

一、填空题 1、Java 的三大体系分别是_JavaSE____、_JavaEE____、_JavaME____。

2、Java 程序的运行环境简称为 JRE____。 3、编译 Java 程序需要使用 javac____ 命令。
4、javac.exe 和 java.exe 两个可执行程序存放在 JDK 安装目录的 bin____ 目录下。 5、
path____ 环境变量用来存储 Java 的编译和运行工具所在的路径，而 classpath____ 环境变量
则用来保存保存 Java 虚拟机要运行的“.class”文件路径。

二、选择题

1、以下选项中，哪些属于 JDK 工具？（多选）ABCD

A、Java 编译器 B、Java 运行工具 C、Java 文档生成工具 D、Java 打包工具

2、Java 属于以下哪种语言？C A、机器语言 B、汇编语言 C、高级语言

3、下面哪种类型的文件可以在 Java 虚拟机中运行？D D、.class

4、安装好 JDK 后，在其 bin 目录下有许多 exe 可执行文件，其中 java.exe 命令的作用是以下
哪一种？B A、Java 文档制作工具 B、Java 解释器 C、Java 编译器

5、如果 jdk 的安装路径为：d:\jdk，若想在命令窗口中任何当前路径下，都可以直接
使用 javac 和 java 命令，需要将环境变量 path 设置为以下哪个选项？B A. d:\jdk;
B. d:\jdk\bin; C. d:\jre\bin; D. d:\jre;

三、问答题

1、简述 Java 的特点。面向对象、跨平台性、健壮性、安全性、可移植性、多线程性、动
态性等。

2、简述 JRE 与 JDK 的区别 JRE（Java Runtime Environment，Java 运行时环境），它相
当于操作系统部分，提供了 Java 程序运行时所需要的基本条件和许多 Java 基础类，例如，
IO 类、GUI 控件类、网络类等。JRE 是提供给普通用户使用的，如果你只想运行别人开发好
的 Java 程序，那么，你的计算机上必须且只需安装 JRE。 JDK（Java Development Kit，Java
开发工具包），它包含编译工具、解释工具、文档制作工具、打包工具多种与开发相关的工
具，是提供给 Java 开发人员使用的。初学者学习和使用 Java 语言时，首先必须下载和安装
JDK。JDK 中已经包含了 JRE 部分，初学者安装 JDK 后不必再去下载和安装 JRE 了。

第二章

、Java 语言的八种基本数据类型有：byte 字节型，占一个字节。short 短整型，占两个字节。
int 整型，占 4 个字节。long 长整型，占 8 个字节。float 单精度浮点型，占 4 个字节。double
双精度浮点型，占 8 个字节。char 字符型，占两个字节。boolean 型，表示逻辑值，有 true
和 false 两个值，分别占一个字节。

2、如果使用“&”在表达式之间进行连接，那么无论任何情况，“&”两边的表达式都会参
与计算。如果使用“&&”进行连接，当“&&”左边的表达式为 false，则不会执行其右边的
表达式。例如定义 int x = 2, y = 0; boolean b = x < y & x / 2 > 0 表达是会发生被 0 除异常，因为
x / y 的表达式执行了。而 boolean b = x < y & x / 2 > 0 是不会出现这种异常的，因为 x < y 为
false，表达式 x / y 不会执行。

3、方法重载指的是在一个类中可以声明多个同名的方法，而方法中参数的个数或者数据类
型不一致。调用这些同名的方法时，JVM 会根据实际参数的不同绑定到不同的方法

冒泡排序：

```
public class Test03 { public static void main(String[] args) { int[] arr = { 25, 24, 12, 76, 101,
96, 28 }; for (int i = 0; i < arr.length - 1; i++) { // 定义内层循环 for (int j = 0; j <
arr.length - i - 1; j++) { if (arr[j] > arr[j + 1]) { // 比较相邻元素 // 下面的三行代码用
于交换两个元素 int temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] =
temp; } } } for (int i = 0; i < arr.length; i++) { System.out.print(arr[i] + " ");
// 打印元素和空格 } }
```

第三章

1、**构造方法**是类的一个特殊成员，它会在类实例化对象时被自动调用。而普通方法只有在使用的时 候才会被调用。在定义构造方法时要求方法名与类名相同、在方法名的前面没有返回值类型的声 明、在方法中不能使用 `return` 语句返回一个值

2、**单例模式**可以保证在整个程序运行期间针对该类只存在一个实例对象。

Java 中的变量可分为两种数据类型，分别是_基本数据类型_____和_引用数据类型_____。 5、在 Java 中，`byte` 类型数据占___1___个字节，`short` 类型数据占___2___个字节，`int` 类型数据占___4___个字节，`long` 类型数据占___8___个字节。

面向对象的三大特征是封装性_、_继承性_____和_多态性_____。 2、在 Java 中，可以使用关键字___`new`___来创建类的实例对象。 3、定义在类中的变量被称为_成员变量___，定义在方法中的变量被称为局部变量_____ 4、面向对象程序设计的重点是_类___的设计，_类___是用来创建对象的模板。 5、在非静态成员方法中，可以使用关键字___`this`___访问类的其它非静态成员。 6、当一个对象被当成垃圾从内存中释放时，它的___`finalize()`___方法会被自动调用。 7、被 `static` 关键字修饰的成员变量被称为_静态变量_，它可以被该类所有的实例对象共享。 8、在一个类中，除了可以定义属性、方法，还可以定义类，这样的类被称为_内部类_____。 9、在 Java 中，提供了一个___`javadoc`___命令，用于将程序中的文档注释提取出来，生成 HTML 格式的文档。 10、所谓类的封装是指在定义一个类时，将类中的属性私有化，即使用___`private`___关键字来修饰。

第四章：

在 Java 语言中，允许使用已存在的类作为基础创建新的类，这种技术我们称为继承_____。 2、一个类**如果实现一个接口**，那么它就需要实现接口中定义的全部_方法___，否则该类就必须定义成_抽象类_____。 3、在程序开发中，要想将一个包中的类导入到当前程序中，可以使用___`import`___关键字。 4、一个类可以从其它的类派生出来，派生出来的类称为_子类_____，用于派生的类称为_基类_____或者_父类_____。 5、JDK 中定义了大量的**异常类**，这些类都是_Exception_____类的子类或者间接子类。 6、定义一个 Java 类时，如果前面加使用___`final`___关键字修饰，那么该类不可以被继承。 7、如果子类想使用父类中的成员，可以通过关键字___`super`___引用父类的成员。 8、在 Java 语言中，所有的类都直接或间接继承自_Object_____类。 9、异常的捕获通常由 `try`、`catch` 两部分组成，___`try`___代码块用来存放可能发生异常，___`catch`___代码块用 来处理产生的异常。 10、在 Java 语言中，要想是将一个已编译的类打包成 `jar` 文件，可以使用的命令是___`jar-cvf`___，要 想在命令行窗口运行 `jar` 文件，可以使用的命令是___`java-jar`___。

1、什么是**方法重写**？ 在继承关系中，子类的方法与父类的某一方法具有相同的方法名、返回类型和参数列表，则称子类的该方法重写(覆盖)父类的方法。 2、什么是**多态**？ 多态意味着一个对象有着多种形态，可以在特定的情况下，表现不同的状态，从而对应着不同的属性和方法。简单的说，多态就是使用父类类型的变量引用子类对象，根据被引用子类对象的特性，程序会得到不同的运行效果。 3、**抽象类和接口的区别**？

在 Java 中，使用 `abstract` 关键字修饰的类称之为抽象类。抽象类是不能被实例化的，通常需要写一个子类来继承抽象类，同时实例化子类来获得该类的对象。抽象类通常用于表示一种抽象的概念。接口可以说是一种特殊的抽象类，接口中只能定义常量和抽象方法。由于接口的特殊性，在定义时需要使用 `interface` 关键字。

第六章:

1、在 Java 中定义了两个类来封装对字符串的操作，它们分别是 `String` 和 `StringBuffer`。 2、Java 中的操作日期的类有 `Date`、`Calendar`、`DateFormat` 等。 3、在程序中若想取得一个 `Runtime` 实例，则可以调用 `Runtime` 类的静态方法 `getRuntime()`。 4、`Math` 类中用于计算所传递参数平方根的方法是 `sqrt()`。 5、Java 中专门用于将日期格式化为字符串的类是 `DateFormat`。 6、`Math` 类中有两个静态常量 `PI` 和 `E`，分别代表数学常量 π 和 `e`。 7、Java 中的用于产生随机数的类是 `Random`，它位于 `java.util` 包中。 8、`String` 类中用于返回字符串字符个数的方法是 `length()`。 9、`System` 类中所提供的属性和方法都是静态的，想要引用这些属性和方法，直接使用 `System` 类调用即可。 10、已知 `sb` 为 `StringBuffer` 的一个实例，且 `sb.toString()` 的值为 "abcde"，则执行 `sb.reverse()` 后，`sb.toString()` 的值为 "edcba"。

1、`String` 和 `StringBuffer` 有什么区别？ `String` 类是不可变类，即字符串值一旦初始化后就不可能改变。`StringBuffer` 是可变字符串类，类似 `String` 的缓冲区，可以修改字符串的值。 2、`Date` 和 `Calendar` 类有什么区别和联系？ `Date` 类用来表示某个特定的瞬间，能够精确到毫秒。而在实际应用中，往往需要把一个日期中的年、月、日等信息单独返回进行显示或处理，这个类中的大部分方法都已被标记过时。`Calendar` 类基本取代了 `Date` 类，该类中定义了一系列用于完成日期和时间字段操作的方法。`Calendar` 的 `getTime()` 方法，`getTime()` 返回一个表示 `Calendar` 时间值的 `Date` 对象，同时 `Calendar` 有一个 `setTime(Date date)` 方法，`setTime()` 方法接收一个 `Date` 对象，将 `Date` 对象表示的时间值设置给 `Calendar` 对象，通过这两个方法就可以完成 `Date` 和 `Calendar` 对象之间的转换。

I/O

1、Java 中的 IO 流，按照传输数据不同，可分为字节流和字符流。 2、在 Java 中，`File` 类用于操作磁盘中文件和目录，位于 `java.io` 包中。 3、在 Java 中，`File` 类用来把两个或更多的 `InputStream` 输入流对象合并为单个 `InputStream` 输入流对象使用。 4、Java 中提供了一个类 `File`，它不但具有读写文件的功能，并且可以随机地从文件的任何位置开始执行读写数据的操作。 5、在 Java 中，能实现线程间通信的流是 `PipedInputStream`。 6、Java 中提供了一个可以在读文件的同时记录行号的类，这个类是 `LineNumberReader`，它是 `BufferedReader` 的直接子类，它通过 `setLineNumber()` 和 `getLineNumber()` 方法设置和获取当前行号。 7、`InputStreamReader` 类是用于将 `InputStream` 转换为 `Reader`。 8、`System.out` 是 `PrintStream` 类的对象，称为标准输出流，调用 `System` 类的 `out` 方法可以实现标准输出流的重定向。 9、Java 中一个字符占用两个字节，所有字符采用的都是 UTF-8 码表。 10、`BufferedWriter` 的 `newLine()` 方法可以写入一个换行符。

第 8 章 IO（输入输出） 一、填空题 1、字节流、字符流 2、`File`、`java.io` 3、`SequenceInputStream` 4、`RandomAccessFile` 5、管道流 6、`LineNumberReader`、`BufferedReader`、`setLineNumber(int)`、`getLineNumber()` 7、字节流、字符流 8、`PrintStream`、`setOut(PrintStream out)` 9、`Unicode` 10、`newLine()`

1、简述流的概念。 2、Java 流被分为字节流、字符流两大流类，两者有什么区别？ 3、简要说明管道流。

1、Java 程序通过流来完成输入和输出，流是输入或输出信息的抽象。流通过 Java 的输入/输出系统与外设连接进行数据通信。流是抽象的对象，具体实现代码在 `java.io` 包中。 2、字节流的两个基类是 `InputStream` 和 `OutputStream`，字符流的两个基类是 `Reader` 和 `Writer`，它们都是 `Object` 类的直接子类，字节流是处理以 8 位字节为基本单位的字节流类；`Reader` 和 `Writer` 类是专门处理 16 位字节的字符流类。 3、管道流分为管道输入流（`PipedInputStream`）和管道输出流（`PipedOutputStream`），通常由一个 `PipedInputStream` 实例对象和一个 `PipedOutputStream` 实例对象相互连接而实现管道通信，`PipedOutputStream` 向管道写入数据，`PipedInputStream` 从管道中读取 `PipedOutputStream` 写入的数据，管道流主要用来完成线程之间的通信。

GUI1、

1、在 Java 中，图形用户界面简称_____，它的组件包含在_____和_____这两个包中。 2、为了避免实现监听器中定义的所有方法，造成代码的臃肿，在 JDK 中提供了一些_____类，这些类实现了接口所有的方法，但是方法中没有任何代码，属于一种空实现。 3、在 Java 中，_____类相当于一个抽象的画笔对象，使用它可以在组件上绘制图形。 4、_____负责监听事件源上发生的事件，并对各种事件做出响应处理。 5、大部分的 Swing 组件都是_____类的直接或者间接子类，其名称都是在原来 AWT 组件名称前加上字母 J。 6、AWT 事件有多种多样，大致可以分为_____、_____、_____、_____等。 7、如果点击 Frame 窗口右上角的关闭按钮能将其关闭，那么这个 Frame 窗口添加了_____监听器，并实现了监听器的实现了_____方法。 8、对话框可以分为_____和_____两种。 9、每一个容器都有一个默认的布局管理器，如果不希望通过布局管理器对容器进行布局，可以调用容器的_____方法将其取消。 10、创建下拉式菜单需要使用三个组件，分别是_____、_____和_____。

1、java.awt、javax.swing 2、适配器 3、Graphics 4、事件监听器 5、JComponent 6、窗体事件、键盘事件、鼠标事件、动作事件 7、WindowListener、WindowClosing(WindowEvent) 8、模态对话框、非模态对话框 9、setLayout(null) 10、JMenuBar、JMenu、JMenuItem

11、实现时间监听的步骤

通过实现 XxxListener 接口或者继承 XxxAdapter 类实现一个事件监听器类，并对处理监听动作的 方法进行重写 创建事件源对象和事件监听器对象 调用事件源的 addXxxListener()方法，为事件源注册事件监听器对象

2、awt 和 swing 的区别

AWT 是 Abstract Window Toolkit (抽象窗口工具包)的缩写，这个工具包提供了一套与本地图形界面进行交互的接口。AWT 中的图形函数与操作系统所提供的图形函数之间有着——对应的关系，当我们利用 AWT 来构件图形用户界面的时候，我们实际上是在利用操作系统所提供的图形库。由于不同操作系统的图形库所提供的功能是不一样的，在一个平台上存在的功能在另外一个平台上则可能不存在。为了实现 Java 语言所宣称的"一次编译，到处运行"的概念，AWT 不得不通过牺牲功能来实现其平台无关性，也就是说，AWT 所提供的图形功能是各种通用型操作系统所提供的图形功能的交集。由于 AWT 是依靠本地方法来实现其功能的，我们通常把 AWT 控件称为重量级控件。

Swing 是在 AWT 的基础上构建的一套新的图形界面系统，它提供了 AWT 所能够提供的功能，并且用纯粹的 Java 代码对 AWT 的功能进行了大幅度的扩充。由于在 Swing 中没有使用本地方法来实现图形功能，我们通常把 Swing 控件称为轻量级控件。

AWT 和 Swing 之间的基本区别：AWT 是基于本地方法的 C/C++程序，其运行速度比较快；Swing 是基于 AWT 的 Java 程序，其运行速度比较慢。对于一个嵌入式应用来说，目标平台的硬件资源往往非常有限，而应用程序的运行速度又是项目中至关重要的因素。在这种矛盾的情况下，简单而高效的 AWT 当然成了嵌入式 Java 的第一选择。而在普通的基于 PC 或者是工作站的标准 Java 应用中，硬件资源对应用程序所造成的限制往往不是项目中的关键因素，所以在标准版的 Java 中则提倡使用 Swing，也就是通过牺牲速度来实现应用程序的功能。

简述 java.awt 包中提供了哪些布局管理器。

在 java.awt 包中提供了五种布局管理器，分别是 FlowLayout(流式布局管理器)、BorderLayout(边界布局管理器)、GridLayout(网格布局管理器)、GridBagLayout(网格包布局管理器)和 CardLayout(卡片布局管理器)。

简述在事件处理机制中所涉及到的概念

事件处理机制专门用于响应用户的操作，比如，想要响应用户的点击鼠标、按下键盘等操作，就需要使用 AWT 的事件处理机制。在学习如何使用 AWT 事件处理机制之前，首先向大家介绍几个比较重要的概念，具体如下所示：

- 1、事件对象（Event）：封装了 GUI 组件上发生的特定事件(通常就是用户的一次操作)。
- 2、事件源（组件）：事件发生的场所，通常就是产生事件的组件。
- 3、监听器（Listener）：负责监听事件源上发生的事件，并对各种事件做出响应处理的对象(对象中包含事件处理器)。
- 4、事件处理器：监听器对象对接收的事件对象进行相应处理的方法。

TCP 协议的特点是 ，即在传输数据前先在 和 建立逻辑连接。

2、在计算机中，端口号是用 字节，也就是 16 位的二进制数表示，它的取值范围是

。 3、TCP/IP 协议被分为四个层，分别是

4、在 JDK 中，IP 地址用 类来表示的，该类提供了许多和 IP 地址相关的操作。。

5、使用 UDP 协议开发网络程序时，需要使用两个类，分别是 和 。

答案：

1、面向连接、客户端、服务器端 2、2、0-65535 3、链路层、网络层、运输层、应用层 4、InetAddress 5、DatagramPacket、DatagramSocket

1、网络通信协议是什么？ 2、TCP 协议和 UDP 协议有什么区别？ 3、Socket 类和 ServerSocket 类各有什么作用？

1、在 Internet 中传输数据都需要遵守一定的规则，这种规则通常被称作网络通信协议。网络通信协议对数据传输格式、传输速率、传输步骤等作了统一规定，通信双方必须共同遵守这个规定才能完成数据的交互。到目前为止，网络通信协议已经有很多种，其中 TCP/IP 协议在世界范围内应用最为广泛。 2、UDP 协议是无连接通信协议，所谓的无连接就是指数据的发送端和接收端不建立逻辑连接。由于 UDP 协议消耗资源小，通信效率高，通常都会用于音频、视频和普通数据的传输。UDP 协议在传输数据时不能保证数据的完整性，因此在传输重要数据时不建议使用 UDP 协议。 TCP 协议是面向连接的通信协议，即在传输数据前先在发送端和接收端建立逻辑连接，然后再传输数据，它提供了两台计算机之间可靠无差错的数据传输。在 TCP 连接中必须要明确客户端与服务端，由客户端向服务端发出连接请求，每次连接的创建都需要经过“三次握手”。

3、ServerSocket 类用于创建服务端程序，通过调用 ServerSocket 对象的 accept()方法，接收来自客户端的请求。 Socket 类用于创建客户端程序，当客户端和服务端的两个 Socket 建立了专线连接后，连接的一端既能向另一端连续写入字节，也能从另一端读取字节。Socket 类中定义了 getInputStream() 方法返回 Socket 的输入流对象，定义了 getOutputStream() 方法返回 Socket 的输出流对象。只要连接的一端向该输出流对象写入了数据，连接的另一端就能从其输入流对象中读取到。

方法返回 Socket 的输出流对象。只要连接的一端向该输出流对象写入了数据，连接的另一端就能从其输入流对象中读取到。