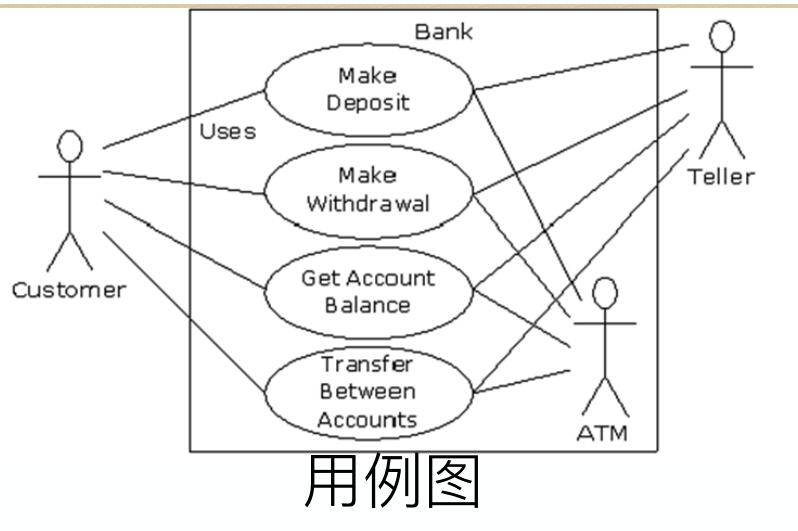
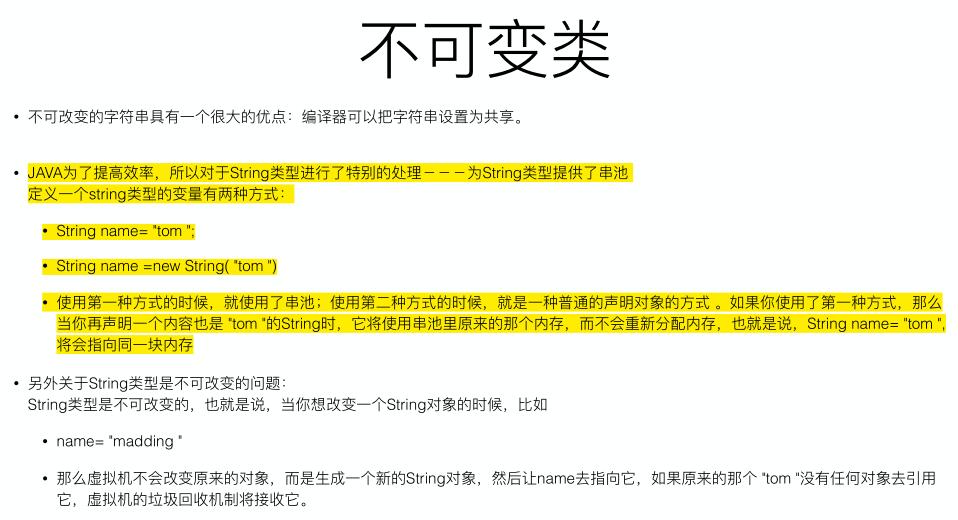
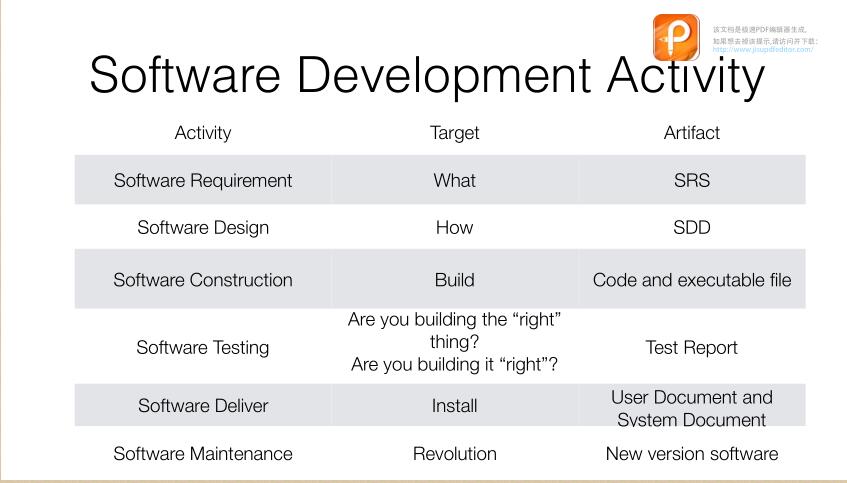
# 软工笔记

1. 用例图四要素：
2. 小人：参与者
3. 椭圆：用例
4. 线：连接
5. 框：系统边界



1. 考点：software development activity



## 关于句柄：

在Java里，任何东西都可看作对象。尽管将一切都“看作”对象，但操纵的标识符实际是指向一个对象的“句柄”（Handle），有的人将其称作一个“引用”，甚至一个“指针”。

主类型的数据成员可直接初始化，是变量，不是句柄。

       String s; 创建的只是句柄，并不是对象。若此时向s发送一条消息，就会获得一个错误（运行期），这是由于s实际并未与任何东西连接。

       创建句柄时，我们希望它同一个新对象连接。通常用new关键字达到这一目的。new的意思是：“把我变成这些对象的一种新类型”。

String s = new String("asdf");它不仅指出“将我变成一个新字串”，也通过提供一个初始字串，指出了“如何生成这个新字串”.  
String s = new String("xx");

这里在堆（heap）中创建一个string 对象，还会在文字池（string pool）里面创建一个"abc"对象；s是这个对象的引用变量（句柄），它不是对象，对象一般通过new创建。

字符串对象的创建

由于字符串对象的大量使用(它是一个对象，一般而言对象总是在heap分配内存)，Java中为了节省内存空间和运行时间，在编译阶段就把所有的字符串文字放到一个文字池(pool of literal strings)中，而运行时文字池成为常量池的一部分。文字池的好处，就是该池中所有相同的字符串常量被合并，只占用一个空间。

我们知道，对两个引用变量，使用==判断它们的值(引用)是否相等，即指向同一个对象：

String s1 = "abc" ;

String s2 = "abc" ;

这里 s1 == s2 成立  //因为比较的不是句柄，而是内容

**String s = new String("abc") ; 在编译阶段在文字池中创建“abc”对象，运行new时，将pool中的对象复制一份放到heap中，并且把heap中的这个对象的引用交给s持有，因此这条语句创建了2个String对象。**

String s1 = new String("abc") ;

String s2 = new String("abc") ;

这里 s1 == s2 不成立，s1.equals(s2)成立  //因为比较的是句柄，内容虽然相同但句柄不.

## 可修改性

* 构造方法是没办法覆盖的。
* 不要擅用继承，也不要滥用组合。继承的代价很大，继承没有灵活性，一旦确定了继承关系，以后就不可改变。
* **[必考]**类的初始化问题：对于静态变量、静态初始化块、变量、初始化块、构造器，它们的初始化顺序依次是（静态变量、静态初始化块）>（变量、初始化块）>构造器。

<http://www.360doc.com/content/11/1121/20/7874148_166290570.shtml>

(1)访问SubClass.main(),(这是一个static方法)，于是装载器就会为你寻找已经编译的SubClass类的代码（也就是SubClass.class文件）。在装载的过程中，装载器注意到它有一个基类（也就是extends所要表示的意思），于是它再装载基类。不管你创不创建基类对象，这个过程总会发生。如果基类还有基类，那么第二个基类也会被装载，依此类推。

(2)执行根基类的static初始化，然后是下一个派生类的static初始化，依此类推。这个顺序非常重要，因为派生类的“static初始化”有可能要依赖基类成员的正确初始化。

(3)当所有必要的类都已经装载结束，开始执行main()方法体，并用new SubClass（）创建对象

(4)类SubClass存在父类，则调用父类的构造函数，你可以使用super来指定调用哪个构造函数（也就是Beetle（）构造函数所做的第一件事）。基类的构造过程以及构造顺序，同派生类的相同。首先基类中各个变量按照字面顺序进行初始化，然后执行基类的构造函数的其余部分。

(5)对子类成员数据按照它们声明的顺序初始化，执行子类构造函数的其余部分。