**中软国际考题-笔试**

1. 单选题

1、下列哪个是Dubbo的默认注册中心（B）

A.Redis B.Zookeeper C. Multicast D. Simple

2、Dubbo有哪几种配置方式？（B）

A.1 B.2 C.3 D.4

1）Spring 配置方式

2）Java API 配置方式

3、当Dubbo服务提供层向消费层传输大数据容量的对象时，会受到Dubbo的限制，这个最大的量默认为2的（B）次方。

A.16 B.23 C.24 D.32

4、如下不是Dubbo内置服务容器的是（A）

A.EJB Container B.Log4j Container C.Jetty Container D.Spring Container

5、Dubbo中，服务提供者能实现失效踢出是什么原理（C）

A．基于MapReduce；

B．基于路由器的路由管理；

C．基于Zookeeper的临时节点原理；

D．基于阻塞原理;

6、远程方法调用指的是以下哪种技术（B）

A. JNDI B. RMI C. REST D. Socket

7、Dubbo是哪个公司开发的开源服务框架。（C）

A.谷歌 B.百度 C.阿里巴巴 D.腾讯

8、关于Dubbo协议的描述,其中错误的是(D)

A. 采用NIO复用单一长连接,减少了握手和挥手

B. 使用线程池并发处理请求,加大了并发效率

C. 在大文件传输时,单一连接会成为瓶颈

D. 它不适用于生产环境

1. 多选题
2. dubbo都支持什么协议？（A、B、C、D）

A.dubbo:// B.rmi:// C. http:// D.thrift://

2、Dubbo内置了哪几种服务容器(A、B、C)

A. Spring Container B. Jetty Container C. Log4j Container D. ORM Container

3、以下哪些是Dubbo里面的节点角色？（B、C、D）

A. Administrator B. Provider C. Consumer D. Registry

4、在 Provider 上可以配置的 Consumer 端的属性有哪些（A、B、C、D）

A、timeout：方法调用超时

B、retries：失败重试次数，默认重试 2 次

C、loadbalance：负载均衡算法，默认随机

D、actives 消费者端，最大并发调用限制

5、下面哪些是Dubbo的负载均衡策略？（A、B、D、E）

A. Random B. RoundRobin C. FirstActive

D. ConsistenHash E. LeastActive F. LastQuery

6、dubbo都支持什么协议？（A、B、C、D）

A. dubbo://

B. rmi://

C. hessian://

D. webservice://

7、下列哪些是Dubbo可选的通信框架（A，B）

A．netty

B．mina

C．https

D．telnet

8、 PRC组件主要需要借助Java的以下几个特征？（A、B 、 C）

A. 网络开发 B. 反射机制

C. 动态代理 D. 数据库操作

9、Dubbo调用方式有哪几种？（BC）

A.同步调用 B.异步调用 C.本地调用 D.异地调用

10、下列满足分布式CAP中CA特性的是？（A,C,D）

A.Zookeeper B.Oracle C.Redis D.Mongodb

1. 填空题（每空1分，共10分）
2. Dubbo有支持\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_两配置方式。(Spring 配置方式、Java API 配置方式)
3. Dubbo 2014 年开始停止维护过几年，17 年开始重新维护，并进入了\_\_\_\_\_\_\_\_\_\_\_\_\_\_项目。(Apache)
4. RPC是实现\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_结构的有力方式。( 服务化、微服务、分布式架构 )
5. Dubbo的配置方式有\_\_\_\_\_\_\_和\_\_\_\_\_\_\_。(Spring、Java API)
6. Hessian 协议用于集成 Hessian 的服务，Hessian 底层采用 通讯，采用 （ ）暴露服务，Dubbo 缺省内嵌 Jetty 作为[服务器](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)实现。（Http，servlet）
7. Java常见的修改字节码实现动态代理的库是\_\_\_\_\_\_\_\_\_\_( CGLib)
8. Dubbo框架技术的创始团队公司是\_\_\_\_\_\_\_\_\_\_（阿里巴巴）
9. Dubbo内置了哪三个服务器Spring Container、 Jetty Container、 Log4j Container。
10. Dubbo有 Spring配置方式、Java API配置方式 这个两种配置方式。
11. Dubbo内置了哪几种服务容器？：\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(Spring Container、Jetty Container,Log4j Container)
12. 判断题
13. Dubbo 提供了声明式缓存，用于加速热门数据的访问速度，以减少用户加缓存的工作量（ T ）
14. 默认是同步等待结果阻塞的，不支持异步调用。（F）
15. Dubbo不支持服务多协议吗。（ F ）
16. Dubbo 能集成 Spring Boot。（ T ）
17. Dubbo不可以对结果进行缓存。（ F ）
18. 采用Snowflake技术来生成分布式ID，该方法会受冬令时切换影响。（ F ）
19. Dubbo是基于Java RPC的高性能开源框架。（ T ）
20. Dubbo是跨语言，跨平台的微服务治理框架。（ T ）
21. Dubbo支持ZooKeeper、Redis、Nacos和Apollo等第三方注册和配置中心。（ F ）
22. 在Dubbo 2.7.0 中，通过对 URL 的改造，将注册中心拆分成了三个中心，分别是注册中心、配置中心和元数据中心。（ T ）
23. Dubbo中Monitor节点负责服务的发现和监控。（ F ）
24. Future模式可以使子线程在run方法返回值。（ T ）
25. Multicast注册中心不需要启动任何中心节点，只要广播地址一样，就可以互相发现。（ T ）
26. Dubbo主要面向接口的远程方法调用，提供智能容错和负载均衡，以及服务自动注册和发现。（ T ）
27. Dubbo的配置方式不支持注解的方法。（ F ）
28. Dubbo需要 Web 容器吗。（ F ）
29. Dubbo使用Zookeeper注册中心，没有别的选择。（ F ）
30. Dubbo 缺省协议采用单一长连接和 NIO 异步通讯，适合于小数据量大并发的服务调用，以及服务消费者机器数远大于服务提供者机器数的情况。（ T ）
31. Dubbo 缺省协议非常适合传送大数据量的服务，比如传文件，传视频等（ F ）
32. Dubbo基于 WebService 的远程调用协议，基于 [Apache CXF](http://cxf.apache.org) [1](http://dubbo.io/books/dubbo-user-book/references/protocol/webservice.html#fn_1) 的 frontend-simple 和 transports-http 实现 [，](http://dubbo.io/books/dubbo-user-book/references/protocol/webservice.html#fn_2)可以和原生 WebService 服务互操作（T）
33. Java中通常使用静态代理模式在代理处理器中完成分布式RPC调用的网络连接、数据传输和并发控制处理（F）
34. 面向服务的架构体系简称SOA（T）
35. Dubbo不需要web，如果硬要用web容器，只会增加复杂性，也浪费资源（正确）
36. Dubbo允许配置多协议，在不同服务上支持不同的协议或者同一服务上同时支持多个

协议（正确）

1. Dubbo 默认使用 Netty 框架（ T ）

五、简答题

1、Dubbo需要 Web 容器吗？

答案：不需要，如果硬要用 Web 容器，只会增加复杂性，也浪费资源。

2、Dubbo启动时如果依赖的服务不可用会怎样？

答案：Dubbo 缺省会在启动时检查依赖的服务是否可用，不可用时会抛出异常，阻止 Spring 初始化完成，默认 check="true"，可以通过 check="false" 关闭检查。

3、简要描述Dubbo 服务暴露的过程。

答案：Dubbo 会在 Spring 实例化完 bean 之后，在刷新容器最后一步发布 ContextRefreshEvent 事件的时候，通知实现了 ApplicationListener 的 ServiceBean 类进行回调 onApplicationEvent 事件方法，Dubbo 会在这个方法中调用 ServiceBean 父类 ServiceConfig 的 export 方法，而该方法真正实现了服务的（异步或者非异步）发布。

4、请简单描述Dubbo的节点及其作用。

答案：Dubbo主要有如下几个节点

| **节点** | **角色说明** |
| --- | --- |
| Provider | 暴露服务的服务提供方 |
| Consumer | 调用远程服务的服务消费方 |
| Registry | 服务注册与发现的注册中心 |
| Monitor | 统计服务的调用次数和调用时间的监控中心 |
| Container | 服务运行容器 |

5、请简单描述Dubbo的架构

答案：

1. 服务容器负责启动，加载，运行服务提供者。
2. 服务提供者在启动时，向注册中心注册自己提供的服务。
3. 服务消费者在启动时，向注册中心订阅自己所需的服务。
4. 注册中心返回服务提供者地址列表给消费者，如果有变更，注册中心将基于长连接推送变更数据给消费者。
5. 服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。
6. 服务消费者和提供者，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监控中心。



6、Dubbo是什么？

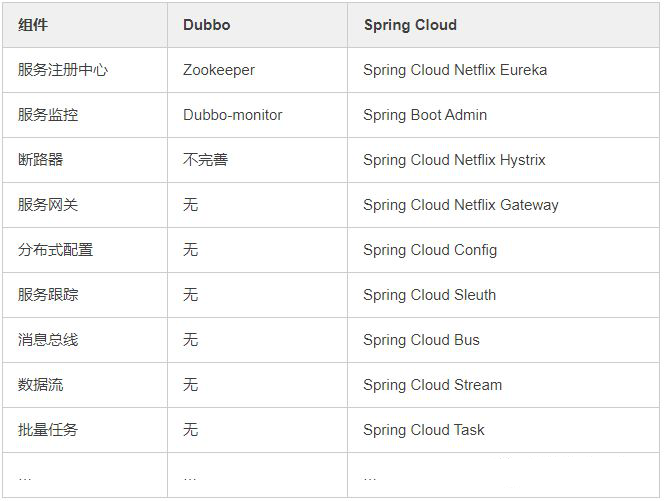
答案：Dubbo是阿里巴巴开源的基于 Java 的高性能 RPC 分布式服务框架，现已成为 Apache 基金会孵化项目。

7、Dubbo 和 Spring Cloud 有什么区别?

答案：1）通信方式不同

Dubbo 使用的是 RPC 通信，而 Spring Cloud 使用的是 HTTP RESTFul 方式。

1. 组成部分不同。



8、简述Dubbo的Http实现、特性及适用范围。

答：Dubbo基于 HTTP 表单的远程调用协议，采用 Spring 的 HttpInvoker 实现

特性

连接个数：多连接

连接方式：短连接

传输协议：HTTP

传输方式：同步传输

序列化：表单序列化

适用范围：

传入传出参数数据包大小混合，提供者比消费者个数多，可用浏览器查看，可用表单或URL传入参数，暂不支持传文件。

适用场景：

需同时给应用程序和浏览器 JS 使用的服务。

9、请简述Dubbo服务调用流程？（5分）

答案：

0.服务容器负责启动，加载，运行服务提供者。

1.服务提供者在启动时，向注册中心注册自己提供的服务。

2.服务消费者在启动时，向注册中心订阅自己所需的服务。

3.注册中心返回服务提供者地址列表给消费者，如果有变更，注册中心将基于长连接推送变更数据给消费者。

4.服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。

5.服务消费者和提供者，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监控中心

10、请简述Dubbo注册中心解决的问题？（5分）

对于服务提供方，它需要发布服务，而且由于应用系统的复杂性，服务的数量、类型也不断膨胀；

对于服务消费方，它最关心如何获取到它所需要的服务，而面对复杂的应用系统，需要管理大量的服务调用。

对于服务提供方和服务消费方来说，他们还有可能兼具这两种角色，即既需要提供服务，有需要消费服务。

11、Dubbo提供的注册中心有几种类型可供选择？（5分）

Multicast注册中心

Zookeeper注册中心

Redis注册中心

Simple注册中心

12、Dubbo优缺点？

优点：

透明化的远程方法调用

像调用本地方法一样调用远程方法；只需简单配置，没有任何API侵入

软负载均衡及容错机制

可在内网替代nginx lvs等硬件负载均衡器

服务注册中心自动注册 & 配置管理

不需要写死服务提供者地址，注册中心基于接口名自动查询提供者ip

使用类似zookeeper等分布式协调服务作为服务注册中心，可以将绝大部分项目配置移入zookeeper集群

服务接口监控与治理

Dubbo-admin与Dubbo-monitor提供了完善的服务接口管理与监控功能，针对不同应用的不同接口，可以进行 多版本，多协议，多注册中心管理

缺点：

只支持JAVA语言

13、Zookeeper是什么？

Zookeeper是一个分布式的服务框架，是树型的目录服务的数据存储，能做到集群管理数据 ，这里能很好的作为Dubbo服务的注册中心。

14、Dubbo管理控制台功能有哪些功能？

路由规则，动态配置，服务降级，访问控制，权重调整，负载均衡

15、hessian是什么？

Hessian 协议用于集成 Hessian 的服务，Hessian 底层采用 Http 通讯，采用 Servlet 暴露服务

16、Dubbo 和 Spring Cloud 有什么区别？

Dubbo 使用的是 RPC 通信，而 Spring Cloud 使用的是 HTTP RESTFul 方式

17、在Provider上可以配置的Consumer端的属性有哪些？

答案：

timeout：方法调用超时。

retries：失败重试次数，默认重试2次。

loadbalance：负载均衡算法，默认随机。

actives：消费者端，最大并发调用限制。

18、Dubbo 和 Dubbox有什么区别？

答案：

Dubbox是继Dubbo停止维护后，当当网基于Dubbo做的一个扩展项目。如：增加了服

务可以Restful调用；更新了开源组件 等。

19、**在 Provider 上可以配置的 Consumer 端的属性有哪些？**

1）timeout：方法调用超时  
2）retries：失败重试次数，默认重试 2 次  
3）loadbalance：负载均衡算法，默认随机  
4）actives 消费者端，最大并发调用限制

六、程序实现题

1、结合Spring和Dubbo实现一个简单的服务发布与消费（写出配置和实现的核心部分）

|  |
| --- |
| public interface DemoService {  /\*\*  \* 提供分布式业务的测试方法  \*/  List<String> getPermissions(Long id);  }  -----------------------------------------------------  <!--定义了提供方应用信息，用于计算依赖关系；在 dubbo-admin 或 dubbo-monitor 会显示这个名字，方便辨识-->  <dubbo:application name="demotest-provider" owner="programmer" organization="etc"/>  <!--使用 zookeeper 注册中心暴露服务，注意要先开启 zookeeper-->  <dubbo:registry address="zookeeper://localhost:2181"/>  <!-- 用dubbo协议在20880端口暴露服务 -->  <dubbo:protocol name="dubbo" port="20880" />  <!--使用 dubbo 协议实现定义好的 api.PermissionService 接口-->  <dubbo:service interface="com.chinasofti.dubbo.demo.DemoService" ref="demoService" protocol="dubbo" />  <!--具体实现该接口的 bean-->  <bean id="demoService" class="com.chinasofti.dubbo.demo.impl.DemoServiceImpl"/>  ------------------------------------------------  public class DemoServiceImpl implements DemoService {  public List<String> getPermissions(Long id) {  List<String> demo = new ArrayList<String>();  demo.add(String.format("Permission\_%d", id - 1));  demo.add(String.format("Permission\_%d", id));  demo.add(String.format("Permission\_%d", id + 1));  return demo;  }  }  --------------------------------------------------------------  public class DemoServiceImpl implements DemoService {  public List<String> getPermissions(Long id) {  List<String> demo = new ArrayList<String>();  demo.add(String.format("Permission\_%d", id - 1));  demo.add(String.format("Permission\_%d", id));  demo.add(String.format("Permission\_%d", id + 1));  return demo;  }  }  ----------------------------------------------  <dubbo:application name="demotest-consumer" owner="programmer"  organization="dubbox" />  <!--向 zookeeper 订阅 provider 的地址，由 zookeeper 定时推送 -->  <dubbo:registry address="zookeeper://localhost:2181" />  <!--使用 dubbo 协议调用定义好的 api.PermissionService 接口 -->  <dubbo:reference id="permissionService"  interface="com.chinasofti.dubbo.demo.DemoService" />  ---------------------------------------------------------  public class Consumer {  public static void main(String[] args) {  // 测试常规服务  ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("consumer.xml");  context.start();  System.out.println("consumer start");  DemoService demoService = context.getBean(DemoService.class);  System.out.println("consumer");  System.out.println(demoService.getPermissions(1L));  }  } |

2、使用示例代码实现Dubbo服务发布服务暴露。（10分）

|  |
| --- |
| dubbo应用包括三部分：服务提供者provider、服务消费者consumer、注册中心。    前两个都是自己编写的，第三个注册中心推荐使用稳定性更好的zookeeper。    所以我们需要先下载zookeeper: http://apache.fayea.com/zookeeper/current/  下载后解压到某个目录，进入里面的conf目录。将zoo\_sample.cfg复制一份，改名为zoo.cfg。  修改其内容为  Xml代码  tickTime=2000    initLimit=10    syncLimit=5    dataDir=D:\\data\\zookeeper    clientPort=2181  其中的dataDir可以自由修改。  进入zookeeper的bin目录，运行zkServer.cmd  { 要测试是否运行良好的话可以运行zkCli.cmd }      接下来打开eclipse，新建maven项目，arctype使用默认的quickstart就可以。  修改pom.xml文件：  Xml代码  <dependencies>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  <version>2.0.13</version>  </dependency>  <dependency>  <groupId>org.apache.zookeeper</groupId>  <artifactId>zookeeper</artifactId>  <version>3.3.6</version>  <exclusions>  <exclusion>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  </exclusion>  </exclusions>  </dependency>  <dependency>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  <version>1.2.16</version>  </dependency>  </dependencies>    在Java源文件夹下新建接口，这个接口要给provider和consumer都使用  Java代码  public interface DemoService {  public void sayHello();    public String returnHello();    public MsgInfo returnMsgInfo(MsgInfo info);  }  新建其实现类  Java代码  public class DemoServiceImpl implements DemoService{    public void sayHello() {  System.err.println("Hello world.");  }    public String returnHello() {  return "hello world";  }    public MsgInfo returnMsgInfo(MsgInfo info) {  info.getMsgs().add("done!");  return info;  }    }  新建一个测试实体  Java代码  public class MsgInfo implements Serializable {    int id;  String name;  List<String> msgs;    //getters setters    }    在资源文件夹下新建目录spring（虽然不是必要的，不过我们是和spring集成的）。  在里面新建xml文件dubbo-provider.xml  Xml代码  <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd">    <!-- 提供方应用信息，用于计算依赖关系 -->  <dubbo:application name="hello-world-app" />    <dubbo:registry protocol="zookeeper" address="127.0.0.1:2181" />    <!-- 用dubbo协议在20880端口暴露服务 -->  <dubbo:protocol name="dubbo" port="20880" />    <!-- 声明需要暴露的服务接口 -->  <dubbo:service interface="com.abc.qkdb.DemoService" ref="demoService" />    <!-- 和本地bean一样实现服务 -->  <bean id="demoService" class="com.abc.qkdb.DemoServiceImpl" />    </beans>  其中的<dubbo:registry>就是注册中心这里使用的是zookeeper，其他选项可以参考http://dubbo.io/Administrator+Guide-zh.htm#AdministratorGuide-zh-Redis%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85  相同目录下新建dubbo-consumer.xml  Xml代码  <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd">  <!--消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->  <dubbo:application name="consumer-of-helloworld-app" />  <!--zookeeper注册中心 -->  <dubbo:registry protocol="zookeeper" address="127.0.0.1:2181" />    <!-- 生成远程服务代理，可以和本地bean一样使用demoService-->  <dubbo:reference id="demoService" interface="com.abc.qkdb.DemoService" />  </beans>  注意里面也包含<dubbo:registry>结点。      回到Java文件夹，新建服务类  Java代码  import java.io.IOException;    import org.springframework.context.ApplicationContext;  import org.springframework.context.support.ClassPathXmlApplicationContext;    public class LuncherProvider {  public static void main(String[] args) throws InterruptedException, IOException {  LuncherProvider provider = new LuncherProvider();  provider.start();  System.in.read();  }    void start() {  String configLocation = "spring/dubbo-provider.xml";  ApplicationContext context = new ClassPathXmlApplicationContext(configLocation);  String[] names = context.getBeanDefinitionNames();  for (String name : names) {  System.err.println(name);  }  }  }  新建消费者类  Java代码  import java.util.ArrayList;  import java.util.List;    import org.springframework.context.ApplicationContext;  import org.springframework.context.support.ClassPathXmlApplicationContext;    public class LuncherConsumer {  public static void main(String[] args) {  LuncherConsumer provider = new LuncherConsumer();  provider.start();  }    void start() {  String configLocation = "spring/dubbo-consumer.xml";  ApplicationContext context = new ClassPathXmlApplicationContext(configLocation);  DemoService ds = (DemoService) context.getBean("demoService");  String[] names = context.getBeanDefinitionNames();  System.out.print("Beans:");  for (String string : names) {  System.out.println(string);  }    MsgInfo info = new MsgInfo();  info.setId(1);  info.setName("ruisheh");  List<String> msgs = new ArrayList<String>();  msgs.add("I");  msgs.add("am");  msgs.add("test");  info.setMsgs(msgs);    System.out.println(ds.returnMsgInfo(info).getMsgs());  System.err.println(ds.returnHello());  }  }    编码完毕。  先运行LuncherProvider类，没有错误输出后（会输出context中的bean）运行LuncherConsumer，输出  [I, am, test, done!]  hello world  即为正常。 |

3、根据给出的基础程序，完成Dubbo的快速启动小实例。

接口：

|  |
| --- |
| **package** org.apache.dubbo.demo;  **public** **interface** **DemoService** {  String sayHello(String name);  } |

1. 完成该接口的实现类

答案：

|  |
| --- |
| **package** org.apache.dubbo.demo.provider;    **import** org.apache.dubbo.demo.DemoService;    **public** **class** **DemoServiceImpl** **implements** **DemoService** {  **public** String **sayHello**(String name) {  **return** "Hello " + name;  }  } |

4、在下方的provider.xml配置中发布该服务，要求如下

* 应用信息：hello-world-app
* 广播地址为224.5。6.7:1234
* 设置dubbo协议在20880端口进行暴露
* 声明要暴露的服务接口，引用为demoService
* 实例本题中的实现类

答案：（出题时请去掉所有红色的部分）

|  |
| --- |
| **<?xml version="1.0" encoding="UTF-8"?>**  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:dubbo="http://dubbo.apache.org/schema/dubbo"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd http://dubbo.apache.org/schema/dubbo http://dubbo.apache.org/schema/dubbo/dubbo.xsd">    *<!-- 提供方应用信息，用于计算依赖关系 -->*  <dubbo:application name="hello-world-app" />    *<!-- 使用multicast广播注册中心暴露服务地址 -->*  <dubbo:registry address="multicast://224.5.6.7:1234" />    *<!-- 用dubbo协议在20880端口暴露服务 -->*  <dubbo:protocol name="dubbo" port="20880" />    *<!-- 声明需要暴露的服务接口 -->*  <dubbo:service interface="org.apache.dubbo.demo.DemoService" ref="demoService" />    *<!-- 和本地bean一样实现服务 -->*  <bean id="demoService" class="org.apache.dubbo.demo.provider.DemoServiceImpl" />  </beans> |

5、用代码演示Springboot如何集成dubbo?

|  |
| --- |
| Pom.xml：  <!-- Spring Boot Dubbo 依赖 -->          <dependency>              <groupId>io.dubbo.springboot</groupId>              <artifactId>spring-boot-starter-dubbo</artifactId>              <version>${dubbo-spring-boot}</version>          </dependency>  Application.properties：  ## Dubbo 服务提供者配置  spring.dubbo.application.name=provider  spring.dubbo.registry.address=zookeeper://127.0.0.1:2181  spring.dubbo.protocol.name=dubbo  spring.dubbo.protocol.port=20880  spring.dubbo.scan=org.spring.springboot.dubbo  Dubbo服务层实现类：  // 注册为 Dubbo 服务  @Service(version = "1.0.0")  public class CityDubboServiceImpl implements CityDubboService {      public City findCityByName(String cityName) {          return new City(1L,2L,"温岭","是我的故乡");      }  } |

6、、请编写一个zookeeper配置示例，配置心跳时间，LF初始通信时限，LF同步通信时限，数据文件目录，客户端连接端口，集群信息

（10分）

|  |
| --- |
| tickTime=2000  initLimit=5  syncLimit=2  dataDir=/home/michael/opt/zookeeper/data  clientPort=2181  server.1=mazhongyi05:2888:3888 |