**Java程序设计复习大纲**

**题型：**

**选择题（共20分，每题2分）**

**填空题（共20分，每题2分）**

**判断题（共10分，每题2分）**

**简答题 （共10分，每题5题）**

**程序阅读题（共20分，每题5题）**

**编码题（共20分，每题10分）**

选择+判断题+填空：重载+重写；main方法，Java命名; 构造方法；类方法；虚拟机；抽象类和接口；import;package;子类和父类的各种关系；对象和类和内存；set,list, map;FileInputStream和FileOutputStream等类和异常处理；通信管道，包括它的类；类包括内部类的权限；异常处理，运行时异常，非运行时异常；Java的绑定方式；static,this, final,abstract等；面向对象3大特性；线程基本概念；多线程共享的相关概念；什么是数据流；java有什么数据类型，数据类型变量占多少位;什么是引用数据类型；匿名内部类；不同类型的I/O流；串行化机制；网络通信核心等；静态和非静态；java编译；线程，例如线程的5个运行状态；过滤流；外部类和内部类；常量定义；线程同步方法；

简答题：编译器的3个步骤等；类方法使用的3条限制；串行化和主要任务；4种访问权限的区别；

**程序阅读题：数组处理（例如九九乘法表，例如螺旋矩阵等，例如杨辉三角）；初始化问题；内部类；父类和子类；字符串输出等。可组合，可分开**

编码题：例如递归方法，例如斐波那契数列计算等；网络客户端，服务器端编程；**文件处理，包括文件建立，更新，复制等；**。

下面是参考，并不完整哈。

**其他题型：**

1. 面向对象三大特征；

封装、继承、多态

1. Java在JDK中的编译命令及其编译对象，运行命令及其运行对象；

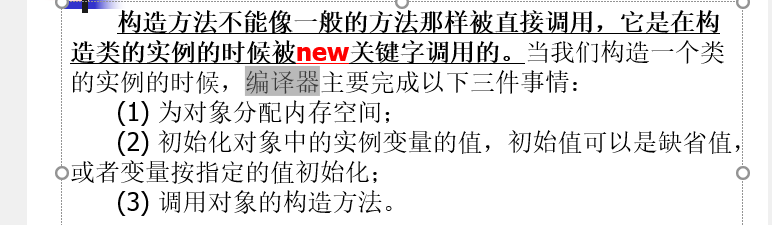
编译命令：javac xx.java

编译对象：java文件

运行命令：java xx

运行对象：class文件

1. 构建类的实例时，java编译器的三部曲



1. Java的源代码格式；

第一行是package语句，接着是import语句，然后定义类

1. 构造器

构造器的名字必须与类名相同

构造器没有返回类型

构造器可以重载

构造器调用构造器 this问题 一个类中，一个构造器调用另一个构造器，用关键词 this.

1. Java的赋值，

对象间的赋值；

方法的形式参数（或叫局部变量）为对象时，在方法内对对象赋值；

1. 基本的操作符，+,-,\*,/,=, ==, !=等（注意String类的+, +=）；
2. 二元算术运算符：运算结果的数据类型一般为两个操作数中表达范围较大的类型。e.g.整数 op 浮点数 → 浮点数；
3. java的==：基本数据类型比较值，对象类型比较地址。注意Integer的-128~127、String的字符串常量池问题；
4. String类的+：当操作数中有一个是字符串时，程序运行时会对另一个操作数进行字符串转换。注意+的左结合性（运算先后顺序）。e.g.1+2+"=3"→"3=3"；"12="+1+2→"12=12"。
5. Import, package 语法

import：在package语句（若有）之后，分单类型导入（仅一个public类或接口，e.g.import java.io.File）和按需类型导入（导入某包下所有当前需使用的类，不包括子目录，可能会因同名类造成命名冲突，e.g.import java.io.\*)。java.lang下所有public类自动导入；导入静态成员

e.g.import static java.lang.Math.max;

（2）package：包声明，在源文件非注释的第一行，最多一个（如不使用，将被放在无名包中）。

1. 覆盖+重载

覆盖：

子类对父类方法的重写

子类方法和父类方法同名同参同返回类型

重载：

同一个类中建立多个同名方法

重载的方法同名不同参

1. 初始化（重点）

注意，static定义的变量或方法，可以不用声明类对象，就可以直接调用，但是要注意初始化顺序；

1. Java的访问权限控制，关键词，权限大小，适用范围

类的访问权限控制和import的联系

一定要注意注意注意，从访问权限上来说

不是public > protected > default > private

因为protected 和 default是继承才有区别。注意注意。

1. 抽象类、接口的定义，特点，注意事项等
2. static、final的概念，适用范围，注意事项等

（1） static 静态修饰关键字，可以修饰静态变量、类方法和静态内部类

注意：静态方法不能被重写，静态方法只能调用静态的东西

静态块在类被加载时执行且只执行一次

（2）final 可以修饰常量、类和方法

注意：被final修饰的类不能被继承，被修饰的方法不能重写

final修饰的常量需要在声明时初始化或构造函数中初始化

1. 各种容器的基本概念，特点；list, set，map，队列

（List 接口存储一组不唯一，有序（插入顺序）的对象。 Set 接口存储一组唯一，无序的对象。 Map接口存储一组键值对象，提供key到value的映射。key无序，唯一。value不要求有序，允许重复。）

1. 运行异常，try-catch-finally运行原理。

（try { //执行的代码，其中可能有异常。一旦发现异常，则立即跳到catch执行。否则不会执行catch里面的内容 } catch { //除非try里面执行代码发生了异常，否则这里的代码不会执行 } finally { //不管什么情况都会执行，包括try catch 里面用了return ,可以理解为只要执行了try或者catch，就一定会执行 finally }

在try 和catch中如果要return，会先去执行finally中的内容再返回。）

1. 管道【查一下】

Java提供管道功能，实现管道通信的类有两组：PipedInputStream和PipedOutputStream或者是PipedReader和PipedWriter。管道通信主要用于不同线程间的通信。

一个PipedInputStream实例对象必须和一个PipedOutputStream实例对象进行连接而产生一个通信管道。PipedOutputStream向管道中写入数据，PipedIntputStream读取PipedOutputStream向管道中写入的数据。一个线程的PipedInputStream对象能够从另外一个线程的PipedOutputStream对象中读取数据

1. 线程，进程（线程运行的5个状态）

一个进程就是一个执行中的程序，而每一个进程都有自己独立的一块内存空间、一组系统资源。在进程概念中，每一个进程的内部数据和状态都是完全独立的。Java程序通过流控制来执行程序流，程序中单个顺序的流控制称为线程，多线程则指的是在单个程序中可以同时运行多个不同的线程，执行不同的任务。

1. 局部内部类的声明对象，调用（重点）

局部内部类不能被访问修饰符和static修饰，且只能访问final变量和形参。

1. 通过new一个对象来调用

Inner in = new Inner（）；

1. 使用匿名对象来调用（这个调用的只能用一次而且创建多了浪费资源）

new Inner（）.inner（）；

1. 匿名内部类和嵌套类的概念和注意事项

匿名内部类：

只创建这个类的一个对象，不用为它命名。在定义类的同时，就生成该类的一个实例，并且不会在其他地方听到这个类。

用于构造对象的任何参数都要被放在超类名后面的括号内。

匿名内部类不能有构造器。

匿名内部类既可以拓展类，也可以拓展接口。同时只能且必须实现一个类或者是一个接口。

匿名内部类是局部内部类的一种。

嵌套类：

在一个类中定义另外一个类。

嵌套类的范围受其封闭类的范围限制。

嵌套类可以访问封闭类的成员，包括私有成员。

嵌套类可以被声明为private public protected或package private。

内部类是非静态嵌套类。

1. Inputstream, outputstream, fileinputstream, fileoutputst

InputStream和OutputStream都是抽象类，不能实例化，因此在实际应用中都使用的是他们的子类。Java通过系统类System实现标准输入输出的功能，定义了3个流变量，in，out和err。System.in作为字节输入流类InputStream的对象实现标准输入。System.out作为打印流类PrintStream的对象实现标准输出。

FileInputStream和FileOutputStream用于进行文件的输入输出处理，其数据源和接收器都是文件。 FileInputStream用于顺序访问本地文件，FileInputStream重写了抽象类InputStream的读取数据的方法。FileOutputStream用于向一个文本文件写数据，FileOutputStream重写了抽象类OutputStream的写数据的方法。

1. stream概念

Stream是java的1个类, 这个类专门用于程序和外部设备的输入输出(IO). 基本上所有流都在 java.io这个包中.

实际上Stream就是数据在程序和外部设备的单向管道, 流的各种方法相当于管道上的各种按钮.

1. 互斥对象的概念和使用

1 为某个对象设置一个“互斥锁”标记。该标记保证在某一个时刻，只能有一个线程拥有该互斥锁，其它线程如果需要获得该互斥锁，必须等待当前拥有该锁的线程将其释放。该对象称为互斥对象。

2 为了配合使用对象的互斥锁，Java语言提供了保留字synchronized。其基本用法如下：

synchronized(互斥对象){

    临界代码段

}

1. 线程的几种状态（5种）

创建态，就绪态，运行态，阻塞态，死亡状态

1. Java数据类型的几类

基本数据类型：整数类型、浮点类型、字符类型、布尔类型

引用数据类型：类、接口、数组

1. 临界资源或同步资源概念

临界资源：

在并发程序设计中，对多线程共享的资源或数据称为临界资源，而把每个线（进）程中访问临界资源的那一段代码段成为临界代码段。通过为临界代码段设置信号灯，就可以保证资源的完整性，从而安全地访问共享资源 。ppt10.27

许多物理设备都属于临界资源，如打印机等。此外，还有许多变量、数据等都可以被若干进程共享，也属于临界资源。

1. 网络通信核心

协议

1. java几种类型变量所占位数

byte 8 short 16 int 32 long 64 float 32 double 64 char 16

1. 向上转型mklm

1）上转型对象不能操作子类新增的成员变量和方法。

  2）上转型对象可以操作子类继承或重写的成员变量和方法。

3）如果子类重写了父类的某个方法后,当对象的上转型对象调用这个方法时一定是调用了这个重写的方法

1. 内部类

可以将一个类的定义放在里另一个类的内部，这就是内部类。广义上我们将内部类分为四种：成员内部类、静态内部类、局部（方法）内部类、匿名内部类。