一、Java 基础

1.JDK 和 JRE 有什么区别？

2.== 和 equals 的区别是什么？

3.两个对象的 hashCode()相同，则 equals()也一定为 true，对吗？

4.final 在 java 中有什么作用？

5.java 中的 Math.round(-1.5) 等于多少？

6.String 属于基础的数据类型吗？

7.java 中操作字符串都有哪些类？它们之间有什么区别？

8.String str="i"与 String str=new String("i")一样吗？

9.如何将字符串反转？

10.String 类的常用方法都有那些？

11.抽象类必须要有抽象方法吗？

12.普通类和抽象类有哪些区别？

13.抽象类能使用 final 修饰吗？

14.接口和抽象类有什么区别？

15.java 中 IO 流分为几种？

16.BIO、NIO、AIO 有什么区别？

17.Files的常用方法都有哪些？

二、容器

18.java 容器都有哪些？

19.Collection 和 Collections 有什么区别？

20.List、Set、Map 之间的区别是什么？

21.HashMap 和 Hashtable 有什么区别？

22.如何决定使用 HashMap 还是 TreeMap？

23.说一下 HashMap 的实现原理？

24.说一下 HashSet 的实现原理？

25.ArrayList 和 LinkedList 的区别是什么？

26.如何实现数组和 List 之间的转换？

27.ArrayList 和 Vector 的区别是什么？

28.Array 和 ArrayList 有何区别？

29.在 Queue 中 poll()和 remove()有什么区别？

30.哪些集合类是线程安全的？

31.迭代器 Iterator 是什么？

32.Iterator 怎么使用？有什么特点？

33.Iterator 和 ListIterator 有什么区别？

34.怎么确保一个集合不能被修改？

三、多线程

35.并行和并发有什么区别？

36.线程和进程的区别？

37.守护线程是什么？

38.创建线程有哪几种方式？

39.说一下 runnable 和 callable 有什么区别？

40.线程有哪些状态？

41.sleep() 和 wait() 有什么区别？

42.notify()和 notifyAll()有什么区别？

43.线程的 run()和 start()有什么区别？

44.创建线程池有哪几种方式？

45.线程池都有哪些状态？

46.线程池中 submit()和 execute()方法有什么区别？

47.在 java 程序中怎么保证多线程的运行安全？

48.多线程锁的升级原理是什么？

49.什么是死锁？

50.怎么防止死锁？

51.ThreadLocal 是什么？有哪些使用场景？

52.说一下 synchronized 底层实现原理？

53.synchronized 和 volatile 的区别是什么？

54.synchronized 和 Lock 有什么区别？

55.synchronized 和 ReentrantLock 区别是什么？

56.说一下 atomic 的原理？

四、反射

57.什么是反射？

58.什么是 java 序列化？什么情况下需要序列化？

59.动态代理是什么？有哪些应用？

60.怎么实现动态代理？

五、对象拷贝

61.为什么要使用克隆？

62.如何实现对象克隆？

63.深拷贝和浅拷贝区别是什么？

六、Java Web

64.jsp 和 servlet 有什么区别？

65.jsp 有哪些内置对象？作用分别是什么？

66.说一下 jsp 的 4 种作用域？

67.session 和 cookie 有什么区别？

68.说一下 session 的工作原理？

69.如果客户端禁止 cookie 能实现 session 还能用吗？

70.spring mvc 和 struts 的区别是什么？

71.如何避免 sql 注入？

72.什么是 XSS 攻击，如何避免？

73.什么是 CSRF 攻击，如何避免？

七、异常

74.throw 和 throws 的区别？

75.final、finally、finalize 有什么区别？

76.try-catch-finally 中哪个部分可以省略？

77.try-catch-finally 中，如果 catch 中 return 了，finally 还会执行吗？

78.常见的异常类有哪些？

八、网络

79.http 响应码 301 和 302 代表的是什么？有什么区别？

80.forward 和 redirect 的区别？

81.简述 tcp 和 udp的区别？

82.tcp 为什么要三次握手，两次不行吗？为什么？

83.说一下 tcp 粘包是怎么产生的？

84.OSI 的七层模型都有哪些？

85.get 和 post 请求有哪些区别？

86.如何实现跨域？

87.说一下 JSONP 实现原理？

九、设计模式

88.说一下你熟悉的设计模式？

89.简单工厂和抽象工厂有什么区别？

十、Hibernate

90.为什么要使用 hibernate？

1.对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。

2. Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作

3. hibernate使用Java反射机制，而不是字节码增强程序来实现透明性。

4. hibernate的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系。

91.什么是 ORM 框架？

object Relational mapping 对象关系映射，是一种为了解决面向对象与关系型数据库存在不匹配现象的技术，

简单说，orm通过描述对象和数据库之间映射的元数据，将程序中的对象自动持久化到关系型数据库中

ORM技术特点：

1、提高了开发效率，由于ORM可以自动对Entity对象与数据库表中进行字段和属性的映射，所以我们就不需要了数据访问层

2、ORM提供了对数据库的映射，不用写sql语句就可以操作数据库

ORM优缺点

优点：

提高开发效率，降低开发成本

纯粹的面向对象开发

可移植

缺点

1、从系统结构上来看，采用ORM框架一般分多层，系统的层次多了，会降低系统性能，ORM是完全面向对象的做法，而面向对象的做法也会性能产生影响

2、在数据库应用上可能被误用，主要体现在对持久对象的提取和数据加工处理上，如果用上ORM，可能将全部的数据提取到内存对象中，这就会产生性能问题

3、在做对象持久化时，ORM会持久化所有的对象，我们不希望这样，也会降低程序性能

4、在进行多级联表查询时，ORM语法会变得复杂

5、EF对各个数据的支持程度是不对称的

92.hibernate 中如何在控制台查看打印的 sql 语句？

hibernate配置文件中添加

<property name="hibernate.show\_sql" value="true"/>

<property name="hibernate.format\_sql" value="true"/>

<property name="hibernate.use\_sql\_comments" value="true"/>

93.hibernate 有几种查询方式？

HQL: Hibernate Query Language. 面向对象的写法:

QBC: Query By Criteria.(条件查询)

HQL是面向对象查询操作的，SQL是结构化查询语言 是面向数据库表结构的

94.hibernate 实体类可以被定义为 final 吗？

是的，你可以将Hibernate的实体类定义为final类，但这种做法并不好。

因为Hibernate会使用代理模式在延迟关联的情况下提高性能，如果你把实体类定义成final类之后，

因为 Java不允许对final类进行扩展，所以Hibernate就无法再使用代理了，

如此一来就限制了使用可以提升性能的手段。不过，

如果你的持久化类实现了一个接口而且在该接口中声明了所有定义于实体类中的所有public的方法轮到话，你就能够避免出现前面所说的不利后果。

95.在 hibernate 中使用 Integer 和 int 做映射有什么区别？

1、返回数据库字段值是null的话，int类型会报错。int是基本数据类型，其声明的是变量，而null则是对象。所以hibernate实体建议用integer；

2、通过jdbc将实体存储到数据库的操作通过sql语句，基本数据类型可以直接存储，对象需要序列化存储。

3、在很多应用中，需要对某些对象进行序列化，让它们离开内存空间，入住物理硬盘，以便长期保存。比如最常见的是Web服务器中的Session对象，

当有 10万用户并发访问，就有可能出现10万个Session对象，内存可能吃不消，于是Web容器就会把一些seesion先序列化到硬盘中，等要用了，再把保存在硬盘中的对象还原到内存中。

96.hibernate 是如何工作的？

1.Hibernate 的初始化.

读取Hibernate 的配置信息-〉创建SessionFactory

1)创建Configeration类的实例。

它的构造方法：将配置信息(Hibernate config.xml)读入到内存。

一个Configeration 实例代表Hibernate 所有Java类到Sql数据库映射的集合。

2)创建SessionFactory实例

把Configeration 对象中的所有配置信息拷贝到SessionFactory的缓存中。

SessionFactory的实例代表一个数据库存储员源，创建后不再与Configeration 对象关联。

缓存(cache):指Java对象的属性(通常是一些集合类型的属性－－占用内存空间。

SessionFactory的缓存中：Hibernate 配置信息。OR映射元数据。

缓存－大：重量级对象 小：轻量级对象

3)调用SessionFactory创建Session的方法

1】用户自行提供JDBC连接。

Connection con=dataSource.getConnection();

Session s=sessionFactory.openSession(con);

\*\*开启事务

2】让SessionFactory提供连接

Session s=sessionFactory.openSession();

4)通过Session 接口提供的各种方法来操纵数据库访问。

\*\*提交事务

\*\*\* 关闭session

\*\*\*关闭sessionFactory

97.get()和 load()的区别？

hibernate中get方法和load方法的根本区别在于：如果你使用load方法，

hibernate认 为该id对应的对象（数据库记录）在数据库中是一定存在的，所以它可以放心的使用，

它可以放心的使用代理来延迟加载该对象。在用到对象中的其他属性数据时 才查询数据库，但是万一数据库中不存在该记录，

那没办法，只能抛异常，所说的load方法抛异常是指在使用该对象的数据时，数据库中不存在该数据时抛异 常，

而不是在创建这个对象时。由于session中的缓存对于hibernate来说是个相当廉价的资源，所以在load时会先查一下session缓存

看看该id对应的对象是否存在，不存在则创建代理。所以如果你知道该id在数据库中一定有对应记录存在就可以使用load方法来实现延迟加载。

对于get方法，hibernate会确认一下该id对应的数据是否存在，首先在session缓存中查找，然后在二级缓存中查找，还没有就查数据库，数据库中没有就返回null。

98.说一下 hibernate 的缓存机制？

Hibernate是一个持久层框架，经常访问物理数据库，为了降低应用程序对物理数据源访问的频次，

从而提高应用程序的运行性能。缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，

在特定的时刻或事件会同步缓存和物理数据源的数据Hibernate缓存分类：

Hibernate缓存包括两大类：Hibernate一级缓存和Hibernate二级缓存Hibernate一级缓存又称为“Session的缓存”，

它是内置的，不能被卸载（不能被卸载的意思就是这种缓存不具有可选性，必须有的功能，不可以取消session缓存）。

由于Session对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。第一级缓存是必需的，

不允许而且事实上也无法卸除。在第一级缓存中，持久化类的每个实例都具有唯一的OID。 Hibernate二级缓存又称为“SessionFactory的缓存”，

由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，

因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。第二级缓存是可选的，是一个可配置的插件，在默认情况下，SessionFactory不会启用这个插件。

什么样的数据适合存放到第二级缓存中？

1 很少被修改的数据

2 不是很重要的数据，允许出现偶尔并发的数据

3 不会被并发访问的数据

4 常量数据

不适合存放到第二级缓存的数据？

1 经常被修改的数据

2 .绝对不允许出现并发访问的数据，如财务数据，绝对不允许出现并发

3 与其他应用共享的数据。

Hibernate查找对象如何应用缓存？

当Hibernate根据ID访问数据对象的时候，首先从Session一级缓存中查；查不到，如果配置了二级缓存，

那么从二级缓存中查；如果都查不到，再查询数据库，把结果按照ID放入到缓存,删除、更新、增加数据的时候，同时更新缓存。

Hibernate管理缓存实例无论何时，当你给save()、update()或saveOrUpdate()方法传递一个对象时，

或使用load()、 get()、list()、iterate() 或scroll()方法获得一个对象时, 该对象都将被加入到Session的内部缓存中。

当随后flush()方法被调用时，对象的状态会和数据库取得同步。 如果你不希望此同步操作发生，或者你正处理大量对象、需要对有效管理内存时，

你可以调用evict() 方法，从一级缓存中去掉这些对象及其集合。

99.hibernate 对象有哪些状态？

（1）瞬态：表示该实体对象在内存中是自由存在的，也就是说与数据库中的数据没有任何的关联即，该实体从未与任何持久化上下文联系过，没有持久化标识（相当与主键）。瞬态实体的特征有：

与数据库中的记录没有任何关联，也就是没有与其相关联的数据库记录 与Session没有任何关系，也就是没有通过Session对象的实例对其进行任何持久化的操作。

（2）持久态：指该实体对象处于Hibernate框架所管理的状态，也就是说这个实体对象是与Session对象的实例相关的。处于持久态的实体对象的最大特征是对其所作的任何变更操作都将被Hibernate持久化到数据库中。处于持久态的对象具有的特征为：

每个持久态对象都于一个Session对象关联

处于持久态的对象是于数据库中的记录相关联的

Hibernate会根据持久态对象的属性的变化而改变数据库中的相应记录

（3）游离态：处于持久态的实体对象，当他不再与Session对象关联时，这个对象就变成了游离态。。游离态对象的特征有：

游离态对象一定是由持久态对象转换而来

游离态实体不再于Session关联

游离态实体对象与数据库中的数据没有直接联系，主要表现在对其进行的修改不再影响到数据库中的数据

游离态实体对象在数据库中有相应的数据记录（如果该记录没有被删除）

100.在 hibernate 中 getCurrentSession 和 openSession 的区别是什么？

1. openSession 从字面上可以看得出来，是打开一个新的session对象，而且每次使用都是打开一个新的session，

假如连续使用多次，则获得的session不是同一个对象，并且使用完需要调用close方法关闭session。

2. getCurrentSession ，从字面上可以看得出来，是获取当前上下文一个session对象，当第一次使用此方法时，会自动产生一个session对象，

并且连续使用多次时，得到的session都是同一个对象，这就是与openSession的区别之一，简单而言，getCurrentSession 就是：如果有已经使用的，用旧的，如果没有，建新的。

注意：在实际开发中，往往使用getCurrentSession多，因为一般是处理同一个事务（即是使用一个数据库的情况），

所以在一般情况下比较少使用openSession或者说openSession是比较老旧的一套接口了；

101.hibernate 实体类必须要有无参构造函数吗？为什么？

Hibernate框架会调用这个默认构造方法来构造实例对象，即Class类的newInstance方法 ，这个方法就是通过调用默认构造方法来创建实例对象的 。

当查询的时候返回的实体类是一个对象实例，是Hibernate动态通过反射生成的。反射的Class.forName(“className”).newInstance()需要对应的类提供一个无参构造方法，

必须有个无参的构造方法将对象创建出来，单从Hibernate的角度讲 他是通过反射创建实体对象的 所以没有默认构造方法是不行的，另外Hibernate也可以通过有参的构造方法创建对象。

如果你没有提供任何构造方法，虚拟机会自动提供默认构造方法（无参构造器），但是如果你提供了其他有参数的构造方法的话，虚拟机就不再为你提供默认构造方法，

这时必须手动把无参构造器写在代码里，否则new Xxxx()是会报错的，所以默认的构造方法不是必须的，只在有多个构造方法时才是必须的，这里“必须”指的是“必须手动写出来”。

十一、Mybatis

102.mybatis 中 #{}和 ${}的区别是什么？

1.#{}是预编译处理，${}是字符串替换。

2.mybatis在处理#{}时，会将sql中的#{}替换为?号，调用PreparedStatement的set方法来赋值；mybatis在处理${}时，就是把${}替换成变量的值。

3.使用#{}可以有效的防止SQL注入，提高系统安全性。

103.mybatis 有几种分页方式？

数组分页

sql分页

拦截器分页

RowBounds分页

104.RowBounds 是一次性查询全部结果吗？为什么？

105.mybatis 逻辑分页和物理分页的区别是什么？

物理分页就是数据库本身提供了分页方式，如MySQL的limit，好处是效率高，不好的地方就是不同数据库有不同的搞法。

逻辑分页利用游标分页，好处是所有数据库都统一，坏处就是效率低。

106.mybatis 是否支持延迟加载？延迟加载的原理是什么？

延迟加载机制是为了避免一些无谓的性能开销而提出来的，所谓延迟加载就是当在真正需要数据的时候，才真正执行数据加载操作；

可以简单理解为，只有在使用的时候，才会发出sql语句进行查询；延迟加载的有效期是在session打开的情况下，当session关闭后，会报异常。

当调用load方法加载对象时，返回代理对象，等到真正用到对象的内容时才发出sql语句。

在mybatis核心配置文件中配置：(默认是关闭的)

lazyLoadingEnabled、aggressiveLazyLoading

107.说一下 mybatis 的一级缓存和二级缓存？

一级缓存

　　Mybatis对缓存提供支持，但是在没有配置的默认情况下，它只开启一级缓存，一级缓存只是相对于同一个SqlSession而言。

所以在参数和SQL完全一样的情况下，我们使用同一个SqlSession对象调用一个Mapper方法，往往只执行一次SQL，

因为使用SelSession第一次查询后，MyBatis会将其放在缓存中，以后再查询的时候，如果没有声明需要刷新，并且缓存没有超时的情况下，

SqlSession都会取出当前缓存的数据，而不会再次发送SQL到数据库。

二级缓存：

　　MyBatis的二级缓存是Application级别的缓存，它可以提高对数据库查询的效率，以提高应用的性能。

SqlSessionFactory层面上的二级缓存默认是不开启的，二级缓存的开席需要进行配置，实现二级缓存的时候，MyBatis要求返回的POJO必须是可序列化的。

也就是要求实现Serializable接口，配置方法很简单，只需要在映射XML文件配置就可以开启缓存了<cache/>，如果我们配置了二级缓存就意味着：

映射语句文件中的所有select语句将会被缓存。

映射语句文件中的所欲insert、update和delete语句会刷新缓存。

缓存会使用默认的Least Recently Used（LRU，最近最少使用的）算法来收回。

根据时间表，比如No Flush Interval,（CNFI没有刷新间隔），缓存不会以任何时间顺序来刷新。

缓存会存储列表集合或对象(无论查询方法返回什么)的1024个引用

缓存会被视为是read/write(可读/可写)的缓存，意味着对象检索不是共享的，而且可以安全的被调用者修改，不干扰其他调用者或线程所做的潜在修改。

108.mybatis 和 hibernate 的区别有哪些？

1 简单简介

1.1 Hibernate 框架

Hibernate是一个开放源代码的对象关系映射框架,它对JDBC进行了非常轻量级的对象封装,建立对象与数据库表的映射。是一个全自动的、完全面向对象的持久层框架。

1.2 Mybatis框架

Mybatis是一个开源对象关系映射框架，原名：ibatis,2010年由谷歌接管以后更名。是一个半自动化的持久层框架。

2 两者区别

2.1 开发方面

在项目开发过程当中，就速度而言：

hibernate开发中，sql语句已经被封装，直接可以使用，加快系统开发；

Mybatis 属于半自动化，sql需要手工完成，稍微繁琐；

但是，凡事都不是绝对的，如果对于庞大复杂的系统项目来说，发杂语句较多，选择hibernate 就不是一个好方案。

2.2 sql优化方面

Hibernate 自动生成sql,有些语句较为繁琐，会多消耗一些性能；

Mybatis 手动编写sql，可以避免不需要的查询，提高系统性能；

2.3 对象管理比对

Hibernate 是完整的对象-关系映射的框架，开发工程中，无需过多关注底层实现，只要去管理对象即可；

Mybatis 需要自行管理 映射关系；

2.4 缓存方面

相同点：

Hibernate和Mybatis的二级缓存除了采用系统默认的缓存机制外，都可以通过实现你自己的缓存或为其他第三方缓 存方案，创建适配器来完全覆盖缓存行为。

不同点：

Hibernate的二级缓存配置在SessionFactory生成的配置文件中进行详细配置，然后再在具体的表-对象映射中配置是那种缓存。

MyBatis的二级缓存配置都是在每个具体的表-对象映射中进行详细配置，这样针对不同的表可以自定义不同的缓存机制。并且Mybatis可以在命名空间中共享相同的缓存配置和实例，通过Cache-ref来实现。

比较：

Hibernate 具有良好的管理机制，用户不需要关注SQL，如果二级缓存出现脏数据，系统会保存，；

Mybatis 在使用的时候要谨慎，避免缓存CAche 的使用。

Hibernate优势

Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。

Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便。

Hibernate数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。

Hibernate有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳。

Mybatis优势

MyBatis可以进行更为细致的SQL优化，可以减少查询字段。

MyBatis容易掌握，而Hibernate门槛较高。

一句话总结

Mybatis：小巧、方便、高效、简单、直接、半自动化

Hibernate：强大、方便、高效、复杂、间接、全自动化

109.mybatis 有哪些执行器（Executor）？

SimpleExecutor、ReuseExecutor、BatchExecutor。

SimpleExecutor： 每执行一次update或select，就开启一个Statement对象，用完立刻关闭Statement对象。

ReuseExecutor：执行update或select，以sql作为key查找Statement对象，存在就使用，不存在就创建，用完后，不关闭Statement对象，而是放置于Map内，供下一次使用。

简言之，就是重复使用Statement对象。

BatchExecutor：执行update（没有select，JDBC批处理不支持select），将所有sql都添加到批处理中（addBatch()），等待统一执行（executeBatch()），

它缓存了多个Statement对象，每个Statement对象都是addBatch()完毕后，等待逐一执行executeBatch()批处理。与JDBC批处理相同。

作用范围：Executor的这些特点，都严格限制在SqlSession生命周期范围内。

110.mybatis 分页插件的实现原理是什么？

111.mybatis 如何编写一个自定义插件？

十二、Spring/Spring MVC

112.为什么要使用 spring？

为什么要学习Spring的框架

\* 方便解耦，简化开发

\* Spring就是一个大工厂，可以将所有对象创建和依赖关系维护，交给Spring管理

\* AOP编程的支持

\* Spring提供面向切面编程，可以方便的实现对程序进行权限拦截、运行监控等功能

\* 声明式事务的支持

\* 只需要通过配置就可以完成对事务的管理，而无需手动编程

\* 方便程序的测试

\* Spring对Junit4支持，可以通过注解方便的测试Spring程序

\* 方便集成各种优秀框架

\* Spring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：Struts2、Hibernate、MyBatis、Quartz等）的直接支持

\* 降低JavaEE API的使用难度

\* Spring 对JavaEE开发中非常难用的一些API（JDBC、JavaMail、远程调用等），都提供了封装，使这些API应用难度大大降低

113.解释一下什么是 aop？

1、AOP为Aspect Oriented Programming的缩写，意为：面向切面编程（也叫面向方面），

可以通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种思想。

2、利用AOP可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低 ，提高程序的可重用性，

同时提高了开发的效率。 当项目中需要编写大量相同功能重复代码时可以考虑使用AOP的形式来实现。如声明式事务管理、权限校验等。

114.解释一下什么是 ioc？

控制反转（Inversion of Control，缩写为IoC），是面向对象编程中的一种设计原则，可以用来减低计算机代码之间的耦合度。

IoC 容器：最主要是完成了完成对象的创建和依赖的管理注入等等。

先从我们自己设计这样一个视角来考虑：

DI，所谓控制反转，就是把原先我们代码里面需要实现的对象创建、依赖的代码，反转给IOC容器（或Spring框架）来实现。

那么必然的我们需要创建一个容器，同时需要一种描述来让容器知道需要创建的对象与对象的关系。这个描述最具体表现就是我们可配置的文件。

115.spring 有哪些主要模块？

Spring有七大功能模块，分别是Spring Core，AOP，ORM，DAO，MVC，WEB，Context。

1，Spring Core

Core模块是Spring的核心类库，Spring的所有功能都依赖于该类库，Core主要实现IOC功能，Sprign的所有功能都是借助IOC实现的。

2，AOP

AOP模块是Spring的AOP库，提供了AOP（拦截器）机制，并提供常用的拦截器，供用户自定义和配置。

3，ORM

Spring 的ORM模块提供对常用的ORM框架的管理和辅助支持，Spring支持常用的Hibernate，ibtas，jdao等框架的支持，

Spring本身并不对ORM进行实现，仅对常见的ORM框架进行封装，并对其进行管理

4，DAO模块

Spring 提供对JDBC的支持，对JDBC进行封装，允许JDBC使用Spring资源，并能统一管理JDBC事物，并不对JDBC进行实现。（执行sql语句）

5，WEB模块

WEB模块提供对常见框架如Struts1，WEBWORK（Struts 2），JSF的支持，Spring能够管理这些框架，将Spring的资源注入给框架，也能在这些框架的前后插入拦截器。

6，Context模块

Context模块提供框架式的Bean访问方式，其他程序可以通过Context访问Spring的Bean资源，相当于资源注入。

7，MVC模块

WEB MVC模块为Spring提供了一套轻量级的MVC实现，在Spring的开发中，我们既可以用Struts也可以用Spring自己的MVC框架，

相对于Struts，Spring自己的MVC框架更加简洁和方便。

116.spring 常用的注入方式有哪些？

Spring通过DI（依赖注入）实现IOC（控制反转），常用的注入方式主要有三种：构造方法注入，setter注入，基于注解的注入。

117.spring 中的 bean 是线程安全的吗？

2.1 我们都知道spring中的bean默认都是单例的，ioc容器中一个类只会存在一个实例对象。这种设计是怎么保证线程安全的？

一般不会出现线程安全问题。在spring中，绝大部分bean都是无状态的，因此即使这些bean默认是单例的，也不会出现线程安全问题的。

比如controller、service、dao这些类，这些类里面通常不会含有成员变量，因此它们被设计成单例的。如果这些类中定义了实例变量，

就线程不安全了，所以尽量避免定义实例变量。

2.2 对于spring中有状态的bean，比如RequestContextHolder、TransactionSynchronizationManager、

LocaleContextHolder，为什么也能够设计成单例的呢？它是怎么保证线程安全的？

对于有状态的bean，spring采用ThreadLocal进行处理，

使它们成为线程安全可以共享的对象。

对于有状态的bean，也可以使用原型模式（prototype），每次使用时都会重新生成一个对象，解决了线程不安全的问题。

118.spring 支持几种 bean 的作用域？

当通过spring容器创建一个Bean实例时，不仅可以完成Bean实例的实例化，还可以为Bean指定特定的作用域。Spring支持如下5种作用域：

singleton：单例模式，在整个Spring IoC容器中，使用singleton定义的Bean将只有一个实例

prototype：原型模式，每次通过容器的getBean方法获取prototype定义的Bean时，都将产生一个新的Bean实例

request：对于每次HTTP请求，使用request定义的Bean都将产生一个新实例，即每次HTTP请求将会产生不同的Bean实例。

只有在Web应用中使用Spring时，该作用域才有效

session：对于每次HTTP Session，使用session定义的Bean豆浆产生一个新实例。同样只有在Web应用中使用Spring时，该作用域才有效

globalsession：每个全局的HTTP Session，使用session定义的Bean都将产生一个新实例。典型情况下，

仅在使用portlet context的时候有效。同样只有在Web应用中使用Spring时，该作用域才有效

　　 其中比较常用的是singleton和prototype两种作用域。对于singleton作用域的Bean，每次请求该Bean都将获得相同的实例。容器负责跟踪Bean实例的状态，

负责维护Bean实例的生命周期行为；如果一个Bean被设置成prototype作用域，程序每次请求该id的Bean，Spring都会新建一个Bean实例，然后返回给程序。

在这种情况下，Spring容器仅仅使用new 关键字创建Bean实例，一旦创建成功，容器不在跟踪实例，也不会维护Bean实例的状态。

　　 如果不指定Bean的作用域，Spring默认使用singleton作用域。Java在创建Java实例时，需要进行内存申请；销毁实例时，需要完成垃圾回收，

这些工作都会导致系统开销的增加。因此，prototype作用域Bean的创建、销毁代价比较大。而singleton作用域的Bean实例一旦创建成功，可以重复使用。

因此，除非必要，否则尽量避免将Bean被设置成prototype作用域。

119.spring 自动装配 bean 有哪些方式？

120.spring 事务实现方式有哪些？

121.说一下 spring 的事务隔离？

事务能够始终保证数据保持在一种一致的状态，但是如果严格按照事务的定义来处理事务，那么事务的执行效率将会很低，

因为只有保证了所有事务的串行执行才能保证事务，因而在事务规范中为事务定义了四种隔离级别：Read uncommitted、Read committed、Repeatable read和Serializable。

关于这四种隔离级别，其主要区别在于三个点：脏读、不可重复读和幻读。这三个点的主要含义如下：

脏读：脏读表示一个事务能够读取另一个事务中还未提交的数据。比如，某个事务尝试插入记录A，此时该事务还未提交，然后另一个事务尝试读取记录A，

这时其是会成功读取到记录A的；

不可重复读：不可重复读表示当前事务对同一记录的两次重复读取结果不一致。比如一个事务首先读取一条记录A，读完之后另一个事务将该记录修改并且成功提交了，

然后当前事务再次读取记录A，此时该事务会发现两次读取的结果不一致；

幻读：幻读指的是一个事务在进行一次查询之后发现某个记录不存在，然后会根据这个结果进行下一步操作，此时如果另一个事务成功插入了该记录，

那么对于第一个事务而言，其进行下一步操作（比如插入该记录）的时候很可能会报错。从事务使用的角度来看，在检查一条记录不存在之后，

其进行插入应该完全没问题的，但是这里却抛出主键冲突的异常。

关于事务的四种隔离级别，其主要区别点也就在于是否能够解决这三个问题。这四种事务的隔离级别主要区别如下：

Read uncommitted：这是隔离性最低的一种隔离级别，在这种隔离级别下，当前事务能够读取到其他事务已经更改但还未提交的记录，也就是脏读；

Read committed：顾名思义，这种隔离级别只能读取到其他事务已经提交的数据，也就解决了脏读的问题，但是其无法解决不可重复读和幻读的问题；

Repeatable read：从事务的定义上，这种隔离级别能够解决脏读和不可重复读的问题，但是其无法解决幻读的问题；

Serializable：也称为序列化读，这是隔离性最高的一种隔离级别，所有的事务执行（包括查询）都会为所处理的数据加锁，操作同一数据的事务将会串行的等待。

从事务隔离级别的定义上可以看出，Serializable级别隔离性最高，但是其效率也最低，因为其要求所有操作相同记录的事务都串行的执行。这里需要说明的是，对于MySql而言，其默认事务级别是Repeatable read，虽然在定义上讲，这种隔离级别无法解决幻读的问题，但是MySql使用了一种Next key-lock的算法来实现Repeatable read，这种算法是能够解决幻读问题的。关于Next key-lock算法，在进行查询时，其不仅会将当前的操作记录锁住，也会将查询所涉及到的范围锁住。也就是说，其他事务如果想要在当前事务查询的范围内进行数据操作，那么其是会被阻塞的，因而MySql在Repeatable read隔离级别下就已经具备了Serializable隔离级别的事务隔离性。

122.说一下 spring mvc 运行流程？

1、 用户向服务器发送请求，请求被 Spring 前端控制 Servelt DispatcherServlet 捕获(捕获)

2、 DispatcherServlet对请求  URL进行解析，得到请求资源标识符（URI）。然后根据该  URI，调用 HandlerMapping获得该Handler配置的所有相关的对象（

包括  Handler对象以及   Handler对象对应的拦截器），最后以 HandlerExecutionChain对象的形式返回；(查找   handler)

3、 DispatcherServlet  根据获得的 Handler，选择一个合适的  HandlerAdapter。  提取Request 中的模型数据，填充 Handler 入参，

开始执行 Handler（Controller), Handler执行完成后，向 DispatcherServlet 返回一个 ModelAndView 对象(执行  handler)

4、DispatcherServlet  根据返回的 ModelAndView，选择一个适合的 ViewResolver（必须是已经注册到 Spring 容器中的 ViewResolver)

(选择  ViewResolver)

5、通过 ViewResolver 结合 Model 和 View，来渲染视图,DispatcherServlet 将渲染结果返回给客户端。

（渲染返回）

快速记忆技巧：

核心控制器捕获请求、查找Handler、执行Handler、选择ViewResolver,通过ViewResolver渲染视图并返回

123.spring mvc 有哪些组件？

HandlerMapping

我们可以看到HandlerMapping接口中只定义了一个方法，就是通过request找到HandlerExecutionChain，

而HandlerExecutionChain包装了一个Handler和一组Interceptors。

HandlerAdapter

之所以需要HandlerAdapter是因为Spring MVC没有对Handler做任何规定，它可以是类，可以是方法，也可以是任何其他东西，

我们可以看到Handler的类型是Object，这样会非常灵活。但是怎么让任意类型的Handler处理固定格式的请求呢？没错，就是使用适配器，

每种Handler都要有对应的HandlerAdapter才能处理请求。

HandlerExceptionResolver

在处理请求的过程中，难免会出现异常，HandlerExceptionResolver就是专门来处理异常的组件，

它能根据异常设置ModelAndView，然后交给render进行渲染。

ViewResolver

ViewResolver用来将String类型的视图名和本地化信息Local解析成View类型的视图

RequestToViewNameTranslator

ViewResolver是根据viewName查找View，

但有的Handler处理完后并没有设置View也没有设置viewName，这时就需要RequestToViewNameTranslator从request中找到默认的View了

LocalResolver

ThemeResolver

MultipartResolver

FlashMapManager

124.@RequestMapping 的作用是什么？

@RequestMapping是一个用来处理请求地址映射的注解，可用于类或者方法上。用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径

125.@Autowired 的作用是什么？

@Autowired表示被修饰的类需要注入对象,spring会扫描所有被@Autowired标注的类,然后根据 类型 在ioc容器中找到匹配的类注入

十三、MySql

126.数据库的三范式是什么？

127.一张自增表里面总共有 7 条数据，删除了最后 2 条数据，重启 mysql 数据库，又插入了一条数据，此时 id 是几？

128.如何获取当前数据库版本？

129.说一下 ACID 是什么？

130.char 和 varchar 的区别是什么？

131.float 和 double 的区别是什么？

132.mysql 的内连接、左连接、右连接有什么区别？

133.mysql 索引是怎么实现的？

134.怎么验证 mysql 的索引是否满足需求？

135.说一下数据库的事务隔离？

136.说一下 mysql 常用的引擎？

137.说一下 mysql 的行锁和表锁？

138.说一下乐观锁和悲观锁？

139.mysql 问题排查都有哪些手段？

140.如何做 mysql 的性能优化？

十四、JVM

141.说一下 jvm 的主要组成部分？及其作用？

142.说一下 jvm 运行时数据区？

143.说一下堆栈的区别？

144.队列和栈是什么？有什么区别？

145.什么是双亲委派模型？

146.说一下类加载的执行过程？

147.怎么判断对象是否可以被回收？

148.java 中都有哪些引用类型？

149.说一下 jvm 有哪些垃圾回收算法？

150.说一下 jvm 有哪些垃圾回收器？

151.详细介绍一下 CMS 垃圾回收器？

152.新生代垃圾回收器和老生代垃圾回收器都有哪些？有什么区别？

153.简述分代垃圾回收器是怎么工作的？

154.说一下 jvm 调优的工具？

155.常用的 jvm 调优的参数都有哪些？

十五、Spring Boot/Spring Cloud

156.什么是 spring boot？

157.为什么要用 spring boot？

158.spring boot 核心配置文件是什么？

159.spring boot 配置文件有哪几种类型？它们有什么区别？

160.spring boot 有哪些方式可以实现热部署？

161.jpa 和 hibernate 有什么区别？

162.什么是 spring cloud？

163.spring cloud 断路器的作用是什么？

164.spring cloud 的核心组件有哪些？

十六、Redis

165.redis 是什么？都有哪些使用场景？

166.redis 有哪些功能？

167.redis 和 memecache 有什么区别？

168.redis 为什么是单线程的？

169.什么是缓存穿透？怎么解决？

170.redis 支持的数据类型有哪些？

171.redis 支持的 java 客户端都有哪些？

172.jedis 和 redisson 有哪些区别？

173.怎么保证缓存和数据库数据的一致性？

174.redis 持久化有几种方式？

175.redis 怎么实现分布式锁？

176.redis 分布式锁有什么缺陷？

177.redis 如何做内存优化？

178.redis 淘汰策略有哪些？

179.redis 常见的性能问题有哪些？该如何解决？

十七、RabbitMQ

180.rabbitmq 的使用场景有哪些？

181.rabbitmq 有哪些重要的角色？

182.rabbitmq 有哪些重要的组件？

183.rabbitmq 中 vhost 的作用是什么？

184.rabbitmq 的消息是怎么发送的？

185.rabbitmq 怎么保证消息的稳定性？

186.rabbitmq 怎么避免消息丢失？

187.要保证消息持久化成功的条件有哪些？

188.rabbitmq 持久化有什么缺点？

189.rabbitmq 有几种广播类型？

190.rabbitmq 怎么实现延迟消息队列？

191.rabbitmq 集群有什么用？

192.rabbitmq 节点的类型有哪些？

193.rabbitmq 集群搭建需要注意哪些问题？

194.rabbitmq 每个节点是其他节点的完整拷贝吗？为什么？

195.rabbitmq 集群中唯一一个磁盘节点崩溃了会发生什么情况？

196.rabbitmq 对集群节点停止顺序有要求吗？

十八、Kafka

197.kafka 可以脱离 zookeeper 单独使用吗？为什么？

198.kafka 有几种数据保留的策略？

199.kafka 同时设置了 7 天和 10G 清除数据，到第五天的时候消息达到了 10G，这个时候 kafka 将如何处理？

200.什么情况会导致 kafka 运行变慢？

201.使用 kafka 集群需要注意什么？

十九、Zookeeper

202.zookeeper 是什么？

203.zookeeper 都有哪些功能？

204.zookeeper 有几种部署模式？

205.zookeeper 怎么保证主从节点的状态同步？

206.集群中为什么要有主节点？

207.集群中有 3 台服务器，其中一个节点宕机，这个时候 zookeeper 还可以使用吗？

208.说一下 zookeeper 的通知机制？