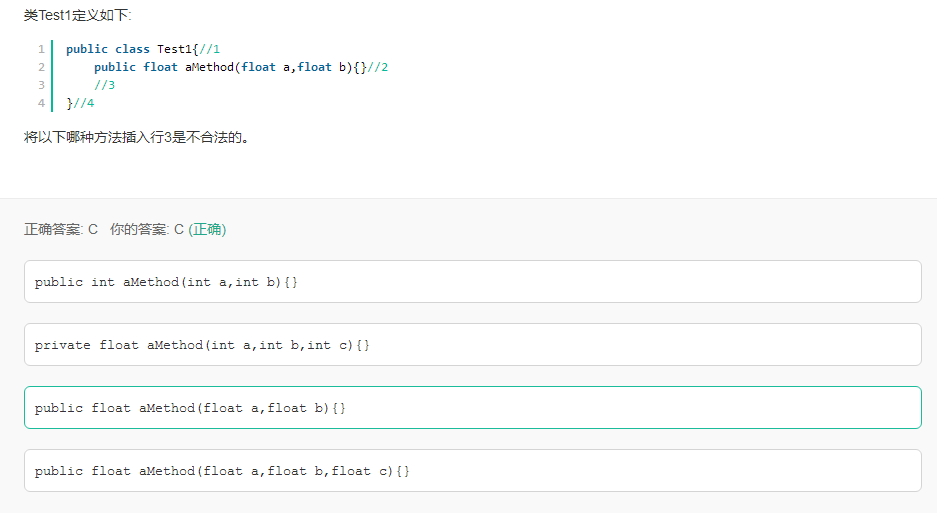
**1.知识点：方法的重载**

****

**解析：**

**关于方法的重载：**方法重载是指在一个类中定义多个**同名的方法，但要求每个方法具有不同的参数的类型或参数的个数。**调用重载方法时，Java编译器能通过检查调用的方法的参数类型和个数选择一个恰当的方法。方法重载通常用于创建完成一组任务相似但参数的类型或参数的个数不同的方法。

**方法重载具体规范**

一.方法名一定要相同。

二.方法的参数表必须不同，包括参数的类型或个数，以此区分不同的方法体。

三.方法的返回类型、修饰符可以相同，也可不同。

**知识扩展：方法的重写**

方法的重写是指：子类包含与父类同名方法的现象被称为方法重写，也称方法覆盖。

方法的重写遵循“两同两小一大”规则。

“两同”：方法名同，形参列表同。

“两小”：子类方法返回值类型比父类方法返回值类型小或者相等。

“一大”：子类方法的访问权限比父类更大或者相等。

**2.知识点：封装**



**解析：**

（1）什么是封装？

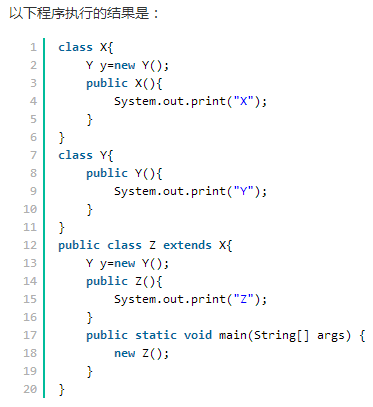
封装就是将属性私有化，提供公有的方法访问私有属性。

（2）为什么需要封装？

通过封装，可以实现对属性的数据访问限制，同时增加了程序的可维护性。

由于取值方法和赋值方法隐藏了实现的变更，因此并不会影响读取或修改该属性的类，避免了大规模的修改，程序的可维护性增强。

**3.知识点：初始化问题**



**解析：**

初始化过程：

1.初始化父类中的静态成员变量和静态代码块；

2.初始化子类中的静态成员变量和静态代码块；

3.初始化父类的普通成员变量和代码块，再执行父类的构造方法；

4.初始化子类的普通成员变量和代码块，再执行子类的构造方法；

**（**1）初始化父类的普通成员变量和代码块，执行Y y=new Y();输出Y

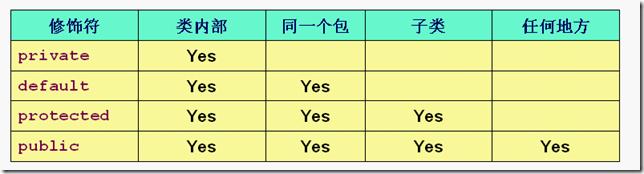
（2）再执行父类的构造方法；输出X

（3）初始化子类的普通成员变量和代码块，执行Y y=new Y();输出Y

（4）再执行子类的构造方法；输出Z

所以输出YXYZ。

**4.知识点：访问级别**

****

**5.知识点：访问级别**

****

（1）/sbin/init在核心完整的加载后，开始运行系统的第一支程序，主要的功能就是准备软件运行的环境，包括系统的主机名称、网络配置、语系处理、文件系统格式及其他服务的启动等。

（2）/bin/sh解释脚本的shell命令，开机后运行

（3）/etc/sysvinit就是 system V 风格的 init 系统，顾名思义，它源于 System V 系列 UNIX。sysvinit 中运行模式描述了系统各种预订的运行模式。

（4）/etc/inittab定义了系统引导时的运行级别, 进入或者切换到一个运行级别时做什么。

**6.知识点：线程调度**

线程调度分为协同式调度和抢占式调度，Java使用的是抢占式调度，也就是每个线程将由操作系统来分配执行时间，线程的切换不由线程本身来决定（协同式调度）。这就是平台独立的原因。

**7.知识点：Linux命令**

在linux命令中reboot是重新启动，**shutdown -r now是立即停止然后重新启动，都说他们两个是一样的，其实是有一定的区别的**。

**shutdown**命令可以**安全地关闭**或重启Linux系统，它在系统关闭之前给系统上的所有登录用户提示一条警告信息。该命令还允许用户指定一个时间参数，可以是一个精确的时间，也可以是从现在开始的一个时间段。

精确时间的格式是hh:mm，表示小时和分钟，时间段由+ 和分钟数表示。系统执行该命令后会自动进行数据同步的工作。

该命令的一般格式: shutdown [选项] [时间] [警告信息]

命令中各选项的含义为:

- k 并不真正关机而只是发出警告信息给所有用户

- r 关机后立即重新启动

- h 关机后不重新启动

- f 快速关机重启动时跳过fsck

- n 快速关机不经过init 程序

- c 取消一个已经运行的shutdown

需要特别说明的是该命令只能由超级用户使用。

例1，系统在十分钟后关机并且马上重新启动: # shutdown –r +10

例2，系统马上关机并且不重新启动:# shutdown –h now

**halt**是最简单的关机命令，其**实际上是调用shutdown -h命令**。halt执行时，杀死应用进程，文件系统写操作完成后就会停止内核。

halt命令的部分参数如下：

[-f] 没有调用shutdown而强制关机或重启

[-i] 关机或重新启动之前，关掉所有的网络接口

[-p] 关机时调用poweroff，此选项为缺省选项

**reboot**的工作过程与halt类似，其作用是重新启动，而halt是关机。其参数也与halt类似。reboot命令重启动系统时是删除所有的进程，而不是平稳地终止它们。因此，**使用reboot命令可以快速地关闭系统，但如果还有其它用户在该系统上工作时，就会引起数据的丢失。所以使用reboot命令的场合主要是在单用户模式**。

**init**是所有进程的祖先，其进程号始终为1。init用于切换系统的运行级别，切换的工作是立即完成的。**init 0命令用于立即将系统运行级别切换为0，即关机；init 6命令用于将系统运行级别切换为6，即重新启动**

**8.知识点：单例模式**

**9.知识点：接口的作用:** **契约式设计以封装实现**

**契约式设计**或者**Design by Contract**  ( **DbC** )是一种设计[计算机软件](http://baike.baidu.com/view/8318.htm)的方法。这种方法要求 [软件设计](http://baike.baidu.com/subview/575391/575391.htm) 者为软件组件定义正式的，精确的并且可验证的接口，这样，为传统的[抽象数据类型](http://baike.baidu.com/view/178593.htm)又增加了先验条件、后验条件和不变式。

**10.知识点：抽象类**



包含抽象方法的类称为**抽象类**，但并不意味着抽象类中只能有抽象方法，它和普通类一样，同样可以拥有成员变量和普通的成员方法。注意，**抽象类和普通类的主要有三点区别**：

1）抽象方法必须为public或者protected（因为如果为private，则不能被子类继承，子类便无法实现该方法），缺省情况下默认为public。

2）抽象类不能用来创建对象；

3）如果一个类继承于一个抽象类，则子类必须实现父类的抽象方法。如果子类没有实现父类的抽象方法，则必须将子类也定义为abstract类。

在其他方面，抽象类和普通的类并没有区别。