JavaCore

1. 面向对象和面向过程有什么区别？

* 面向过程，程式化的套路（作文，总分总，分总结构），按照计算机的思路来编程
  + 面向对象，按照人能够理解的思路来编程。每个人对类和对象的理解是不同的，需要统一思想，达成共识

1. 如何理解面向对象？它有哪些特征？

面向对象OOP,一切皆为对象，它是对现实中的具体对象进行归纳总结划分类别。面向对象主要关注对象以及对象之间的协作关系，

它是以人可以理解的思维方式来解决软件系统中的业务问题。

面向对象有以下特征：

抽象性：对具体对象进行归纳总结，划分类别。

封装性：把对象的属性和行为绑定在一起，对外以对象的形式展现，它是一种信息隐藏技术

继承性：子类继承父类，子类可以当做父类来使用，它是一种代码复用技术

多态性：多种行为，多种状态。包含参数化多态（重载）和运行时多态（重写）。

同一个接口，不同的实例执行不同的操作；提高代码的复用性和可扩展性

唯一性：每个对象唯一身份标识，用来区分不同的对象

1. 如何进行抽象？

* 划分类别（对象归类）
* 找特征（数据抽象）
* 找行为（过程抽象）

1. Java数据类型有哪些？每种类型占几个字节？每种数据类型的默认值？

* 分2种：基本类型 引用类型
* 基本数据类型，包括byte,short,int,long,float,double,boolean,char
* 引用数据类型，包括class,enum,interface,array

1. 流程控制语句有几种？

* 顺序语句，从上到下，从左到右，按照程序既定顺序执行
* 条件语句（分支语句、选择语句），包含if...else if...else... , switch...case...
* 循环语句（迭代语句）包含while循环，do...while循环，for循环，foreach循环
  + 注意continue和break含义

1. Switch使用哪些类型？

* char、byte、short、int、enum(枚举)、String

1. 重载和重写有什么区别？

* 重载：同一类中，方法名相同，参数列表不同（类型、个数、顺序），与返回值类型无关
* 重写：父子类中，方法名相同，参数列表完全相同，权限修饰符相同（子大于父）

1. Java序列化、反序列化是什么意思？如何实现序列化？为什么要序列化？

* 序列化：把对象转化为字符序列的过程 反序列化：把字节序列恢复为java对象的过程
* 实现Serializable接口、实现Externalnalizable接口
* 数据的持久化，数据保存到物理硬盘，数据在网络上传输

1. 如何理解static？

* static静态的，可以修饰方法，属性，静态块，类的所有对象共享static变量和方法，独立于对象，类专属使用，通过类来调用。

1. 如何理解final？

* 最终，一份，定义类不能被继承，定义方法不能被覆盖，定义变量不能被改变

1. ==和equals有什么区别?

* == 既可用于基本类型又可用于引用类型，equals只能用于引用类型
  + 一般情况下==比较引用地址，equals比较内容
* 自定义类没有实现equals和hashcode方法，==和equals都比较的是引用地址
* 自定义类已经实现equals和hashcode方法，==比较引用地址，equals比较内容
* 基本类型的包装类型

Byte,Short,Integer,Long,Character,Boolean实现常量池技术,在byte区间[-128,127]使用常量池,超过常量池失效

* String字符串类型，实现了常量值

1. == 既可用于基本类型又可用于引用类型，equals只能用于引用类型
2. 一般情况下==比较引用地址，equals比较内容；主要看类是否实现Object的equals和hashcode方法
3. String类型和部分基本类型的包装类型，实现常量池技术，比较类容和地址一致。【第三条可不说】
4. 抽象类和接口有什么区别？

抽象类：包含抽象方法的类，不能被实例化；可以有常量，有部分实现方法，部分定义未实现的方法。

接口：可理解为抽象的抽象类，比抽象类抽象级别更高，是一种标准、规范，有定义无实现；jdk8接口方法默认实现，必须加default关键字；

1. Java可以实现多继承吗？如果有，怎么实现？

一般情况下，就java是单继承多实现；可以侧面实现多继承，接口继承多个接口

1. String是基本数据类型吗？String可以被继承吗？

不是，不可以被继承，因为它是final的

1. 说说常用的日期类？

Date、DateFormat、SimpleDateFormat、Calendar、LocalDate、LocalTime、LocalDateTime

1. Java的IO流分几种？有哪些常用类？

流的方向：输入流、输出流；流的数据单位：字节流、字符流

常用类：File、InputStream、FileInputStream、OutputStream、FileOutputStrean、Reader、FileReader、Writer、FileWriter......

1. List和Set有什么区别？
2. ArrayList跟LinkedList区别，还有Vector？
3. HashMap和Hashtable的区别？List和 Map区别？
4. Map有什么特点？
5. 所有集合类的父类/父接口是Collection吗？

不是，是Object，Collection只包含List和Set，不包含Map

1. 数组有没有length方法？字符串有没有length方法？

数组只有length属性，字符串有length方法

1. Error 和Exception有什么区别？
2. CheckedException 和UnCheckedException有什么区别？
3. Throw和throws有什么区别？
4. Final 、finally、finalize有什么区别？
5. 进程和线程有什么区别？Java实现多线程有哪几种方法？

一个程序至少包含一个进程，一个进程可以包含多个线程。

进程：进程有独立的地址空间，可以单独运行

线程：是一个进程中的不同执行路径，线程之间没有单独的地址空间；线程不能够单独执行，它必须运行在处于活动状态的应用程序进程中

Java实现多线程两种方法：实现Runnable 接口、继承Thread类

1. Java多线程的生命周期

创建状态：new操作符创建一个新的线程对象，一个空的线程对象，系统不为它分配资源

就绪状态：执行线程的start()方法，等待操作系统调度分配必须的系统资源（时间片），排队等待，此时线程并未运行

运行状态：线程得到系统的时间片资源，运行线程的run()方法

终止状态：线程运行完成，释放资源，线程终止

阻塞状态：线程进入运行状态中执行sleep/join/wait方法，线程进入阻塞状态

线程转换：

阻塞状态中执行notify/notifyAll方法，阻塞结束，线程进入就绪状态

运行状态执行yield方法，线程进入就绪状态

1. 什么是死锁？如何避免死锁？

死锁：并发运行的多个线程间彼此等待对方锁定的临界资源、都无法运行的状态。线程之间互相调用对方，不能释放资源

避免死锁：在方法或语句块中加关键字synchronized

1. 堆和栈有什么区别？

堆：先进先出，java中存放对象，数组；优点：动态地分配内存大小；缺点：运行时动态分配内存，存取速度较慢

栈：后进先出，java中堆存储的是基本类型、对象引用、方法形参；优点：存取速度快；缺点：栈中的数据大小与生存期必须确定，缺乏灵活性

1. 值传递和引用传递？
   1. 值传递：参数传递的基本数据类型，传递的是实参值的副本。传递前后变量的值不变
   2. 引用传递，参数传递的是引用数据类型，传递的是实参的引用。传递前后变量（对象）的值会改变
2. 权限修饰符有哪些？各代表什么意思？

Public、protected 、default、private

1. JDK1.5+的新特性？
   1. 泛型
   2. 注解
   3. 可变参数
   4. 静态导入
   5. 默认实现
   6. Lambda表达式
2. Collection和Collections的区别？
3. Java解析xml有几种方式
   1. Sax 基于事件模型，一次性解析，边加载到内存边读取，解析效率高，生成xml文件比较繁琐
   2. Dom 整个xml文件的以dom树结构到内存中，可以对树结构进行上下导航操作，前期加载花费时间较多
   3. Jdom Java特定文档模型，JDOM仅使用具体类而不使用接口，性能高于dom
   4. Dom4j 前期是jdom的一个分支，DOM4J使用接口和抽象基本类方法，具有性能优异、功能强大和极端易用使用的特点，性能最高
4. 垃圾回收是什么意思？有哪些回收算法？有哪些垃圾收集器？

垃圾回收：

JAVA虚拟机（JVM）回收那些不再被引用的对象内存的过程。一般我们认为正在被引用的对象状态为“alive”,而没有被应用或者取不到引用属性 的对象状态为“dead”。垃圾回收是一个释放处于”dead”状态的对象的内存的过程。而垃圾回收的规则和算法被动态的作用于应用运行当中，自动 回收。垃圾回收一般采用分代回收策略：**Young（年轻代） 、Tenured（年老代）、Perm（持久代）**

回收算法：

1. Mark-Sweep（标记-清除）算法
2. Copying（复制）算法
3. Mark-Compact（标记-整理）算法

4.Generational Collection（分代收集）算法

垃圾回收器：

1. Serial/Serial Old
2. ParNew
3. Parallel Scavenge
4. Parallel Old
5. CMS
6. G1
7. Socket是什么意思？Java中如何实现socket编程？

定义：

网络上的两个程序通过一个双向的通信连接实现数据的交换，这个连接的一端称为一个socket，即套接字；点对点的通信

每个socket必须提供协议,地址和端口号

Java socket：

Socket类 根据服务器的IP地址和端口号实现客户端套接字

ServerSocket类：服务器套接字。服务器套接字提供IP地址和端口号等待请求通过网络传入。

WebCore

1. 说几个html5用到的标签？
2. Html中如何使用css？
3. Html中如何使用javascript？
4. 什么是jquery？
5. Jquery如何取得表单控件的值（文本框、单选、多选、下拉、文本域等）？
6. Jquery的常用选择器有哪些？
7. Json什么意思？有什么特点？Java操作json有哪些方式？

定义：官网http://www.json.org/

JSON：JavaScript 对象表示法（JavaScript Object Notation）。

JSON 是存储和交换文本信息的语法。类似 XML。

JSON 比 XML 更小、更快，更易解析。

JSON 数据的书写格式是：名称/值对 { 键 ：值}

特点：

轻量级的文本数据交换格式

独立于语言

具有自我描述性，更易理解

Java操作json的第三方库【java对象和json的互相转换】：

阿里Fastjson Yidong Fang的JSON.simple Google的GSON FasterXML的Jackson Oracle的JSONP

推荐阿里Fastjson

1. Ajax是什么意思？有什么优点？

AJAX 指异步 JavaScript 及 XML（Asynchronous JavaScript And XML）Ajax的核心是JavaScript对象XmlHttpRequest

AJAX 并非一种新的技术，而是几种原有技术的结合体:

  使用CSS和XHTML来表示。

   使用DOM模型来交互和动态显示。

   使用XMLHttpRequest来和服务器进行异步通信。

   使用javascript来绑定和调用。

优点：

异步交互，页面无刷新，用户体验好

不打断用户操作，具有后台的响应能力

服务器部分工作转移到客户端，按需取数据

主流浏览器支持，不需要插件

1. Java和javascript有什么区别？
   1. Java是sun/oracle公司,JavaScript是网景公司
   2. Java是强类型语言，JavaScript是弱类型语言
   3. Java是面向对象的，JavaScript是基于对象的
   4. Java是编译执行，JavaScript是解释执行

也可以讲细节，比如数组的区别，对象的区别

JavaWeb

1. jsp的9个隐式对象具体有哪些？
2. Jsp的4大作用域，各代表什么意思？
3. Jsp静态导入和动态导入有什么区别？
4. 跳转和重定向有什么区别？
5. Cookie和session有什么区别？
6. Servlet的生命周期？
7. 页面同步交互和异步交互的区别？

同步交互：最常见的（单击-刷新）click-refresh模式，点一个连接或提交一个表单到后台java代码，然后整个页面被刷新

异步交互：在不刷新整个页面的情况下，把页面请求提交到后台java代码，常用的技术是AJAX.

1. 分页的实现原理？

分页有两种策略：

一种是真分页，分段到数据库中取数据，强烈推荐

一种是假分页，一次性获取所有数据，大数据量不推荐。

1. 设计页面：首页、上一页、下一页、尾页
2. 写一个分页工具类，封装分页逻辑，比如每页显示多少条记录，总页数，首页、上一页、下一页、尾页等
3. Dao中实现2个方法，一个是获取总的记录数，二是分页获取列表（从哪开始，每次取多少条记录）
4. 根据页面请求的页数来进行分页查询，调用分页工具类进行分页

Database

1. 创建，修改，删除表结构的SQL语句？
2. 表数据的CRUD语句？
3. 事务有哪些特性（ACID）？你是怎么理解的？

原子性：事务包含的几个步骤，不能分割。

一致性：事务执行的结果，要不成功，要不失败；结果保持一致

隔离性：每个事务之间互不干扰

持久性：事务执行的结果，永久保存在数据库或物理硬盘中。以后可以查询之前事务执行的结果。

1. Join和union是什么意思？

Join是多表连接查询的意思，包含（inner join，left join, right join, full join, cross join),union是合并多个查询语句的结果

1. 如何来设计表的主键？

数字类型，自动编号，与业务逻辑无关

1. 主键的关键字？自动编号的关键字？

主键：Primary key 自动编号：auto\_increment(mysql)、identity（SQL Server）、sequence序列（oracle）

1. 约束有哪几种？
2. 非空约束(Not null)
3. 唯一性约束（unique）
4. 主键约束（primary key）-实体完整性
5. 外键约束（foreign key）-引用完整性
6. 检查约束（check）
7. 默认值约束（default）
8. 自动编号（auto\_increment）
9. Sql的性能优化有哪些？
   1. 索引优化
   2. 选择合适的列数据类型
   3. 尽可能使用NOT NULL
   4. 避免使用多表连接，join的字段要加索引
   5. 尽量不要用子查询
   6. 尽量在where子句中使用!=或<>操作符
   7. 慎用in 和 not in
   8. 尽量使用数字类型的字段
   9. 查询尽量避免使用\*，写出需要查询的字段
10. 数据库设计三个范式？

1NF:第一范式就是原子性，字段不可再分割；【无重复的列】

2NF:第二范式就是完全依赖，没有部分依赖；【非主属性部分依赖于主关键字，实体完整性】

3NF:第三范式就是没有传递依赖【属性不依赖于其它非主属性】

1. top N的问题（w3cschool上面有）

Mysql ：先排序oder by ，再 limit

Sql Server：先top，再排序

Oracle：先排序，再rownum

1. 什么是SQL注入？如何预防SQL注入？

SQL注入：客户端发送请求数据（表单/URL）时，将SQL指令发送到后台java代码（DAO层），后台构造SQL语句时被注入非法指令，

导致数据被破坏或泄露的风险。

预防措施：

* 页面的客户端验证
* 服务器端验证，对非法数据进行转义
* 采用预处理绑定变量，不要采用直接拼接SQL语句
* 生产环境服务器错误友好提示
* 账户权限管理
* 对敏感信息进行加密

JavaEE

1. 如何理解MVC? 说说java的分层思想？

M:model模型层 V:view视图层 C:controller控制层

分层的主要目的是，降低代码之间的耦合性，提高代码的服用性；各司其职，互不干扰。

一般分五层：

视图层：主要关注客户请求和页面数据展示

控制层：主要关注视图层与模型层的业务和数据对接，调度作用，主要是收集、处理、转发的作用

模型层【服务层SERVICE】：关注具体的业务逻辑

模型层【数据访问层DAO】:关注实体类与数据库表之间的原子操作（CRUD）

数据库层：关注数据的持久化

1. Jdbc常用的连接池有哪些？

DBCP、C3P0、Proxool、阿里的Druid

1. Struts2的工作原理是什么(原理图)？

* 客户请求经过一系列的过滤器，最后进入StrutsPreparedAndExecuteFilter过滤器（核心控制器）
* 核心过滤器根据用户请求调用ActionMapper对象，查找对应请求（url）和具体处理控制器
* 核心过滤器会创建ActionProxy对象，此对象通过配置管理器加载struts的配置文件
* 代理对象去调用ActionInvocation对象，此对象通过命令模式，负责拦截器和Action的调度
* Action执行方法返回结果生成客户端响应

1. Mybatis的工作原理(原理图)？
   1. 加载mybatis主配置文件（主配置文件包含模块的配置文件）创建SqlSessionFacory对象
   2. 根据SqlSessionFacory对象创建SqlSession对象
   3. SqlSession对象调用模块配置文件中的SQL语句，执行CRUD操作
2. Hibernate的工作原理(原理图)？
   1. 加载hibernate主配置文件（主配置文件包含模块的配置文件）创建SessionFacory对象
   2. 根据SessionFacory对象创建Session对象
   3. Session对象对实体对象（POJO）的操作，hibernate底层转化成相应的SQL语句，执行CRUD操作
3. Mybatis和hibernate的区别？  
    Mybatis：半自动ORM框架,实现了 POJO（实体类） 与 SQL 之间的映射关系，Sql Mapping实现;程序员/DBA可以优化SQL语句。

适合历史遗留项目/升级项目、执行效率要求要求比较高的项目。强项：动态的增删改查操作

Hibernate：全自动ORM框架，，实现了POJO（实体类） 和[数据库](http://lib.csdn.net/base/14" \o "MySQL知识库" \t "http://blog.csdn.net/jiuqiyuliang/article/details/_blank)表之间的映射，以及SQL的自动生成和执行,

通过对对象的操作转化成对数据库CRUD操作，O/R Mapping实现

适合全新的项目。强项：增删改操作

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

1. Struts2获取页面的值有几种方式？

* Action定义单个属性+getter/setter方法
* Action定义对象+getter/setter方法
* 模型驱动，实现ModelDriven接口
* 创建原始request对象，request.getAttribute(“xxx”);
* 创建map的request对象,request.get(“”);

1. Spring的核心思想？
   1. DI/IOC , 依赖容器注入对象，以前开发人员使用对象之前需要new一个对象，控制对象的创建过程；现在把创建对象的过程交给容器来完成，所以控制权反转了。开发人员只要告诉容器，给我一个xx对象，不需要考虑对象的创建细节。spring容器负责组装、管理对象。
   2. AOP，面向切面编程，是面向对象OOP的一个补充，AOP是在不改变现有程序代码和业务逻辑的情况下，针对现有程序的功能增强。具体给需要使用的AOP的方法追加切入点，通过配置，追加功能，底层是用反射和动态代理来完成的。
2. 如何设计一个模块？

* 先进行需求分析，了解模块所具有的功能
* 模块的数据库设计及实现
* 模块的代码设计及实现（接口、实现、页面布局）
* 模块功能的测试

1. Spring用到哪些设计模式？
   1. 工厂模式（简单工厂、工厂方法）
   2. 单例模式
   3. 代理模式
   4. 适配器模式
   5. 包装器模式
   6. 观察者模式
   7. 策略模式
   8. 模版方法
2. 常用的设计模式？

单例模式：确保一个类只有一个实例，单一实例，共享实例

简单工厂模式：对象的创建交给工厂

工厂方法模式：核心工厂定义创建对象的接口，具体创建对象的任务交给子类来实现

适配器模式：把一个类的接口变换成客户端所期待的另一种接口；有点像，USB转接口

代理模式：某一个对象提供一个代理对象，并由代理对象控制对源对象的引用。

观察者模式：定义了一种一对多的依赖关系，让多个观察者对象同时监听某一个主题对象。类似监听器

策略模式：针对一组算法，将每一个算法封装到具有共同接口的独立的类中，从而使得它们可以相互替换。

模板方法：准备一个抽象类，将部分逻辑以具体方法以及具体构造函数的形式实现，然后声明一些抽象方法来迫使子类实现剩余的逻辑

1. Spring的注入方式有哪几种？

构造注入、setter注入、p命名空间注入、注解方式注入

1. Hibernate的一级缓存和二级缓存有什么区别？

一级缓存是Session的缓存，hibernate默认配置，无须程序员操心，

二级缓存是SessionFactory的缓存，默认不开启，想开启的话需要在主配置xml文件或实体的xml中进行相关配置

1. Hibernate中get和load的区别

load方法：延迟加载的机制来加载这个对象；对象id不存在，抛出ObjectNotFoundException异常

get方法：直接通过hibernate生成sql语句操作数据库；hibernate一定要获取到真实的数据，否则返回null