参考： <http://www.cnblogs.com/dudu837/archive/0001/01/01/1455138.html>

我的感受是，静态方法就是对象无关的，当要进行的操作不是针对具体对象的操作就定义为静态方法，如果操作时对对象进行操作就定义为实例方法。参考博文的下面的话可以有个更深刻的理解。

*假设有一个会员类，该类要有删除会员的功能，那么你可能最少也要提供两上删除方法，一个是删除当前实例所对应的会员，一个是删除指定的会员。前者是与具体实例相关的，后者则不与具体实例发生关系。问题就在这里（为什么要提供两个删除会员的方法）。你当然可以用这个删除方法既删除当前实例所对应的会员，也可以用这个方法删除任一指的会员。可是如果从面向对象的设计方法上考虑，会员张三有权利去删除会员李四吗？你高兴时可以将自己的笔记本送给别人，你能把我的笔记本送给别人吗？当然不行，因为我的笔记本是我的，不是你的，你没有权利这样做。会员张三可以调用自己的删除方法把自己删除，而要删除会员李四的时候，要不将会员李四实例化了后让李四调用李四自己的删除方法，要不，就由会员类来删除会员李四，反正张三是不能把李四给删除掉，在现实世界中，这是符合逻辑的。*

静态类

静态类不能被实例化，静态类中所有的成员都是静态的。只要静态类没有被重新编译，静态类的成员变量一旦被赋值则一直存储在内存中，知道程序结束或者被重新编译。

静态构造函数

静态构造函数不可以被直接调用，当创建类实例或者引用任何静态成员之前，静态构造函数被自动执行，并且只执行一次。

静态构造函数无访问修饰符，无参数，只有一个static标志。

静态构造函数可用于静态类，也可用于非静态类。

静态类可以有静态构造函数，静态构造函数不可被继承（应该意思是静态类可以有静态构造函数或者是无构造函数）

静态方法

非静态方法只能调用非静态方法和其他数据成员，而静态方法则能调用静态方法（及其他静态数据成员）和非静态方法（及其他静态数据成员）。