四．JAVA 面试题

1.Java 有那些基本数据类型，String 是不是基本数据类型，他们有何区别。

Java 语言提供了八种基本类型：

六种数字类型（四个整数型，两个浮点型）

字节型 byte 8 位 短整型 short 16 位 整型 int 32 位

长整型 long 64 位 单精度 float 32 位 双精度 double 64 位

一种字符类型

字符型 char 8 位

还有一种布尔型

布尔型：boolean 8 位 可存储"True"和"false"。

String 本身就是一个对象而不是基本数据类型，String 的变量名是对 String 类的引用。

2.字符串的操作：

写一个方法，实现字符串的反转，如：输入 abc，输出 cba

```
public static String reverse(String s){
        int length=s.length();
        StringBuffer result=new StringBuffer(length);
        for(int i=length-1;i>=0;i--)
            result.append(s.charAt(i));
        return result.toString();
    }
```

写一个方法，实现字符串的替换，如：输入 bbbwlirbbb，输出 bbbhhtccc。

```
String s = "bbbwlirbbb";
s.replaceAll("wlirbbb","hhtccc");
```

3.数据类型之间的转换

如何将数值型字符转换为数字（Integer，Double）

使用 Integer.parseInt()和 Double.parseDouble()方法。

如何将数字转换为字符

```
class my
{
int a=12;
Label label=new Label();
label.setText(String.valueOf(a));
}
```

如何取小数点前两位，并四舍五入。

```
System.out.println("四舍五入取整:(3.856)="
    + new BigDecimal(i).setScale(2, BigDecimal.ROUND_HALF_UP));
```

4.日期和时间

如何取得年月日，小时分秒

```
Date dat=new Date();
dat.getYear();  dat.getMonth();  dat.getDay();  dat.getHours();  dat.getMinutes();
dat.getSeconds();
```

如何取得从 1970 年到现在的毫秒数

long now=dat.getTime();

如何获取某个日期是当月的最后一天

DateFormate df=DateFormate.getInstance();df.Format(dat);

如何格式化日期

DateFormate df=DateFormate.getInstance();

df.Format(dat);

5.数组和集合

数组与集合的区别:一：数组声明了它容纳的元素的类型，而集合不声明。这是由于集合以 object 形式来存储它们的元素。二：一个数组实例具有固定的大小，不能伸缩。集合则可根据需要动态改变大小。三：数组是一种可读/可写数据结构没有办法创建一个只读数组。然而可以使用集合提供的 ReadOnly 方　只读方式来使用集合。该方法将返回一个集合的只读版本。

6.文件和目录（I/O）操作

如何列出某个目录下的所有文件

如何列出某个目录下的所有子目录

判断一个文件或目录是否存在

如何读写文件

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
public class FileOpreate {

    public static void main(String [] args)
    {
        FileOpreate fo=new FileOpreate();
        try
        {

        //文件的写入
        /*String[] testStr=new String[50];
        for(int i=0;i<testStr.length;i++)
        {
          testStr[i]="我的测试数据 00"+i;
        }
        fo.writeFile("D:\\","test.txt",testStr);
        */

        //文件的写入
        /*String str="测试";
```

```
        fo.writeFile("D:\\","test001.txt",str);*/

        //创建或者删除文件
        /*if(fo.createAndDeleteFile("D:\\","test001.txt"))
        {
            fo.createAndDeleteFile("D:\\","test002.txt");
        }*/

        //创建或者删除文件夹
        /*fo.createAndDeleteFolder("D:\\","2009-07-06");*/

        //输出一个文件内的文件名称
        /*fo.readFolderByFile("D:\\");*/


        //判断一个文件是否是空的
        /*fo.fileIsNull("D:\\","test002.txt");*/

        //读取全部的文件内容
        /*fo.readAllFile("D:\\","test.txt");*/

        //一行一行读取文件内容
        fo.readLineFile("D:\\","test.txt");
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }

    }
```

7.Java 多态的实现（继承、重载、覆盖）

多态可分为：

1)编译多态：主要是体现在重载，系统在编译时就能确定调用重载函数的哪个版本。

2)运行多态：主要体现在 OO 设计的继承性上，子类的对象也是父类的对象，即上溯造型，所以子类对象可以作为父类对象使用，父类的对象变量可以指向子类对象。因此通过一个父类发出的方法调用可能执行的是方法在父类中的实现，也可能是某个子类中的实现，它是由运行时刻具体的对象类型决定的。

8.编码转换，怎样实现将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串。

```
String str = new String("字符串".getBytes("GB2312"),"ISO-8859-1");
```

9.使用 StringBuffer 类与 String 类进行字符串连接时有何区别？

JAVA 平台提供了两个类：String 和 StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个 String 类提供了数值不可改变的字符串。而这个 StringBuffer 类提供的字符串进行修改。当你知道字符数据要改变的时候你就可以使用 StringBuffer。典型地，你可以使用 StringBuffers 来动态构造字符数据。

10.谈谈 final, finally, finalize 的区别。

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

11.String s = new String("xyz");创建了几个 String Object?

两个，一个是 string s，另一个是”xyz”。

12.Java 有没有 goto?

goto 是 java 中的保留字，现在没有在 java 中使用。

13.数组有没有 length()这个方法? String 有没有 length()这个方法？

数组没有 length()这个方法，有 length 的属性。String 有 length()这个方法。

14.Overload 和 Override 的区别。

override（重写）

1）方法名、参数、返回值相同。

2）子类方法不能缩小父类方法的访问权限。

3）子类方法不能抛出比父类方法更多的异常(但子类方法可以不抛出异常)。

4）存在于父类和子类之间。

5）方法被定义为 final 不能被重写。

overload（重载）

1）参数类型、个数、顺序至少有一个不相同。

2）不能重载只有返回值不同的方法名。

3）存在于父类和子类、同类中。

15.abstract class 和 interface 有什么区别?

Interface 只能有成员常量，只能是方法的声明；

而 abstract class 可以有成员变量，可以声明普通方法和抽象方法。

声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建 abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 static final 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，instanceof 运算符可以用来决定某对象的类是否实现了接口。

16.是否可以继承 String 类?

String 类是 final 类故不可以继承。

17.面向对象的特征有哪些方面?

1）抽象：抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2）继承：继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。

3)封装：封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

4) 多态性：多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

18.int 和 Integer 有什么区别?

Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。

Int 是 java 的原始数据类型，

Integer 是 java 为 int 提供的封装类。

Java 为每个原始类型提供了封装类。

原始类型封装类 booleanBoolean charCharacter byteByte shortShort intInteger longLong floatFloat doubleDouble

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。

19.作用域 public,private,protected,以及不写时的区别?

访问	Public	Protected	缺省的	Private
同类	√	√	√	√
同包	√	√	√	×
子类	√	√	×	×
通用性	√	×	×	×

20.用 java 写一个冒泡排序。

```
    /**
* 对 int 数组进行升序排序
*
* @param intVal：要排序的数组
* @param asc：值为 true，表示升序排序；传入值为 false，表示降序排序
* @return 返回排序后的 int 数组
*/
public static int[] intArraySort(int [] intVal,boolean asc){
    int [] vals=intVal;
    int temp;
    if(vals.length>0){
     if(asc==true){
      for(int i=0;i<=vals.length-2;i++){
```

```
            for(int j=0;j<vals.length-i-1;j++){
              if(vals[j]>vals[j+1]){
                //升序排列
                temp=vals[j];
                vals[j]=vals[j+1];
                vals[j+1]=temp;
              }
             }
            }
           }else{
            for(int i=0;i<=vals.length-2;i++){
              for(int j=0;j<vals.length-i-1;j++){
                if(vals[j]<vals[j+1]){
                  //降序排列
                  temp=vals[j];
                  vals[j]=vals[j+1];
                  vals[j+1]=temp;
                }
               }
              }
             }
           }
           return vals;
      }
```

21.short s1 =1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错?

short s1 = 1; s1 = s1 + 1; （s1+1 运算结果是 int 型，需要强制转换类型） short s1 = 1; s1 += 1;（可以正确编译）

22.float 型 float f=3.4 是否正确?

不正确。精度不准确,应该用强制类型转换，如下所示：float f=(float)3.4 或 float f = 3.4f 在 java 里面，没小数点的默认是 int,有小数点的默认是 double;

23.ArrayList 和 Vector 的区别,HashMap 和 Hashtable 的区别。

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差。

HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key）,由于非线程安全，效率上可能高于 Hashtable。

HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。

HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。 Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。

最大的不同是，Hashtable 的方法是 Synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供外同步。

24.Error 与 Exception 有什么区别？

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

25.statement 和 preparestatement 区别

1）创建时的区别：

Statement stm=con.createStatement();

PreparedStatement pstm=con.prepareStatement(sql);

执行的时候:

stm.execute(sql);

pstm.execute();

2）pstm 一旦绑定了 SQL,此 pstm 就不能执行其他的 Sql，即只能执行一条 SQL 命令。stm 可以执行多条 SQL 命令。

3）对于执行同构的 sql（只有值不同，其他结构都相同）,用 pstm 的执行效率比较的高，对于异构的 SQL 语句，Statement 的执行效率要高。

4）当需要外部变量的时候，pstm 的执行效率更高.

26.写一个数据库连接类，包括查询及结果存储。

```
import java.sql.*;        //需要加入的包
//类的定义
class DBConnect {
    private String con1 = "sun.jdbc.odbc.JdbcOdbcDriver"; //连接 SQL 所需要的字符串
    private String url = "jdbc:odbc:test";
    private String user = "sa", password = ""; //这里请根据您的数据库用户和密码自行修改
    Connection con;                 //用于连接数据库用的
    PreparedStatement ps;           //其实用 Statement 也行,PreparedStatement 集成了 Statement.
    ResultSet rs;                   //一个集合,可以用于执行 SQL 命令
    //构造函数
    DBConnect() {
        try {
            Class.forName(con1); //Class.forName()用于将一些类加载到 JVM
            this.Connect();     //函数调用
            try {
                this.execute();
            } catch (SQLException ex) {
                System.out.println(ex.toString());
            }
        } catch (ClassNotFoundException ce) {
            System.out.println(ce);
        }
    }
    public void Connect() {
```

```
            try {
                con = DriverManager.getConnection(url, user, password); //做这部之前先把
ODBC 配置好
                if (con != null) {
                    System.out.println("Connection Sucessfully!");
                }
            } catch (SQLException ex) {
                System.out.println(ex.toString());
            }
        }
        public void execute() throws SQLException {
            ps = con.prepareStatement("select *from friends"); //把 SQL 语句搞给 ps
            rs = ps.executeQuery();                                    //这里执行,之后让 rs 知
道信息
            while (rs.next())                                          //这里必须加 next(),偏移
量移动.
            {
                System.out.print(rs.getString(2) + "\t");
                System.out.print(rs.getString(3) + "\t");
                System.out.print(rs.getString(4) + "\t");
                System.out.print(rs.getDate(5) + "\t");
                System.out.print(rs.getInt(6) + "\t");
                System.out.println("");
            }
        }
        public void close()                                    //用于释放资源,Java 里没有析构函数,
但是通过重写 protected void finalize(),
        {                                                                  // 之 后 在 调 用
System.runFinalization()和 System.gc()可以提醒 JVM 执行 finalize()以释放,
            try{                                                //在以前的 J2SE 版本里可以通过以
上方法调用 finalize(),但目前的 J2SE5.0 只能提醒 JVM,但 JVM 不一定执行
            rs.close();                                        //最好的方案还是自己写析构 close();
            ps.close();
            con.close();
            }catch(SQLException ce)
            {
                System.out.println(ce.toString());
            }
            System.out.println("Connection released!!!");
        }
        public static void main(String[] args) {
            DBConnect dbc=new DBConnect();
            dbc.close();
        }
```

```
}
```