第六章习题答案

1.填空题

（1）IOC（控制反转）、AOP（面向切面编程）

（2）Beans、Core

（3）Context

（4）Spring5、Java 8

（5）BeanFactory

2.选择题

（1）B

（2）D

（3）A

（4）C

（5）A

3.思考题

（1）

答：与其他开发框架相比，Spring具有无可比拟的优势，这主要表现在以下几个方面。

1. 降低耦合度，方便开发

通过IOC容器，Spring可以管理对象的生命周期、控制对象之间的依赖关系，如此一来，因硬编码造成的程序过度耦合得以避免。

2. 支持AOP编程

通过AOP，Spring可以对程序进行权限拦截、安全监控等操作，这可以减少通过传统OOP方法带来的代码冗余和繁杂。

3. 支持声明式事务

在Spring中，可以直接通过Spring配置文件管理数据库事务，省去了手动编程的繁琐，提升了开发效率。

4. 方便程序测试

Spring 中集成了Junit ，开发人员可以通过Junit进行单元测试。

5. 方便集成各种优秀框架

Spring 提供了一个广阔的基础平台，它不排斥各种优秀的开源框架，其内部提供了对各种优秀框架（如MyBatis、Hibernate、Quartz 等）的直接支持。

6. 降低 JavaEE API 的使用难度

Spring封装了JavaEE 中使用难度较大的API，经过封装，这些 API 更容易被开发人员理解和调用。

（2）

答：容器是Spring框架实现功能的基础，Spring容器类似一家超级工厂，当Spring启动时，所有被配置过的类都会被纳入到Spring容器的管理之中。

4.编程题

Ⅰ.创建person对象，定义属性方法

1. package pojo;
2. public class Person {
3. private String name;
4. private int age;
5. public void say() {
6. System.out.println(age);
7. }
8. public Person(String name, int age) {
9. super();
10. this.name = name;
11. this.age = age;
12. }
13. public Person() {
14. super();
15. // TODO Auto-generated constructor stub
16. }
17. public String getName() {
18. return name;
19. }
20. public void setName(String name) {
21. this.name = name;
22. }
23. public int getAge() {
24. return age;
25. }
26. public void setAge(int age) {
27. this.age = age;
28. }
29. @Override
30. public String toString() {
31. return "Person [name=" + name + ", age=" + age + "]";
32. }
34. }

Ⅱ.编写application.xml配置文件信息

1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xsi:schemaLocation="
5. http://www.springframework.org/schema/beans
6. http://www.springframework.org/schema/beans/spring-beans-3.2.xsd ">
8. <!-- 告诉工厂 给我创建一个person对象 -->
9. <bean id="person" class="pojo.Person">
10. <property name="name" value="zhangsan"></property>
11. <property name="age" value="26"></property>
12. </bean>
13. </beans>

Ⅲ.书写测试类

1. package test;
2. import org.springframework.context.ApplicationContext;
3. import org.springframework.context.support.ClassPathXmlApplicationContext;
4. import pojo.Person;
5. public class TestPerson {
6. public static void main(String[] args) {
7. //使用核心的api初始化工厂 并 从工厂中获取对象
8. ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");
9. // 获取对象
10. Person p=(Person) context.getBean("person");
11. p.say();
12. }
13. }