Spring源码专题

前提：Spring基础。

1，Spring源码开篇

为什么要学习Spring？源码

优秀框架，优雅。时间检验。

面试、设计模式，分层设计，架构体系。

构建起我们自己的知识体系，思考的质量，解决问题的速度。

1+1=2 --> 1+1+1=3 --> 1*3=3 --->33d9乘法口诀

3*3 = 9

Spring-->Spring生态：Spring、SpringBoot、SpringCloud。

?

2，快速回顾Spring用法

目的：熟悉Spring的用法、搭建起一个实验环境

2.1 依赖

```
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.2.0.RELEASE</version>
    </dependency>
</dependencies>
```

2.2 相关类

```
/**
 * 主配置类
 */
@Configuration
@ComponentScan("com.myflx")//todo 自行改路径
public class AppConfig {
}

/**
 * 业务类
 */
```

```
@Repository
public class OrderDao {
    public void hello() {
        System.out.println("OrderDao hello...");
    }
}

@Service
//@Component
public class OrderService {
    @Autowired
    private OrderDao orderDao;

    public void hello() {
        System.out.println("OrderService hello...");
        orderDao.hello();
    }
}

/**
 * 入口类
 */
public class Bootstrap {
    public static void main(String[] args) {
        AnnotationConfigApplicationContext annotationConfigApplicationContext =
new AnnotationConfigApplicationContext(AppConfig.class);
        OrderService orderService = (OrderService)
annotationConfigApplicationContext.getBean("orderService");
        orderService.hello();
    }
}
```

启动Bootstrap#main方法就可以 看到预期效果。简单的几个类就跑通了Spring。

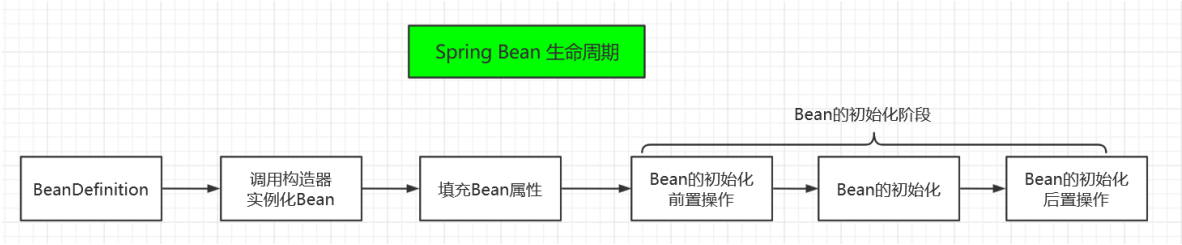
3，手写精简版Spring

1，前提要求

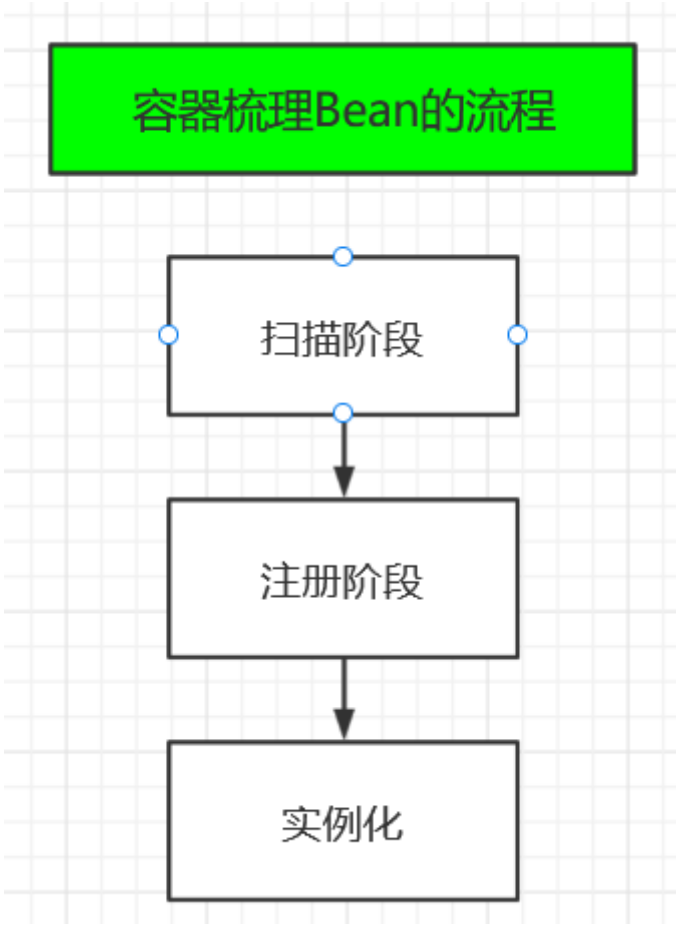
spring的两大核心：AOP、IOC。基于spring的bean，spring bean是核心中的核心。

Spring Bean生命周期

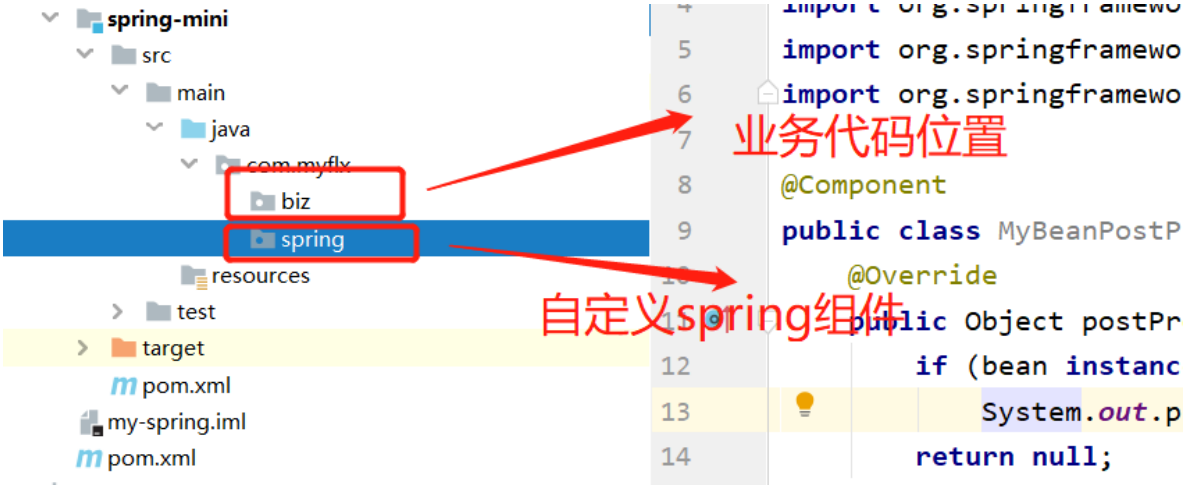
```
//1.实例化阶段
OrderService construct....
//2.属性填充阶段。例子：OrderDao的注入就发生在本阶段

//3.初始化
OrderService postProcessBeforeInitialization
OrderService initializing...
OrderService postProcessAfterInitialization
```



容器处理Bean的流程



2，项目结构



3，自定义Spring组件

自定义注解：`@Component`，`@ComponentScan`，`@Scope`,`@Autowired`

上下文：`AnntationConfigAplicationContext#getBean`

Bean相关组件：`BeanDefinition`，`InitializingBean`，`BeanPostProcessor`

4，代码实现

实现入口

```
public AnnotationConfigApplicationContext(Class<?> configClass) {
    this.configClass = configClass;
    //1.扫描。扫描包路径、解析所有文件信息。做判断
    //2.注册。如果说是符合条件的Bean（@Component 注解的类），将相关的Bean的信息转换成
BeanDefinition
    doScan();
    //3.实例化。单例对象（非懒加载的对象）
    initializeNotLazyBean();
}
```

```
public AnnotationConfigApplicationContext(Class<?> configClass) {
    this.configClass = configClass;
    //1.扫描。扫描包路径、解析所有文件信息。做判断
    //2.注册。如果说是符合条件的Bean（@Component 注解的类），将相关的Bean的信息转换成BeanDefinition
    doScan();
    //3.实例化。单例对象（非懒加载的对象）
    initializeNotLazyBean();
}
/** 扫描+注册 ...*/
private void doScan() {...}

/** 实例化-单例对象 ...*/
private void initializeNotLazyBean() {...}

/** 带缓存的Bean创建 ...*/
private Object createBean(BeanDefinition definition) {...}

/** 直接创建Bean ...*/
private Object doCreateBean(BeanDefinition definition) {...}
private Object initializing(Object instance, BeanDefinition definition) throws Exception {...}
private Object postProcessAfterInitialization(Object instance, BeanDefinition definition) {...}
private Object postProcessBeforeInitialization(Object instance, BeanDefinition definition) {...}
private void populateBean(Object instance, BeanDefinition definition) {...}

public Object getBean(String beanName) {...}
```

5，总结

感谢支持！

精简版的Spring

为什么要写精简版的Spring?

见视频讲解。

为什么要学习Spring?

开篇：提高自己能力

心理准备、认可------> 把事情想明白、搞清楚。

Spring怎么学?

方法论：效率、效果。

- 构建一个实验环境。**环境、准备工作。**
- 了解Spring的用法，用过Spring。**入门。**
- 确定一个核心，要研究的主题。SpringBean。SpringWeb、AOP、接口定义、设计模式、资源及其加载、上下文、事件。**抓住重点。**
- 关键因子：核心概念、生命周期、主脉络、主流程。**深入了解。**
- 模仿、学以致用的阶段。学习他的方法方式，学习他的设计。走一遍Spring走的过路，对他了解更深。**模仿，学以致用，举一反三。**

思考--->独立思考。

4，Spring源码解读说明

99，阅读源码技能

关于实验环境

要求实验环境尽可能的纯粹，不受其他环境的影响。jar不要有多余的。