一，简答题与编码题

1. 写一个Java方法，功能是可以将一个字符串反转，例如“abcde”执行后得到“edcba”；

public static void main(String[] args) {  
 String tempStr = "abcde";  
 *stringfanzhuan*(tempStr);  
  
 String tempStr1 = "";  
 *stringfanzhuan*(tempStr1);  
  
 String tempStr2 = "asdfghjkl";  
 *stringfanzhuan*(tempStr2);  
  
  
}

//忘记 反转 的英语了。。

private static void stringfanzhuan(String tempStr) {  
// 存放新 string 的生成  
 char[] charArr = new char[tempStr.length()];  
// 预防空指针  
 if (!"".equals(tempStr) && tempStr != null && tempStr.length() > 0) {  
  
// 计数作用  
 int count = 0;  
// 定位字符串的两端，用来存放到新的数组里面去  
 int left = 0;  
 int right = tempStr.length() - 1;  
  
 while (count <= tempStr.length() / 2) {  
  
 charArr[left] = tempStr.charAt(right);  
 charArr[right] = tempStr.charAt(left);  
  
 left++;  
 right--;  
  
 count++;  
 }  
 }  
 System.*out*.println(new String(charArr));  
 }

2，写一个简单的单例类（singleton）；

public class SingletonTemp {  
  
 public static SingletonTemp *singletonTemp*;  
  
 public static SingletonTemp getSingletonTemp() {  
  
 if (*singletonTemp* == null) {  
 synchronized (SingletonTemp.class) {  
 if (*singletonTemp* == null) {  
 *singletonTemp* = new SingletonTemp();  
 return *singletonTemp*;  
 }  
 }  
 }  
 return *singletonTemp*;  
 }  
  
 private SingletonTemp() {  
  
 }  
}

public class Singleton {

private volatile static Singleton singleton;

private Singleton (){}

public static Singleton getSingleton() {

if (singleton == null) {

synchronized (Singleton.class) {

if (singleton == null) {

singleton = new Singleton();

}

}

}

return singleton;

}

}

class Singleton{

//私有的构造函数，保证外类不能实例化本类

private Singleton(){}

//自己创建一个类的实例化

private static Singleton singleton = new Singleton();

//创建一个get方法，返回一个实例s

public static Singleton getInstance(){

return singleton;

}

}

1. 写出一种mysql的查询分页方法或者思路；

根据前端传递的 每页显示多少条数据 和 当前是第几页，两个数据

然后查询数据库，获得 整个数据的count条数，总共的页数

然后根据前端传递的 每页显示多少条数据 和 当前是第几页，以及 根据mysql-LIMIT方法，返回对应的数据，以list保存。

1. SQL的左连接，右连接以及内连接分别有什么特点；

左连接：以左边的所有数据去匹配右边的所有数据，左边的数据会全部保存。

右连接：以右边的所有数据去匹配左边的，右边的全部保存

内连接：先 生成两个表的全部笛卡尔积，然后根据条件进行 拆分，获得最终的结果

1. 简单介绍下spring的IOC和AOP思想；

IOC：将项目中所有的对象，全部交由spring管理，减少耦合。

经常在项目中需要在业务层，调用持久层的对象，new对象的时候两个类之间就会有耦合

基于IOC的话，将这些对象全部放在 Spring容器里面，需要的时候直接去容器里面获取，减少层与层之间的耦合。

AOP：功能之间存在重复代码的使用，或者业务代码之间存在交叉，可以基于AOP动态代理的思想，讲这些重复代码全部分隔开，减少彼此的影响，减少重复代码

假如设计一个需求，每一个方法调用都需要调用一次日志功能，以往的方法可能是在使用的时候调用日志类

这样业务功能多的话，就会重复代码过多，而且耦合度很高，修改日志类会影响这些业务代码。

可以使用AOP动态代理的思想，增强方法，在类调用方法的时候，必须调用日志模块，就可以解决这个问题。

5，介绍下servlet的生命周期，servlet是否会被多次初始化；

Servlet生命周期：每一个用户的第一次请求到达tomcat，就会创建一个servlet，调用init()方法初始化，在服务器正常关闭的时候，调用detory()方法销毁

不会，servlet是一个多线程控制的， 每一次访问都是都是从web容器里面，获取对象，每一次访问都只是调用service方法

6，HTTP响应的结构是怎么样的；

响应头：主要包含响应的协议，响应的状态码，状态码的描述

响应行：头的名称

响应体：传输的数据

如果是异步的话，返回的是json数据

8，什么是cookie？Session和cookie有什么区别；

Session是基于cookie实现的

Cookie是浏览器的会话技术，将一些用户不重要的，数据保存在浏览器端，

每个cookie有大小要求，每个网站都有个数要求。

Cookie的存活时间可以设置，默认只在一次会话里面有效

Session是服务器的会话技术，保存用户的一些重要保密数据，

服务器正常关闭的时候，会将用户数据冻化在服务器上，经过配置，可以在下一次服务器启动的时候，加载

Session的存活时间可以设置，默认保存时间为30min

9，什么是post请求？什么是get请求；post与get请求有什么区别；

Post请求：

Get请求：

Get请求的请求数据在浏览器的地址会体现，不利于保护用户数据，并且地址是由上限长度的，每个浏览器不一样的长度要求

Get请求没有请求体

以键值对的形式提交数据

post请求数据保存在请求体里面，

分为请求头，请求体，请求行

10，jsp中sendRedirect（）和forword（）方法有什么区别；

sendRedirect（）重定向

forword（）转发

1. 重定向是两次请求实现的，所以会被服务器的拦截器等拦截两次，转发是一次请求实现的
2. 重定向不可以访问服务器内部保护资源，如WEB-INF下面的，可以访问外部资源，转

发可以访问内部保护资源，不可以访问外部资源

1. 重定向使用后，浏览器的url会改变，转发不会改变

11，eval（）是做什么的；

不记得了。

12，null，undefined的区别；(js的数据类型)

Undefined：变量被声明了，但是没有赋值，就是undefined类型

调用函数时，应该提供的参数没有提供，就是undefined

对象没有赋值的属性，就是undefined

函数没有返回值，默认返回undefined

Null 表示一个对象，但是这个对象没有实例化，是空

Null是object类型，undefined是undefined类型

Undefined转换的时候是NaN，null转换的时候是0；

13，jQuery中一个对象可以同时绑定多个事件，这是如何实现的；

14，collection与collections的区别是什么；

Collection是单列集合的父类，

Collections是集合的工具类，里面有一些方法用来处理集合

15，是否可以继承String类？为什么；

不可以，因为String 是final的，不可以被继承重写。

二，数据库的题目

1:找出那些工资高于他们所在部门的平均工资的员工

2:查询表中工资最高的3个人

Table：S\_EMP;

LAST\_NAME DEPT\_ID SALARY

----------------------------------------------------------------------

Zhaitengfang 50 4750

Zhuzhengkai 41 2280

Yangyuhang 42 2375

Chenxuan 43 2090

Zhangyiyi 44 2470

Wangxiong 45 2483.3

Zhoucan 34 2897.5

Liuxiao 41 2660

Zengdian 42 2280

Jianwei 45 3625

答：

CREATE TABLE S\_EMP(

`eid` INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

`last\_name` VARCHAR(20) NOT NULL,

`dept\_id` VARCHAR(20) NOT NULL,

`salary` DOUBLE NOT NULL

)

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('0000', '45', 4750);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('111', '45', 2280);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('2222', '45', 2375);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('3333', '45', 2090);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('4444', '45', 2470);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('5555', '45', 2483.5);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('6666', '45', 2879.5);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('7777', '45', 2660);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('88888', '45', 2280);

INSERT INTO s\_emp(`last\_name`,`dept\_id`,salary) VALUES ('9999', '45', 3625);

SELECT \* FROM s\_emp WHERE salary >= (SELECT AVG(salary) FROM s\_emp)

SELECT \* FROM s\_emp ORDER BY salary DESC LIMIT 3

1,用一条SQL语句查询出各学科成绩大于80分的学生姓名（要求去掉重复记录）

2,查询出各学科的平均成绩并按分数高低进行排序

Table ：COUSE

Name kecheng fenshu

徐彪 语文 81

徐彪 数学 75

徐彪 英语 92

罗杨 语文 76

罗杨 数学 63

罗杨 英语 98

孙喆 语文 81

孙喆 数学 100

孙喆 英语 90

答：

CREATE TABLE couse(

`cid` INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

`name` VARCHAR(20) NOT NULL,

`kecheng` VARCHAR(20) NOT NULL,

`fenshu` DOUBLE NOT NULL

)

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('徐彪', '语文', 81);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('徐彪', '数学', 75);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('徐彪', '英语', 92);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('罗杨', '语文', 76);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('罗杨', '数学', 63);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('罗杨', '英语', 98);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('孙喆', '语文', 81);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('孙喆', '数学', 100);

INSERT INTO couse(`name`,`kecheng`,fenshu) VALUES ('孙喆', '英语', 90);

SELECT \* FROM couse GROUP BY `name` HAVING SUM(fenshu>80) > 2

SELECT \* FROM couse LEFT JOIN

(SELECT AVG(fenshu),`kecheng` FROM couse GROUP BY `kecheng`) couse2

ON couse.kecheng = couse2.kecheng ORDER BY fenshu DESC

三，谈人生谈理想题

1，请谈一下未来三到五年的职业规划或目标，并说明怎么样达到？

2，你觉得你的优点和缺点是什么？

3，你觉得你目前就业最大的问题是什么？（三条以上）