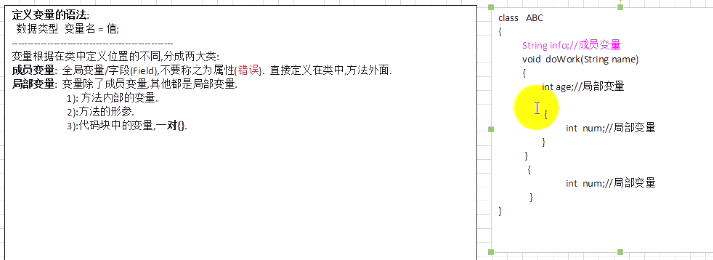
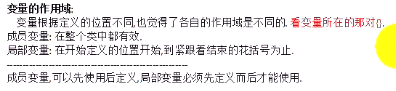
Day 07

**01 变量的分类-作用域-初始值**

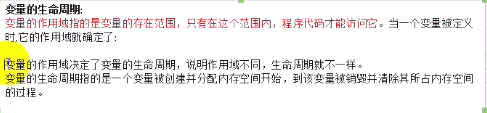


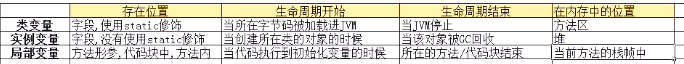


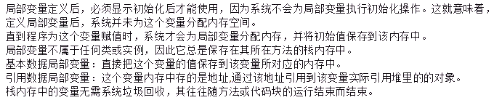


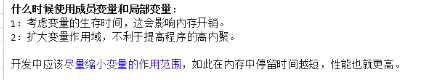
因为调用成员变量是时经定义了对象，而对象存在则成员变量已经存在。

**02变量的生命周期和选用**









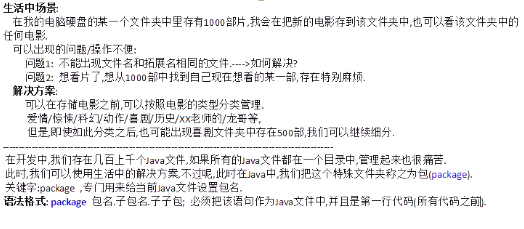
变量在内存中停留的时间越短则性能越高。

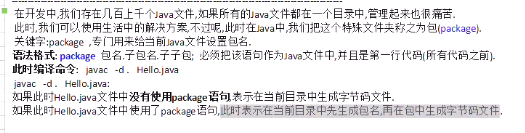
不要随意使用static修饰，非常消耗内存

也不要使用成员变量，因为存在线程的不安全问题（单例模式下）。

能使用局部变量就使用局部变量，因为局部变量是使用时单独开辟空间，用完即销毁。

**03package关键字**

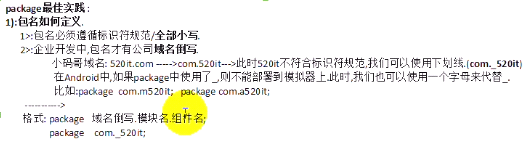




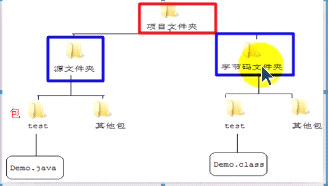
Javac –d . 文件名

运行时执行java pacekage名称.类名

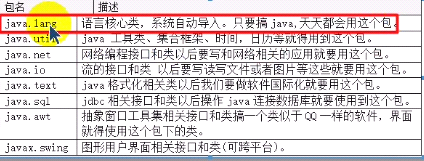
**04 package最佳实践**



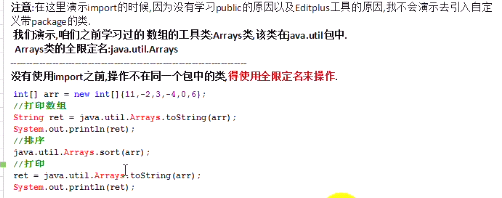
在开发中都是先建立包在定义类



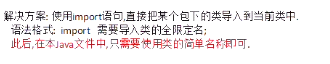
包的建立和字节码的建立时一一对应的。

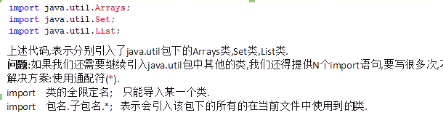


05 import语句



**Java.lang包自动被编译器引入**

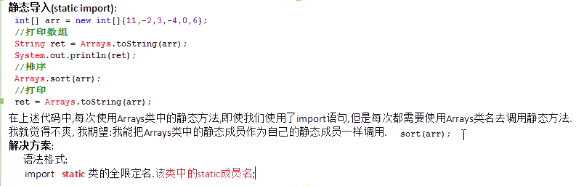




使用通配符\*也是使用哪个类引入哪个类，而不是全部引入，否则会占用很大的内存。

通配符也是语法糖，在编译的字节码文件也是转换为多个import语句

**06 static import静态导入**

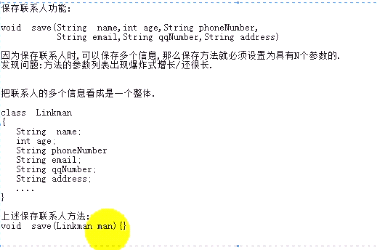


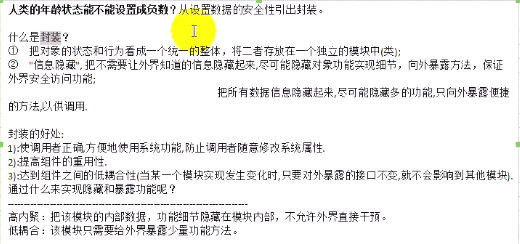
静态导入要导入要使用的成员，也可以使用通配符\*。

成员名可以是成员变量也可以是静态方法

通过反编译工具，其实所谓的静态导入也是一个语法糖，在实际开发中我们不使用静态导入，因为我们分不清方法或字段是当前类的还是导入的。

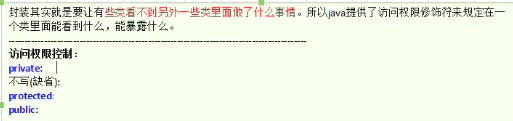
**07封装的思想**





封装就是使用访问修饰符来修饰成员变量和方法

**08权限访问修饰符**

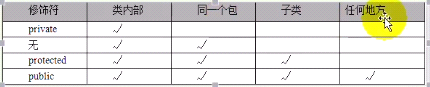


Private 表示私有的，表示类访问权限，只能在本类中访问，离开本类之后就不能直接访问。

不写（缺省） 表示包私有，表示包访问权限，只能在当前包内才能访问，c#中有internal关键字，是在同样的命名空间内存才能访问。

Protected 表示子类访问权限，同包中的可以访问，即使不同包但是有继承关系也可以访问。

Public表示全局的，可以公共访问权限



一般的字段都使用private修饰，表达隐藏，为了安全性

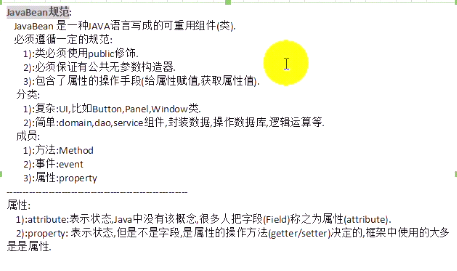
拥有实现细节的方法，一般使用private修饰，不希望外界看到方法的实现细节

一般的方法我们使用public修饰，供外界使用。

一般我们不用缺省，即使使用也仅仅是暴露给同包的其他类

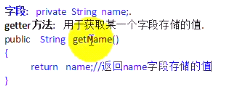
Protected一般在继承关系中父类需要把一个方法暴露给子类

**09java bean规范简述**

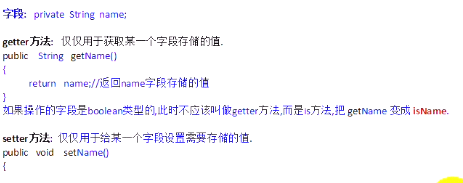


即使提供了带参数的构造器，也得提供无参数的构造器，java bean的规范

我们讲完封装之后，应该把对象中的信息隐藏起来（把类中的字段全部使用private修饰起来，其他类不能直接访问），为了能让外界访问到本类中的私有变量，我们专门提供getter以及setter方法



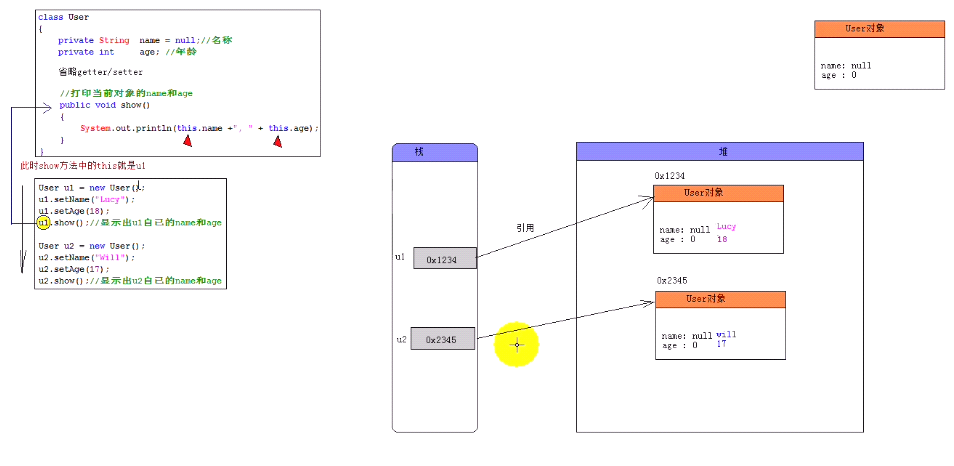
如果操作的字段是布尔类型的，此时不应该叫做getter方法，而是is方法

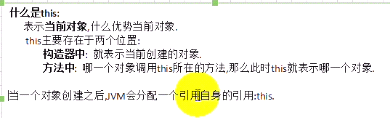


**10 引出this关键字**

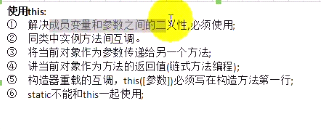
程序有就近原则的思想

**11理解this关键字**



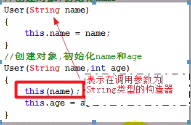


**12this关键字的使用**

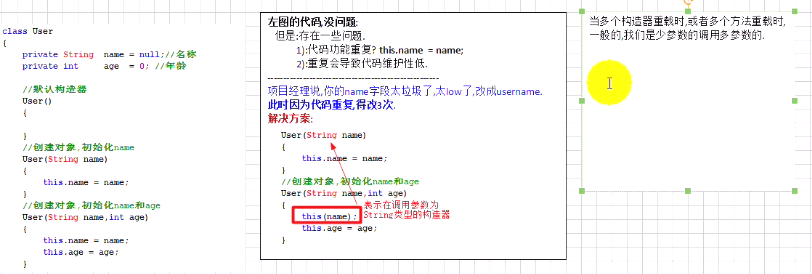


在类中非静态方法之间的调用最好使用this关键字，PHP的类中是强制使用否则找不到方法的错误。

**13使用this关键字解决构造器重载相互调用**

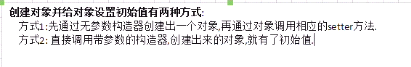


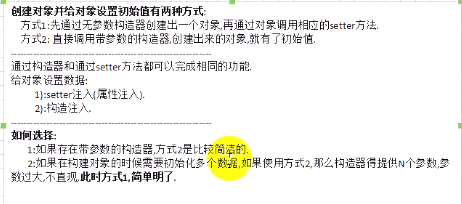
This可以用于构造器重载时的互相调用。而且只能出现在另一个构造器代码块的第一句话

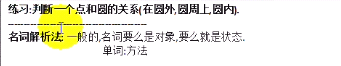


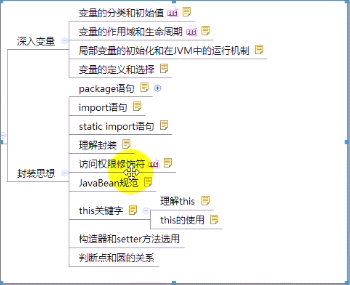
多参数构造器调用少参数的构造器

**14构造器和setter方法的选用**









成员变量有初始值，jvm会为成员变量初始化，而局部变量没有初始值，在初始化之前不能使用。