用构建器自动初始化

构建器的名字与类名相同。这样可保证此方法会在初始化期间自动调用。

Java在类中会有一个默认的构建器，当手动创建了一个构建器后默认构建器会失效。通过构建器这个特殊方法来担保每个对象都会得到正确的初始化。

清除：收尾和垃圾收集

Java可用垃圾收集器回收由不再使用的对象占据的内存。

垃圾收集器只知道释放那些由new分配的内存，所以想要释放特殊的对象内存，Java提供了finalize()方法。

在理想情况下，它的工作原理应该是这样的：一旦垃圾收集器准备好释放对象占用的存储空间，它首先调用finalize()，而且只有在下一次垃圾收集过程中，才会真正回收对象的内存。所以如果使用finalize()，就可以在垃圾收集期间进行一些重要的清除或清扫工作。但垃圾收集并不等于破坏，我们的对象可能不会当作垃圾被收掉。

垃圾收集只跟内存有关。也就是说，垃圾收集器存在的唯一原因是为了回收程序不再使用的内存。垃圾收集器会负责释放所有对象占据的内存，无论这些对象是如何创建的。

对象的创建过程

1. 对象首次创建，Java解析器会去找到.class文件，在事先设好的类路径里搜索。
2. 找到.class文件后，对Class对象首次载入时，它所有的static初始化模块都会运行，并只发生一次。
3. 当new对象时，对象的构建进程首先会在内存堆里为对象分配足够多的存储空间。
4. 之后存储空间会清为零，将对象中的所有基本类型设为它们的默认值。
5. 进行字段定义时发生的所有初始化都会执行。
6. 执行构建器。

总结

构建器使我们能保证正确的初始化和清除，没有正确的构建器调用，编译器不允许对象创建，所以能获得完全的控制权和安全性。

用new创建的对象也必须明确的清除，而垃圾收集器会自动为所有对象释放内存。