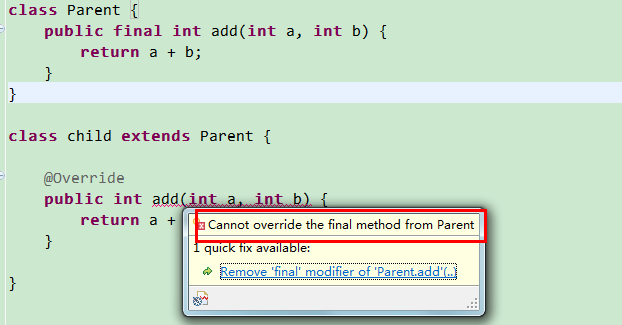
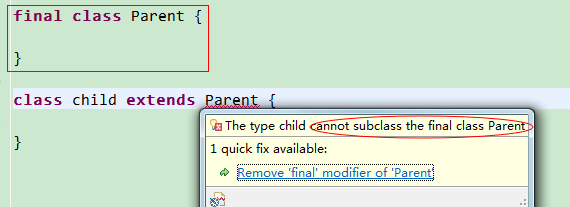
1. final关键字：final的变量的值不能被改变(一个被标记为final的变量代表它一旦被初始化之后就不会改动)
   1. final的成员变量
   2. final的局部变量(或形参)
2. final的方法不能被重写



1. final的类不能够被继承



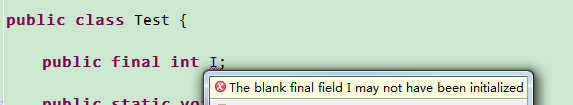
1. 静态的final变量是**常数**，常数变量的名称应该都是**大写字母，并以下划线字符分隔**。以Math.PI为例：

public static final double PI = 3.141592653389793;

说明：

1. public：可供各方读取
2. static：不需要Math的实例即可访问
3. final：PI值不可变

final变量(常数)必须初始化，否则编译器报错

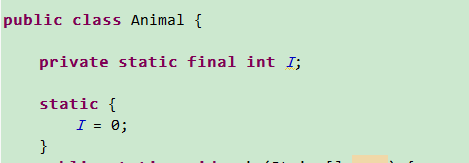


有两种初始化方法：

1. 声明的时候初始化



1. 在静态初始化程序中(静态语句块中)



1. 对于final域，编译器和处理器要遵守两个重排序规则
   1. 写final域的重排序规则：在构造函数内对一个final域的写入，与随后把这个被构造对象的引用赋值给一个引用变量，这两个操作之间不能重排序
   2. 写final域的重排序规则：在构造函数内对一个final**引用**的对象的成员域的写入，与随后把这个被构造对象的引用赋值给一个引用变量，这两个操作之间不能重排序
   3. 读final域的重排序规则：初次读一个包含final域的对象的引用，与随后初次读这个final域，这两个操作之间不能重排序

写final域的重排序规则可以确保：在对象引用为任意线程可见之前，对象的final域已经被正确初始化过了，但是普通域不具有这个保障

读final域的重排序规则可以确保：在读一个对象的final域之前，一定会先读包含这个这个final域的对象的引用