# Java重点知识整理

作者：娄方成

**1.1简述JAVA的语言特点**

a语法简单，功能强大

b分布式与安全性

c与平台无关

d多线程

**1.2什么是Java虚拟机？它包括哪几个部分？**

解： Java 虚拟机 (JVM ) 规范中给出了它的定义： JVM是在一台真正的机器上用软件方式实现的一台假想机。Java 虚拟机是运行 Java 程序必不可少的机制，它是编译后的Java 程序和硬件 系统之间的接口．程序员可以把 JVM 看作一个虚拟的处理 器。编译 后的 Java 程序指令并不直接在硬件系统的CPU上执行，而是由 JVM 执行。它不仅解释执行编译后的Java 指令，而且还进行安全检查。它是 Java 程序能在多平台间进行无缝移植的可靠保证 ，同时也是 Java程序的安全检验引擎。

JVM的具体实现包括指令集（等价于 CPU的指令集）、寄存器组、类文件格式、栈、垃圾收集堆 、内存区。

**1.12列出java.lang.math类的常用方法**

绝对值，三角函数，开根，对数，随机数，幂，最值

**2.3请叙述标识符的定义规则**

**( 1 ）   标识符是以字母、下画线( \_)或美元符( $ )开头 ，由字母、数字、下画线( \_ )或美 元**

**符号（＄）组成的字符串。**

**( 2 )标识符区分大小写。**

**( 3 )标识符的长度没有限制。**

**( 4 )注释不能插在一个标识符或关键字之中。**

**( 5 )Java有许多关键字，它们都 有各自的特殊意义和用法，不得用它们作为标识符标 识符内 可以包 含关键字 ，但不 能 与 关 键 字 完 全 一 样 。 如 t hi s O ne是一个合法的标识符． 但 t h is 是 关 键 字，不能当作标识符。**

**2.4Java中有哪些数据类型？他们分别用什么符号表示？**

a布尔类型   Boolean

b字符类型   char

C整型       byte、short、int、long

d浮点型     float、double

**2.5什么是对象？基本数据类型和对象有何不同？**

解：对象有两个层次的含义：现实生活中对象指的是客观世界的实体；在程序中对象就是一组变屈和相关方法的集合，其中变员表明对象的状态，方法表明对象所具有的行为。

对象是类的一个 实例。类可以看作 是一个模板—— 正在描述的对象的 模型。每当 创建一个类的实例时．得到的就是一个对象。

不同：

基本数据类型是如数或是字符这样的基本值。对象是更复杂的项，通常包含了定义对象的基本数据类型。它们所对应的内存分配的时机、使用的方式都是不同的。

**2.6Java中的类型转换是什么？如何进行安全的类型转换？**

将不同类型的数据转换为同一类型的数据就被称为类型转换。

转换时位数少的类型转换为位数多的类型这样就能进行安全的类型转换。

**2.11设计并实现一个Course类，它代表学校的一门课程。按照实际情况，将这门课程的相关信息组织成它的属性，并定义必要的相应方法。**

核心代码：

system.out.println(“the basic info of this course as fllowed;”)

system.out.println(“courseID:”+this.getID());

system.out.println(“courseName:”+this.getName());

system.out.println(“courseType:”+this.getType());

system.out.println(“classHour:”+this.getClassHou());

system.out.println(“credit:”+this.getCredit());

system.out.println(“-----------------------------------------------);

**2.13设计并实现一个Vehicle类及其子类，它们代表主要交通工具，定义必要的属性信息及访问方法。**

核心代码：

System.out .println ("\n--- - - - - - - -- - - - -- --- --- - - -- - ");

System.out.println("the vehicle info as followed:") ; System.out.print("brand="+this.brand+"\t");

System.out.print("color="+this.color+"\t") ; System.out.print ("price="+ this.price+" \t " ) ;

System. out .print.("number="+ this. number+ " \ t " ) ;

**3.4设n为自然数，n！称为n的阶乘，规定0！=1.**

**n！=1X2X···Xn**

**试编制程序计算2！，4！，6！，8！和10！**

public static int Factorial (int n)

if (n==O) return l ;

returnn\* Factorial (n-1) ;

public  void print()

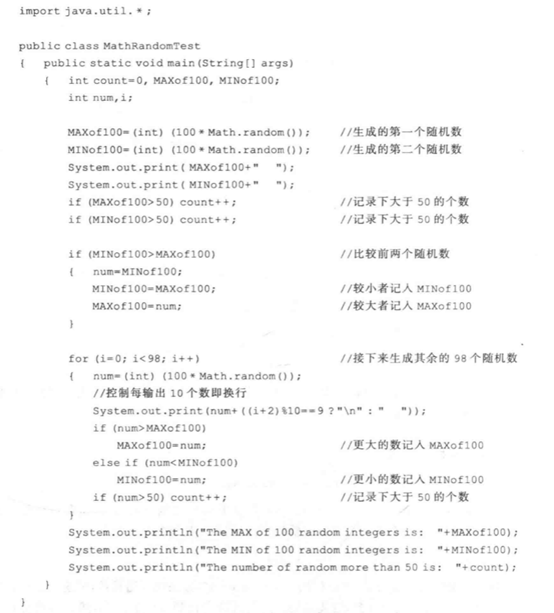
System. out .println (initVal+" != " +r e s u l t ) ;

public void setinitVal (intn)

initVal=n;

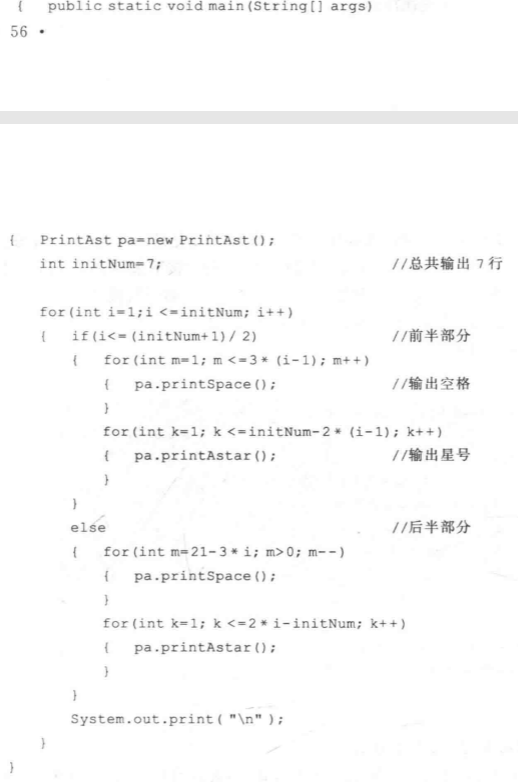
private  int result, initVal;

3.5使用Java.lang.Math类，生成100个0倒99的随机整数，找出它们的最大值和最小值，并统计大于50的整数个数。

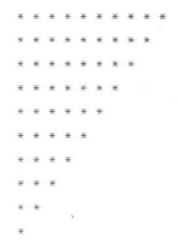
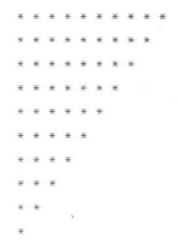
[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619083230218-903724202.png)

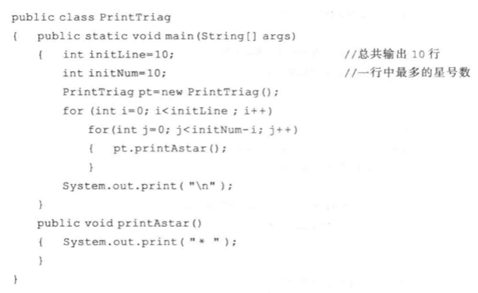
**3.7编写程序打印下面图案。**

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619083843592-898524995.png)

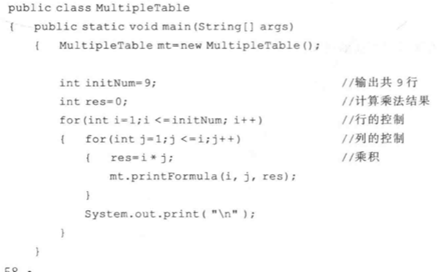
[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619083911515-2089460987.png)

**3.8编写程序打印下面图案.**

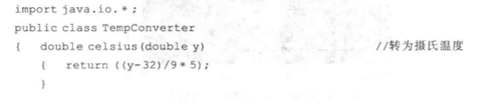
[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619083946770-726795460.png)[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084329099-284059566.png)

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084411631-1892961341.png)

**3.9打印九九乘法表**

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084426707-1593146399.png)

**3.12温度单位转换**

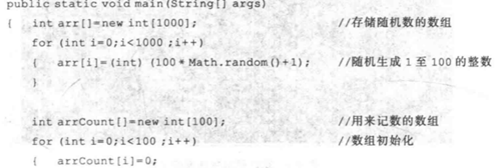
[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084440123-1005739186.png)

**4.2请简述数组创建的过程。如何创建一个对象数组？**

定义加初始化的过程就是数组创建的过程。

先在栈中开辟空间，初始化普通数组的引用；在堆中开辟空间初始化对象的引用及开辟空间初始化对象数据对象的属性和成员方法，最后引用指向对象的数据。

**4.9定义一个一维数组，其中存储随机生成的1000个1~100的整数，统计每个整数出现的次数。**

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084455488-334863628.png)

**5.4什么叫做方法重载？什么叫做方法重写？它们之间的区别是什么？**

方法重载：同一个类中，可以定义同名的多个方法。它们的不同之处在于参数列表不同 ，这其中包括参数的个数不同或是对应的参数类型不完全相同。这就是方法的重载。

方法重写：如果父类中原有的方法不能满足子类的要求，可以在子类中对父类的方法重新编写代码。这称为方法重写。

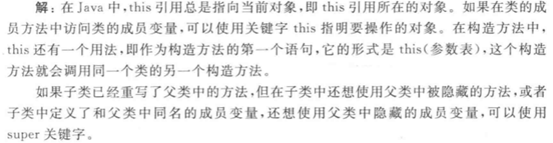
1从对方法重载和方法重写的分析可以看出，它们之间的区别主要有：

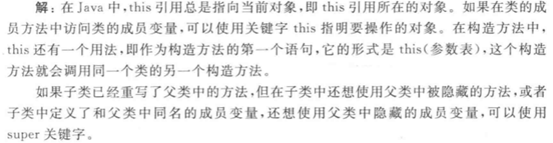
2方法重载时参 数列表必须是不同的，使系统能区别出到底调用哪个方法，而方法重写时参数列表可以是相同的。

方法重载时 ，方法的 返回类 型可能不同 ，也可能相 同；而方法重写时，子类中方法的返3回类型 和父类中同名方法的返 回类型完全一样。

方法重载多出现在同一个类中，方法重写必须是在父子类中。

**5.6关键字this和关键字super在成员方法中的特殊作用是什么？**

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084529834-379551122.png)

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084539980-1643171088.png)

**5.8什么叫做静态方法和静态变量，它们同普通的成员方法和成员变量有什么区别？**

静态变量：是基本数据类型，在类的外部不必创建该类的实例可以直接使用。（偷懒简写）

静态方法：方法在内存中只有一份，无论该类有多少个实例，都共用一个方法。

不同：静态变量和静态方法都可以直接使用而普通成员方法变量不行。

**5.8什么是抽象类？什么是抽象方法？他们有什么特点和用处？**

抽象类：定义了方法但没有定义具体实现的类

抽象方法：这种方法只有方法的声明，而没有方法的实现。这些方法在抽象类的子类中实现

**5.11什么是接口？接口的作用是什么？它与抽象类有何区别？**

接口是抽象类功能的另一种实现方法。可将其想象为一个“纯”的抽象类。

作用：接口的作用是为了保证多重继承，它可以定义多个类的共同屈性。而且，J av a通过允许一个类实现多个接口从 而 实现了比多 重继承更加强大的能力，并具有更加清晰的结构。

**6.1什么是异常？解释“抛出”、“捕获”的含义。**

异常： J av a 把程序运行中可能遇到的错误分为两类 ，一类是非致命性的，通过某种修正后程序还能继续执行。这类错误称作异常。

抛出：针对 J a va 程序中引发的可识别错误，产生与该错误相对应的异常类对象的这一过程称为抛出。

捕获：专门处理异常的过程称为捕获 。

**6.2Java是如何处理异常的？**

当发生异常时，程序中断执行，并输出一条信息。J ava 中，使用 t r y 语句括住可能 抛出异常的代码段，用ca t ch语句指明要捕获的异常及相应的处理代码。

**6.3catch及try语句作用是什么？语法格式如何？**

 对于可能抛出异常的代码段，要使用 t r y 语句括住，用ca t ch 语句指明要捕获的异常及相应的处理代码。

**6.5Java中常见的异常**

算数异常、空指针异常、数组元素个数为负异常、数组下标越界异常、安全异常

**7.1什么是泛型？为什么要使用泛型？**

在设计类和接口时 ，                 说明相关的数据类型。J a va 语言允许在类或接口的定义中，用－ 个 占 位符 替 代 实际 的类 类 型。 这 个 技 术 称 为 泛 型 。

使用泛型的目的：：

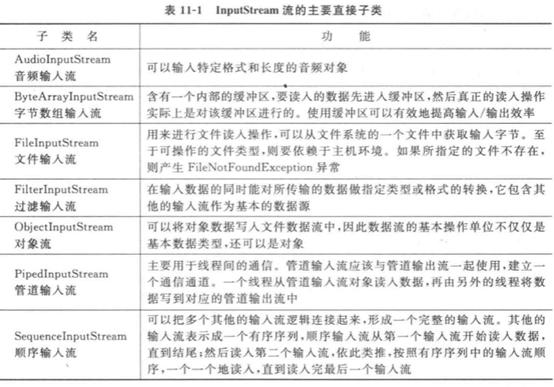
a类型安全              b消除强制类型转换   c潜在的性能收益

**11.2什么叫做流？输入输出流分别对应哪两个抽象类？**

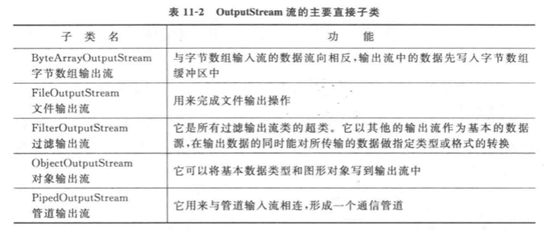
流也称数据 流，是指 在计算 机的输 入输出之间流动的数据的序列。这是一组有顺序的有起点和终点的字节集合。

输入流用抽象类 lnputStream 来实现，输出流用抽象类OutputStream 来实现。

**11.3 InputStream有哪些直接子类？其功能是什么？**

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084631624-1531423225.png)

11.4 Out**putStream有哪些直接子类？其功能是什么？**

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084650530-80543788.png)

**12.1什么叫做线程？什么叫做多线程？Java的多线程有何特点？**

线程：是进程执行过程中产生的多条 执行线索，是比进程更小的执行单位，它在形式上同进程十分相似一—都是用一个顺序执行的语句序列来完成特定的功能。

多线程：目前大多数操作系统支持多任务 ，每一个任务就是一个进程。一个进程在其执行过程中，可以产生多个线程，即多线程。

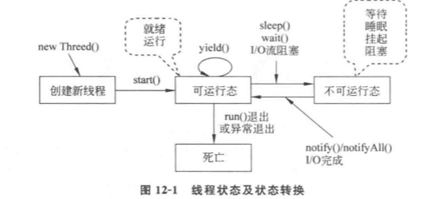
Java多线程的特点：

**12.2什么叫做线程的生命周期？线程的一个生命周期包括哪些状态？各状态之间是如何进行转换的？**

从线程的创建直到线程退出运行称为线程的生命周期。

线程的一个生命周期包括以下一些状态：新建、就绪、运行、阻塞和死亡。

状态转换图：

[](https://img2018.cnblogs.com/blog/1481303/201906/1481303-20190619084714190-1628622712.png)

**12.3有几种创建线程的方法？分别是什么？**

创建线程有两种方法，第一种是继承T h r ea d 类，定义它的 一个子类 ，这个子类就可以表示线程。

创建线程的另一种方法是实现 R u n n a b le 接口。

**12.8为什么多线程系统中要引入同步机制？Java是如何实现同步的？**

多线程访问共享数据时，可能会出现访问了不该访问的数据或是漏过需要访问的数据这种情况，这是因为缺乏线程和共享数据之间的协调管理机制而导致的。

为了解决类似的问题 ，Java 使用对象锁的概念，为可能同时访问共享数据的代码实例定义了一个“锁定标志”，使用 synchro nize d关键字进行同步，它提供了操作锁定标志的方法。

**12.9什么叫死锁？Java同步机制中是否可以阻止死锁的发生？**

 在多线程竞争使用资源的程序中，可能会发生这样的情况： 一个线程等待另一个线程所待有的锁，而第二个线程又在等待第一个线程所持有的锁，也就是它们互相等待对方所持有的锁。一般来讲，每个线程都占有部分资源，而等待其 他线程占有的另外部分资源。因为每个线程都得不到运行所需的全部资源，所以每个线程都不能继续运行。**这就出现了死锁。死锁是指多个线程相互等待，使得程序无法继续运行 。**

Java既不监测也没有采取办法防止死锁的出现。