第十五章实验报告

**实验名称: Spring容器中Bean的生命周期**

**学 院:计算机科学与工程学院/软件学院**

**专 业: 计算机科学与技术**

**年 级: 2021级**

**学 号: 202110803009**

**学生姓名: 欧阳俊杰**

**任课教师: 黄汉民**

实验目的：

探究Spring容器中Bean的生命周期

实验环境：

Java开发环境，Spring框架

实验步骤：

1. 创建一个简单的Java类，实现Bean的基本功能。

2. 编写测试类，观察Bean在不同阶段的行为。

3. 分析Spring容器中Bean的生命周期过程。

实验结果：

Spring容器中Bean的生命周期可以分为以下几个阶段：

1. 实例创建：通过构造器或工厂方法创建Bean实例。

2. 属性设置：为Bean的属性设置值，包括基本数据类型、对象引用等。

3. 初始化：调用Bean的初始化方法，此时Bean已经可以使用。

4. 运行：Bean在应用中发挥作用，执行业务逻辑。

5. 销毁：当容器关闭时，调用Bean的销毁方法，释放资源。

实验代码：

1. 创建一个简单的Java类，实现Bean的基本功能：

```java

package com.example.bean;

import org.springframework.stereotype.Component;

@Component

public class Person {

private String name;

private String address;

private int phone;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public int getPhone() {

return phone;

}

public void setPhone(int phone) {

this.phone = phone;

}

@Override

public String toString() {

return "Person{" +

"name='" + name + '\'' +

", address='" + address + '\'' +

", phone=" + phone +

'}';

}

}

```

2. 编写测试类，观察Bean在不同阶段的行为：

```java

package com.example.test;

import com.example.bean.Person;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.stereotype.Component;

public class BeanLifeCycleTest {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(BeanLifeCycleConfig.class);

// 获取Bean

Person person = context.getBean(Person.class);

// 输出Bean实例

System.out.println("Bean实例：" + person);

// 设置属性

person.setName("张三");

person.setAddress("北京");

person.setPhone(13800138000);

// 输出设置后的Bean实例

System.out.println("设置属性后的Bean实例：" + person);

// 调用Bean的初始化方法

person.init();

// 输出初始化后的Bean实例

System.out.println("初始化后的Bean实例：" + person);

// 调用Bean的销毁方法

context.destroy();

}

}

```

3. 创建一个配置类，用于启动Spring容器：

```java

package com.example.config;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan("com.example.bean")

public class BeanLifeCycleConfig {

}

```

实验结论：

通过实验可以了解到Spring容器中Bean的生命周期，从实例创建、属性设置、初始化、运行到销毁。在实际开发过程中，可以针对Bean的生命周期进行编程，以实现自定义的逻辑操作。