



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0  
公众号后台回复“面试手册”

### 1. 下列说法正确的有 ( C )

- A、class 中的 constructor 不可忽略
- B、constructor 可以作为普通方法被调用
- C、constructor 在一个对象被 new 时被调用
- D、一个 class 只能定义一个 constructor

### 2. 下列运算符合法的是 ( D )

- A、&&
- B、<>
- C、If
- D、:=

### 3. 下列哪种说法不正确 ( ABC )

- A、实例方法可以直接调用超类的实例方法
- B、实例方法可以直接调用超类的类方法
- C、实例方法可以直接调用其他类的实例方法
- D、实例方法可以直接调用本类的类方法

### 4. 执行如下程序代码后，c 的值是 ( C )

```
a=0;
c=0;
do{
--c;

=a-1;
}while(a>0);
```

- A、0 B、1 C、-1 D、死循环

5. ( × ) constructor 必须与 class 同名，但是方法不能与 class 同名

6. constructor 可以被继承，因此可以重写 Overriding，也可以被重载 Overloading

7. ( √ ) String 类是 final 类 故不可以被继承

8. ( × ) 数组的大小可以任意改变，又称动态数组

9. ( × ) try{}里有一个 return 语句，则紧跟在 try 后面的 finally{} 里的 code 将在 return 后执行

## 10. 抽象类和接口有什么不同点？

接口和抽象类的概念不一样。接口是对动作的抽象，抽象类是对根源的抽象。

抽象类表示的是，这个对象是什么。接口表示的是，这个对象能做什么。比如，男人，女人，这两个类（如果是类的话……），他们的抽象类是人。说明，他们都是人。

人可以吃东西，狗也可以吃东西，你可以把“吃东西”定义成一个接口，然后让这些类去实现它。

不同点：

参数	抽象类	接口
默认的方法实现	它可以有默认的方法实现	接口完全是抽象的。它根本不存在方法的实现
实现	子类使用 extends 关键字来继承抽象类。如果子类不是抽象类的话，它需要提供抽象类中所有声明的方法的实现。	子类使用关键字 implements 来实现接口。它需要提供接口中所有声明的方法的实现
构造器	抽象类可以有构造器	接口不能有构造器
与正常 Java 类的区别	除了你不能实例化抽象类之外，它和普通 Java 类没有任何区别	接口是完全不同的类型
访问修饰符	抽象方法可以有 public、protected 和 default 这些修饰符	接口方法默认修饰符是 public。你不可以使用其它修饰符。
main 方法	抽象方法可以有 main 方法并且我们可以运行它	接口没有 main 方法，因此我们不能运行它。
多继承	抽象方法可以继承一个类和实现多个接口	接口只可以继承一个或多个其它接口
速度	它比接口速度要快	接口是稍微有点慢的，因为它需要时间去寻找在类中实现的方法。
添加新方法	如果你往抽象类中添加新的方法，你可以给它提供默认的实现。因此你不需要改变你现在的代码。	如果你往接口中添加方法，那么你必须改变实现该接口的类

## 11.什么时候使用抽象类和接口

如果你拥有一些方法并且想让它们中的一些有默认实现，那么使用抽象类吧。

如果你想实现多重继承，那么你必须使用接口。由于 Java 不支持多继承，子类不能够继承多个类，但可以实现多个接口。因此你就可以使用接口来解决它。

如果基本功能在不断改变，那么就需要使用抽象类。如果不断改变基本功能并且使用接口，那么就需要改变所有实现了该接口的类。

## 12.sleep ( ) 和 wait ( ) 有什么不同点？

主要区别在于：

sleep 表示睡眠，wait 表示等待 sleep 需要手动唤醒线程，而 wait 不需要手动唤醒等待结束后自动执行。

## 13.TreeMap 和 HashMap 有什么不同点？

第一点：HashMap：基于哈希表实现。使用 HashMap 要求添加的键类明确定义了 hashCode()和 equals() [可以重写 hashCode()和 equals()]，为了优化 HashMap 空间的使用，您可以调优初始容量和负载因子。

TreeMap：基于红黑树实现。TreeMap 没有调优选项，因为该树总处于平衡状态。

第二点：HashMap 通过 hashCode 对其内容进行快速查找。

TreeMap 中所有的元素都保持着某种固定的顺序，如果你需要得到一个有序的结果你就应该使用 TreeMap

（HashMap 中元素的排列顺序是不固定的）。

第三点：HashMap 非线程安全

TreeMap 线程安全

第四点：HashMap：适用于在 Map 中插入、删除和定位元素。

Treemap：适用于按自然顺序或自定义顺序遍历键(key)。

总结：HashMap 通常比 TreeMap 快一点(树和哈希表的数据结构使然)，建议多使用 HashMap，在需要排序的 Map时候才用 TreeMap

## 14.throw 和 throws 有什么不同点？

throw 是语句抛出一个异常。

语法：throw (异常对象);

throw e;

throws 是方法可能抛出异常的声明。(用在声明方法时，表示该方法可能要抛出异常)

语法：(修饰符)(方法名)([参数列表])[throws(异常类)]{.....}

public void doA(int a) throws Exception1,Exception3{.....}

## 14.length ( ) 和 length 有什么不同点

java 中的 length 属性是针对数组说的,比如说你声明了一个数组,想知道这个数组的长度则用到了 length 这个属性。

java 中的 length()方法是针对字符串 String 说的,如果想看这个字符串的长度则用到 length()这个方法。

## 15.编写一个 jdbc 查询任意一个数据库（如 mysql、Oracle 等），数据库名为 test，表明为 user，只有一个字段。

```
public class MysqlDemo {
    public static void main(String[] args) throws Exception {
        Connection conn = null;
        String sql;
        // MySQL 的 JDBC URL 编写方式: jdbc:mysql://主机名称:连接端口/数据库的名称?参数=值
        // 避免中文乱码要指定 useUnicode 和 characterEncoding
        String url = "jdbc:mysql://localhost:3306/test?"
            + "user=root&password=root&useUnicode=true&characterEncoding=UTF8";
        try {
            // 之所以要使用下面这条语句，是因为要使用 MySQL 的驱动，所以我们要把它驱动起来，
            // 可以通过 Class.forName 把它加载进去，也可以通过初始化来驱动起来，下面三种形式都可以
            Class.forName("com.mysql.jdbc.Driver");// 动态加载 mysql 驱动
```

```

System.out.println("成功加载 MySQL 驱动程序");
// 一个 Connection 代表一个数据库连接
conn = DriverManager.getConnection(url);
// Statement 里面带有很多方法, 比如 executeUpdate 可以实现插入, 更新和删除等
Statement stmt = conn.createStatement();
sql = "select * from user";
ResultSet rs = stmt.executeQuery(sql);
while (rs.next()) { // 遍历 user 表中所有数据
    String name = rs.getString("name");
    System.out.println("姓名是: " + name); // 假如 user 表中的字段为 name
}
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (rs != null)
            rs.close(); // 关闭结果数据集
        if (stmt != null)
            stmt.close(); // 关闭执行环境
        if (conn != null)
            conn.close(); // 关闭数据库连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

}

}

```

注意：连接 oracle 与 mysql 不同点为，url=jdbc:Oracle:thin:@localhost:1521:orcl

driver = oracle.jdbc.driver.OracleDriver

## 16. 请编写一 socket 的程序，客户端向服务器端发送字符串，服务器端在返回相同的字符串。（类似 echo 功能）

```

public class Client { // 客户端
    public static void main(String args[]) throws Exception
    {
        String clientMessage; // 来自用户输入的的信息
        String serverMessage; // 服务器端的信息
        Socket clientSocket = new Socket("127.0.0.0", 5557); // 参数是本地地址和端口, 客户端套接
        // 字, 发起 TCP
        // 连接
        BufferedReader fromUser = new BufferedReader(new InputStreamReader(System.in)); //
        // 字符读取流,
        // 获取从键盘输入的字符
        BufferedReader fromServer = new BufferedReader(new
        InputStreamReader(clientSocket.getInputStream())); // 获取从服务器端的流, 建立套接字输入
        // 流
        DataOutputStream toServer = new
        DataOutputStream(clientSocket.getOutputStream()); // 建立套接
        // 字输出流
    }
}

```

```

clientMessage=fromUser.readLine();//读取从用户的输入
toServer.writeBytes(clientMessage);//写到服务器端
serverMessage=fromServer.readLine();//从服务器端读取
ClientSocket.close();//关闭套接字连接
}
}
public class Server { //服务端
public static void main(String args[])throws Exception
{
String clientMessage;
String serverMessage;
ServerSocket serversocket=new ServerSocket(5557);//端口要和客户端对应
System.out.println("正在监听 5555 端口");//
while(true)
{
Socket collection=serversocket.accept();//调用 accept()函数，建立 TCP 连接
DataInputStream fromClient=new DataInputStream(collection.getInputStream());
DataOutputStream toClient=new DataOutputStream(collection.getOutputStream());
clientMessage=fromClient.readUTF();//接收来自客户端的信息
toClient.writeBytes(serverMessage);//写到服务器端
System.out.println("成功建立 TCP 连接");
}
}
}
}

```

**17.写一个多线程程序，四个线程对一个 int 变量，2 个加 1，2 个减 1，输出。**

```

public class TMain {
int j=1;
public synchronized void inc(){
j++;
System.out.println(Thread.currentThread().getName()+"-inc:"+j);
}
class T1 implements Runnable{
public void run(){
inc();
}
}
public synchronized void dec(){
j--;
System.out.println(Thread.currentThread().getName()+"-dec:"+j);
}

class T11 implements Runnable{
public void run(){
dec();
}
}
public static void main(String[] args) {
TMain t = new TMain();
T1 t1 =t.new T1();
T11 t11 =t.new T11();
}
}

```

```

for(int i=0;i<2;i++){
    Thread thread=new Thread(t1);
    thread.start();

    Thread thread1=new Thread(t11);
    thread1.start();
}

}
}

```

## 18.说出常用的 10 个 linux 操作命令，至少 5 个，并简述命令的作用。

### LS 命令

- 作用：显示目录内容，类似 DOS 下的 DIR
- 格式：LS 【options】 【filename】
- 常用参数：
- >-a:all，不隐藏任何以"."字符开始的文件
- >-l：使用较长的格式列出信息
- >-r:按照文件名的逆序打印输出
- >-F:加上文件类型的指示符
- ls -lF | grep / 过滤
- man ls 查询 ls 的帮助文件

### cat 命令

- 作用：显示文件内容，concatenate 的缩写，类似 dos 的 type 命令。
- 格式：cat 【options】 【filename】
- 常用参数：
- >-n：显示文件内容的行号。
- >-b：类似-n，但是不对空白行进行编号。
- >-s：当遇到有连续两行以上的空白行时，就代换为一行的空白行。

### mv 命令

- 作用：更改文件或者目录的名字。
- 格式：mv【options】source destination
- 常用参数：
- >-f：强制模式，覆盖文件不提示。
- >-i：交互模式，当要覆盖文件的时候给提示

### rm 命令

- 作用：删除文件命令，类似 dos 的 del 命令
- 格式：rm 【options】 filenames
- 常用参数：
- >-f：强制模式，不给提示。
- >-r,-R：删除目录，recursive

## 19.说出常见的 5 个 linux 系统日志，至少 3 个并做简述日志的用途。

access-log 纪录 HTTP/web 的传输  
acct/pacct 纪录用户命令  
aculog 纪录 MODEM 的活动  
btmpt 纪录失败的纪录  
lastlog 纪录最近几次成功登录的事件和最后一次不成功的登录  
messages 从 syslog 中记录信息 ( 有的链接到 syslog 文件 )  
sudolog 纪录使用 sudo 发出的命令  
sulog 纪录使用 su 命令的使用  
syslog 从 syslog 中记录信息 ( 通常链接到 messages 文件 )  
utmp 纪录当前登录的每个用户  
wtmp 一个用户每次登录进入和退出时间的永久纪录  
xferlog 纪录 FTP 会话

## 20.数据库试题

创建一张员工表,表明 EMPLOYEES,有四个字段, EMPLOYEE\_ID:员工表 ( 主键 )、 DEPT\_ID:部门号、

EMPLOYEE\_NAME:员工姓名、 EMPLOYEE\_SALARY:员工工资。

问题 1、写出建表语句

```
CREATE TABLE EMPLOYEES(  
  EMPLOYEE_ID int not null primary key,  
  DEPT_ID int,  
  EMPLOYEE_NAME char(40),  
  EMPLOYEE_SALARY double  
)
```

问题 2、检索出员工工资最高的员工姓名和工资

```
select * from user where employee_salary= (select max(employee_salary) from  
user)
```

问题 3、检索出部门中员工最多的部门号 and 此部门员工数量

```
select dept_id,count(*) cno from user GROUP BY dept_id desc limit 1
```

## 21.j2ee 中的应用服务器有哪些 ?

- 1 ) Weblogic
- 2 ) Tomcat
- 3 ) JBoss
- 4 ) WebSphere
- 5 ) IIS

## 22.EJB 程序与普通的 java 程序区别有哪些 ?



EJB 是 sun 的服务器端组件模型，最大的用处是部署分布式应用程序当然,还有许多方式可以实现分布式应用，类

似微软的.net 技术。凭借 java 跨平台的优势，用 EJB 技术部署的分布式系统可以不限于特定的平台。

EJB

(EnterpriseJavaBean)是 J2EE 的一部分，定义了一个用于开发基于组件的企业多重应用程序的标准。其特点包括网络服务支持和核心开发工具(SDK)。在 J2EE 里， Enterprise Java Beans(EJB)称为 Java 企业 Bean，是 Java 的核心代码，分别是会话 Bean ( Session Bean )，实体 Bean ( Entity Bean ) 和消息驱动 Bean ( MessageDriven Bean )。

简单来讲：比如做一个工程就和盖房子，如果，你会 java，那么你就拥有了基本的技能，一步一步累砖，总能把

房子盖好但是 EJB 就是一个框架，盖房子的时候，先有这个框架，然后你根据这个框架去累砖，房子就会盖的又快又好。java 是基础，EJB 是在 java 上发展出来的模型，框架。

## 23.请简述什么是集群？ ---了解就可以

服务器集群就是指将很多服务器集中起来一起进行同一种服务，在客户端看来就象是只有一个服务器。集群可

以利用多个计算机进行并行计算从而获得很高的计算速度，也可以用多个计算机做备份，从而使得任何一个机器坏了整个系统还是能正常运行。一旦在服务器上安装并运行了群集服务，该服务器即可加入群集。群集化操作可以减少单点故障数量，并且实现了群集化资源的高可用性。下述各节简要介绍了群集创建和群集操作中的节点行为。)



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜



Java研发军团

Java研发军团《Java面试手册》V1.0  
公众号后台回复“面试手册”