U1：

什么是Java的虚拟机？

Java虚拟机（JVM）其实就是一个字节码解释器，任何一种可以运行Java字节码的软件都可看成是Java虚拟机。可以把Java字节码看成是在Java虚拟机上运行的机器码，Java虚拟机就是以Java字节码为指令的"软CPU"，也就是说，Java虚拟机是可以运行Java字节码的假想计算机。它的作用类似于Windows操作系统，只不过在Windows上运行的是.exe文件，而在JVM上运行的是Java字节码.class文件。JVM是Java程序唯一识别的操作系统，对JVM来说可执行文件就是扩展名为.class的字节码文件.

什么是平台无关性？Java语言是怎样实现平台无关性的？

平台是指由操作系统和处理器所构成的运行环境。

与平台无关就是指应用程序的运行不会因为操作或处理器的不同而无法运行或出现错误，也可以说平台无关性是指一个应用程序能够运行于各种不同的操作系统上.

Java语言是通过虚拟机技术来实现平台无关性的，不同操作系统必须安装属于专属该平台的Java虚拟机。对Java程序而言，只认识一种"操作系统"，这个"操作系统"就是JVM，机器码（.class文件）就是JVM的可执行文件.

U2：

什么是JDK？

Java SE可以分为四个部分：JVM. JRE. JDK和Java语言.

JDK是Java Development Kits（Java开发工具包）的缩写，是一个编写Java Application应用程序和Applet小程序的程序开发环境（其中包括一些Java开发工具和Java的核心类库（Java API）等，是所有Java开发工具的基础.

JRE（Java Runntime Environment，JRE）与JDK的关系是：JRE是个运行环境，JDK是个开发环境。JRE不包含开发工具，如编译器、调试器和其他工具等. 而JDK包含了JRE以及开发过程中需要的一些工具程序，因此安装JDK后除了可以编辑Java程序外，还可以运行Java程序. 因此编写Java程序时需要JDK. 而运行Java程序时需要JRE.

环境变量Path和ClassPath的作用是什么？如何设置这两个环境变量？

Path环境变量的作用是设置供操作系统去寻找和执行应用程序(.exe、.com、.bat等)路径的顺序，对Java而言即Java的安装路径；ClassPath是JVM执行Java程序时搜索类的路径(类所在的文件夹)的顺序，以最先找到的为准，JVM除了在ClassPath的环境变量指定的文件夹中查找要运行的类之外，是不会在当前文件夹下查找相应类的.

设置Path和ClassPath两个环境变量的方法有两种：一种在控制面板中"系统和安全"下的"系统"页面内(或右击"我的电脑"图标，在弹出的快捷菜单中选择"属性"命令)设置；另一种是在命令行窗口中利用set命令进行设置.

Java应用程序与小程序之间有哪些差别？

小程序与应用程序之间的主要差别可用下表来说明

功能要求	应用程序(Application)	小程序(Applet)
使用图形	可选.	固定用图形
运行	主要从文件系统装入运行	通过HTML连接运行
内存要求	最低Java应用程序要求	Java程序加Web浏览器要求
环境输入	命令行参数	嵌入HTML文档的参数
JVM所要求的执行过程	主方法(main())启动过程	init()初始化过程 start()启动过程 stop()暂停/关闭过程 destroy()终止过程 paint()绘图过程

U3

1. 自动类转换的前提是什么？转换时从"矮"到"长"的优先级顺序是怎样的？

Java语言会在下列条件同时成立的前提下，自动进行数据类型的转换：

①转换前的数据类型与转换后的类型兼容.

②转换后数据类型的表示范围比转换前数据类型的表示范围大.

转换从"矮"到"长"的优先关系为：

低 byte → short → char → int → long → float → double → 高.

2. 编写程序，从键盘上输入一个浮点数，然后将该浮点数的整数部分输出.

3. 编写程序，从键盘上输入~~圆~~圆柱体的体半径r和高h，然后计算其体积并输出

2. InputStream、OutputStream、Reader 和 Writer 四个类在功能上有何异同？

U4：

1. 编写一个Java应用程序，在键盘上输入数n，计算并输出 1!+2!+…+n! 的结果。

2. 计算并输出一个整数各位数字之和。如，5423的各位数字之和为 5+4+2+3

U5：

1. 找出 4x5 矩阵中值最小和最大元素，并分别输出其值及所在的行号和列号。

2、编写Java应用程序，比较命令行中给出的两个字符串是否相等，并输出比较的结果。

3、杨辉三角和其他例子。

U6：

1、类与对象的区别是什么？

在面向对象的程序设计语言中，"类"就是把事物的数据与相关功能封装在一起，形成一种特殊的数据结构，用以表达真实事物的一种抽象。类是由数据成员和函数成员封装而成的，其中数据成员表示类的属性，函数成员表示类的行为。

对象则是该类事物的具体的个体，也称为实例。所以说类描述了对象的属性和行为。

2、成员变量与局部变量的区别有哪些？

类的成员变量与方法中的局部变量的区别主要有如下几个方面。

(1)从语法形式上看，成员变量是属于类的，而局部变量是在方法中定义的变量或是方法的参数；成员变量可以被 public、private、static 等修饰符所修饰，而局部变量则不能被访问控制符及 static 所修饰；成员变量和局部变量都可以被 final 所修饰。

(2)从变量在内存中的存储方式上看，成员变量是对象的一部分，而对象是存在于堆内存的，而局部变量是存在于栈内存的。

(3)从变量在内存中的生存时间上看，成员变量是对象的一部分，它随着对象的创建而存在，而局部变量随着方法的调用而产生，随着方法调用的结束而自动消失。

(4)成员变量如果没有被赋值初值，则会自动以类型的默认值赋值，而局部变量则不会自动赋值，也必须显式地赋值后才能使用。

3. 定义一个 Student 类，包含如下内容：

成员变量：学号、姓名、性别、班干部否、数学、语文、外语。

成员方法：输入、总分、平均分。

编程实现这个类，并调用相应的方法输入数据，计算总分和平均分。

U7：

1.对象的相等与指向它们的引用相等，两者有什么不同？

对象的相等一般指对象本身所包含的内容相等，

指向对象的引用相等一般则是指指向对象的首地址相同．所以两者有着本质的不同．

2、什么是静态初始化器？其作用是什么？静态初始化器由谁在何时执行？它与构造方法有何不同？

静态初始化器是由关键字static修饰的一对大括号"{}"括起来的语句组，其作用是对类自身进行初始化．静态初始化器在所属的类被加载入内存时由系统调用执行。

静态初始化器与构造方法的不同主要有如下几个方面。

(1)构造方法是对每个新创建的对象初始化，而静态初始化器是对类自身进行初始化．

(2)构造方法是在用new运算符创建新对象时由系统自动调用执行，而静态初始化器一般不能由程序来调用，它是在所属的类被加载入内存时由系统调用执行．

(3)用new运算符创建多少个对象，构造方法就被调用多少次，但静态初始化器只在类被加载入内存时只执行一次，与创建多少个对象无关．

(4)静态初始化器不同于构造方法，它不是方法，因而没有方法名、返回值和参数．

U8：

1.在调用子类的构造方法之前，会先自动调用父类中没有参数的构造方法，其目的是什么？

Java程序在执行子类的构造方法之前，若没有用super()调用父类中特定的构造方法，则会先自动调用父类中没有参数的构造方法，其目的是为了帮助继承自父类的成员做初始化的操作．

2、方法的"覆盖"与方法的"重载"有何不同？

重载是指在同一个类内定义名称相同，但参数个数或类型不同的多个方法，因此Java可根据参数的个数或类型的不同来调用相应的方法；

覆盖则是指在子类中，定义名称、参数个数与类型均与父类完全相同的方法，用以重写父类里同名方法的功能．

3. 什么是抽象类与抽象方法？使用时应注意哪些问题？

Java语言的抽象类是用abstract修饰符来修饰的类，抽象类是专门用来当作父类的。抽象类类似模板的作用，其目的是要用户根据它的格式来修改并创建新的类。

抽象类的方法可分为两种：一种是一般的方法，另一种是以关键字abstract开头的"抽象方法"。

抽象类中不一定包含抽象方法，但包含抽象方法的类一定要声明为抽象类。

抽象类本身不具备实际的功能，只能用于派生其子类，而定义为抽象的方法必须在子类派生时被覆盖。所以说一个类被定义为抽象类，则该类就不能用new运算符创建具体实例对象，必须通过覆盖的方式来实现抽象类中的方法。

一个类不能既是最终类，又是抽象类，即关键字abstract与final不能合用

另外，abstract不能与private、static、final或native并列修饰同一方法

4. 什么是接口？为什么要定义接口？

接口就是一个特殊的数据结构，它的结构与抽象类非常相似。

接口本身也具有数据成员与抽象方法，Java语言定义接口的目的主要是可以帮助实现类似于类的多重继承问题。

U9.

1. 若try语句结构中有多个catch们子句，这些子句的排列顺序与程序执行效果是否有关？为什么？

有关。由于异常对象与catch块的匹配是按照，catch块的先后排列顺序进行的。

所以在处理多异常时应注意认真设计各catch块的排列顺序。

一般地，将处理较具体、较常见异常的catch块应放在前面，而可以与多种异常类型相匹配的catch块应放在较后的位置。若将子类异常的catch块放在父类的后面，则编译不能通过。

U10.

1. 什么是流？Java语言中分为哪两种流？这两种流有何差异？

流是指计算机各部件之间的数据流动。

按照数据的传输方向，Java语言中的流分为输入流与输出流两种。从流的内容上划分，流分为字节流和字符流。字节流每次读写8位二进制数，由于它又能将数据以二进制的原始方式读写，因此字节流又被称为二进制字节流或位流。而字符流一次读写16位二进制数，并将其作为一个字符而不是二进制位来处理。

Binary Byte Stream

Bits Stream

2. InputStream. OutputStream. Reader 和 Writer 四个类在功能上有何异同?

InputStream 和 OutputStream 类是 Java 语言里用来处理以位(bit)为主的流.
它除了可用来处理二进制文件(图片.音频.视频等)的数据之外,也可用来处理文本文件,
Reader 和 Writer 类则是用来处理"字符流"的,也就是文本文件.

3. 利用文件输入输出流打开第7题创建的文件 file1.txt, 然后在文件的末尾
追加一行字符串"又添加了一行文字!".

UI1:

1. Java 程序实现多线程有哪两个途径?

Java中实现多线程的方法有两种: 一种是继承 java.lang 包中的 Thread 类,
另一种是用户在自己定义的类中实现 Runnable 接口.
无论用哪种方式实现多线程,都需要将执行的代码编写在 run() 方法中,然后启动
线程从 run() 方法开始执行。

2. 在什么情况下必须以类实现 Runnable 接口来创建线程?

如果一个类本身已经继承了某个父类,由于 Java 语言不允许类的多重继承,所以就无法再
继承 Thread 类,特别是小程序,这种情况下若要实现多线程的功能可以创建一个类,
该类必须来实现 Runnable 接口.

UI3.

大题. 设计界面.

UI4.

大题. 可能是 口+口=口.

1. 什么是事件？Java语言的委托事件模型？

事件就是用户使用鼠标或键盘与窗口中的组件进行交互时所发生的事情。
委托事件模型是将事件源和对事件做出具体处理的程序分离开来。
一般情况下，组件（事件源）不处理自己的事件，而是将事件处理委托给外部的处理实体（监听者），这种事件处理模型就是事件委托处理模型，即事件源将事件处理任务委托给了监听者。
换个角度来说，委托事件模型是由产生事件的对象（事件源）、事件对象、以及事件监听者对象之间的关系所组成。其中的事件监听者就是用来处理事件的对象，也就是说，监听者会等待事件的发生，并在事件发生时收到通知。事件源会在事件产生时，将关于该件事件的信息封装在一个对象中，这就是事件对象，并将该事件对象作为参数传递给事件监听者，监听者就可以根据该事件对象内的信息，决定运当的处理方式，即调用相应的事件处理程序进行处理。

2. 若要处理事件，就必须要有事件监听者，通常哪些对象可以担任监听者？

一是让包含事件源的对象来担任监听者
二是定义内部类来担任监听者
三是使用匿名类来来担任监听者