**1 什么是单例模式？常见的创建方式？为什么能保证只有一个实例？单例模式在实际项目中的应用？**

答：1-1 整个程序中只允许有唯一的一个对象

1-2  第一种最简单，但没有考虑线程安全，在多线程时可能会出问题。（懒单例）

示例代码：

public class Singleton

{

    private static Singleton \_instance = null;

　 //1.声明一个私有方法（外部无法调用）

    private Singleton(){}

    //2.声明一个公共的静态的创建上面私有方法（实现外部可以调用到声明的私有的单例方法）

    public static Singleton CreateInstance()

    {  //如果已经存在这个对象了就不要再创建新的，直接返回该对象

　　　//如果这个对象不存在就new一个新的呗

        if(\_instance == null)

        {

            \_instance = new Singleton();

        }

        return \_instance;

    }

}

第二种考虑了线程安全 （恶汉）

public class Singleton

{

    private volatile static Singleton \_instance = null;

    private static readonly object lockHelper = new object();

    private Singleton(){}

    public static Singleton CreateInstance()

    {

        if(\_instance == null)

        {

            lock(lockHelper)

            {

                if(\_instance == null)

                     \_instance = new Singleton();

            }

        }

        return \_instance;

    }

}

1-3 private私有化修饰的静态对象

1-4

  应用1：

用枚举实现单例模式：

public enum SingletonEnum {

INSTANCE;

}

应用2：

数据库初始化加载

**2 session监听器如何实现？**

答:通过实现SessionListener类

**3 怎么实现spring事物的？特性？**

答：3-1 @Transactional(rollbackFor = ServiceException.**class**)

3-2

* 原子性（Atomicity）：事务是一个原子操作，由一系列动作组成。事务的原子性确保动作要么全部完成，要么完全不起作用。
* 一致性（Consistency）：一旦事务完成（不管成功还是失败），系统必须确保它所建模的业务处于一致的状态，而不会是部分完成部分失败。在现实中的数据不应该被破坏。
* 隔离性（Isolation）：可能有许多事务会同时处理相同的数据，因此每个事务都应该与其他事务隔离开来，防止数据损坏。
* 持久性（Durability）：一旦事务完成，无论发生什么系统错误，它的结果都不应该受到影响，这样就能从任何系统崩溃中恢复过来。通常情况下，事务的结果被写到持久化存储器中。

**4 了解动态代理吗？**

答：不用写代理类，虚拟机根据真实对象实现接口产生一个类，通过类实例化一个动态代理，在实例化动态代理时将真实对象和装备注入到动态代理中，向客户端公开的动态代理，当客户端调用动态代理方法时，动态代理根据类的反射得到真实对象的method，调用装备的invoke方法，将动态代理，Method，方法参数传于装备的invoke方法，invoke方法在唤起method方法前或后做一些处理。

**5 什么是springMVC？原理？**

答：5-1 SpringMVC框架是以请求为驱动，围绕Servlet设计，将请求发给控制器，然后通过模型对象，分派器来展示请求结果视图。其中核心类是DispatcherServlet，它是一个Servlet，顶层是实现的Servlet接口。

5-2 原理

第1步：浏览器发送指定的请求都会交给DispatcherServlet,他会委托其他模块进行真正的业务和数据处理

第2步：DispatcherServlet会查找到HandleMapping,根据浏览器的请求找到对应的Controller，并将请求交给目标Controller

第3步：目标Controller处理完业务后，返回一个ModelAndView给DispatcherServlet

第4步：DispatcherServlet通过ViewResolver视图解析器找到对应的视图对象View

第5步：视图对象View负责渲染，并返回到浏览器

**6 redis基本数据类型？**

答：list、string、hash、set、zset; 附数据操作解析:<https://blog.csdn.net/shengqianfeng/article/details/82684354>