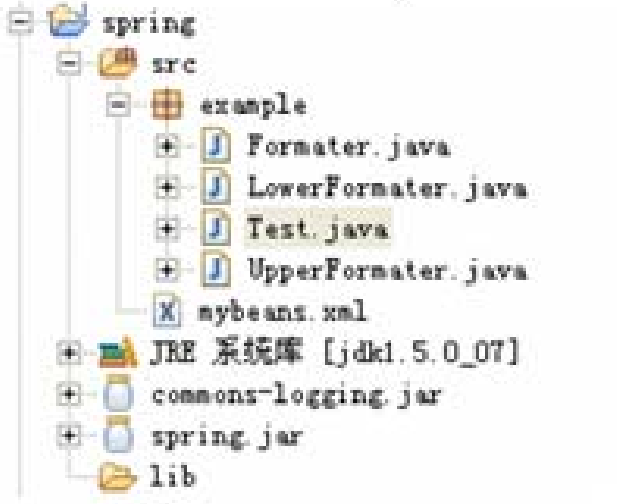
Neuedu

- 编写测试类

```
public class Test {

    public static void main(String[] args) {
        BeanFactory factory = new ClassPathXmlApplicationContext("mybeans.xml");
        Formater f = (Formater) factory.getBean("formaterBean");
        System.out.println(f.execute("neusoft"));
    }

}
```



Hello Spring

Neuedu

- 除测试代码外，所有程序中没有引用Spring任何组件
- UpperFormater和LowerFormater的title属性均由Spring读取配置文件动态设置
- 客户代码仅仅面向接口编程，无需知道具体实现类，同时可以很简单地通过修改配置文件来切换具体的底层实现类

思考：如果没有Spring，程序需要怎样实现?

本章重点总结

Neuedu

- 了解Spring框架组成
- 掌握Spring环境搭建和第一个程序编程流程

Neuedu