1. **final**

根据程序上下文环境，Java关键字final有“这是无法改变的”或者“终态的”含义，它可以修饰非抽象类、非抽象类成员方法和变量。你可能出于两种理解而需要阻止改变：设计或效率。

1. final类不能被继承，没有子类，final类中的方法默认是final的。
2. final方法不能被子类的方法覆盖，但可以被继承。
3. final成员变量表示常量，只能被赋值一次，赋值后值不再改变。
4. final不能用于修饰构造方法。

注意：父类的private成员方法是不能被子类方法覆盖的，因此private类型的方法默认是final类型的

1、final类  
final类不能被继承，因此final类的成员方法没有机会被覆盖，默认都是final的。在设计类时候，如果这个类不需要有子类，类的实现细节不允许改变，并且确信这个类不会载被扩展，那么就设计为final类。

2、final方法  
如果一个类不允许其子类覆盖某个方法，则可以把这个方法声明为final方法。使用final方法的原因有二：  
第一、把方法锁定，防止任何继承类修改它的意义和实现。  
第二、高效。编译器在遇到调用final方法时候会转入内嵌机制，大大提高执行效率。  
        例如：

public class Test1 {   
public static void main(String[] args) {   
    // TODO 自动生成方法存根   
}   
public void f1() {   
    System.out.println("f1");   
}   
//无法被子类覆盖的方法   
public final void f2() {   
    System.out.println("f2");   
}   
public void f3() {   
    System.out.println("f3");   
}   
private void f4() {   
    System.out.println("f4");   
}   
}   
public class Test2 extends Test1 {   
      
public void f1(){       
    System.out.println("Test1父类方法f1被覆盖!");   
}   
public static void main(String[] args) {   
    Test2 t=new Test2();   
    t.f1();      
    t.f2(); //调用从父类继承过来的final方法   
    t.f3(); //调用从父类继承过来的方法   
    //t.f4(); //调用失败，无法从父类继承获得   
}   
}

3、final变量（常量）  
用final修饰的成员变量表示常量，值一旦给定就无法改变！  
final修饰的变量有三种：静态变量、实例变量和局部变量，分别表示三种类型的常量。  
 从下面的例子中可以看出，一旦给final变量初值后，值就不能再改变了。  
  另外，final变量定义的时候，可以先声明，而不给初值，这种变量也称为final空白，必须在初始化对象的时候赋值。但是，final空白在final关键字final的使用上提供了更大的灵活性，为此，一个类中的final数据成员就可以实现依对象而有所不同，却有保持其恒定不变的特征。

package org.leizhimin;   
  
public class Test3 {   
        private final String S = "final实例变量S";   
        private final int A = 100;   
        public final int B = 90;   
  
        public static final int C = 80;   
        private static final int D = 70;   
  
        public final int E; //final空白,必须在初始化对象的时候赋初值   
  
        public Test3(int x) {   
                E = x;   
        }   
  
        /\*\*   
         \* @param args   
         \*/   
        public static void main(String[] args) {   
                Test3 t = new Test3(2);   
                //t.A=101;    //出错,final变量的值一旦给定就无法改变   
                //t.B=91; //出错,final变量的值一旦给定就无法改变   
                //t.C=81; //出错,final变量的值一旦给定就无法改变   
                //t.D=71; //出错,final变量的值一旦给定就无法改变   
  
                System.out.println(t.A);   
                System.out.println(t.B);   
                System.out.println(t.C); //不推荐用对象方式访问静态字段   
                System.out.println(t.D); //不推荐用对象方式访问静态字段   
                System.out.println(Test3.C);   
                System.out.println(Test3.D);   
                //System.out.println(Test3.E); //出错,因为E为final空白,依据不同对象值有所不同.   
                System.out.println(t.E);   
  
                Test3 t1 = new Test3(3);   
                System.out.println(t1.E); //final空白变量E依据对象的不同而不同   
        }   
  
        private void test() {   
                System.out.println(new Test3(1).A);   
                System.out.println(Test3.C);   
                System.out.println(Test3.D);   
        }   
  
        public void test2() {   
                final int a;     //final空白,在需要的时候才赋值   
                final int b = 4;    //局部常量--final用于局部变量的情形   
                final int c;    //final空白,一直没有给赋值.      
                a = 3;   
                //a=4;    出错,已经给赋过值了.   
                //b=2; 出错,已经给赋过值了.   
        }   
}

4、final参数  
  当函数参数为final类型时，你可以读取使用该参数，但是无法改变该参数的值。

public class Test4 {   
        public static void main(String[] args) {   
                new Test4().f1(2);   
        }   
  
        public void f1(final int i) {   
                //i++;    //i是final类型的,值不允许改变的.   
                System.out.print(i);   
        }   
}