**递归算法**

* 程序调用自身的编程技巧称为递归（recursion）

一个过程或函数在其定义或说明中有直接或间接调用自身的一种方法

* 递归问题的特点：

-一个问题可被分解为若干层简单的子问题

-子问题和其上层问题的解决方案一致

-外层问题的解决依赖于子问题的解决

* 递归结构包括两个部分：

-递归结束条件

解答：什么时候不调用自身方法。如果没有条件，将陷入死循环

-递归体

解答：什么时候需要调用自身方法

注意：

在程序中，能不使用递归就不使用递归

使用递归的时候会加大资源的消耗

如果递归的层次比较深，会造成粘溢出

如果不使用递归无法解决问题的话就必须使用递归

比如输出某个磁盘目录下的所有文件名称

* 递归示例：

使用实现斐波那契数列

**import** java.util.Scanner;  
**public class** While{  
 **public static void** main(String[] args){  
  
 Scanner input = **new** Scanner(System.***in***);  
 System.***out***.println(**"请输入斐波那契数列的个数："**);  
 **int** count = input.nextInt();  
 **for** (**int** i=1;i<=count;i++){  
 System.***out***.print(*getNumber*(i)+**"\t"**);  
 }  
 }  
  
 **public static int** getNumber(**int** number){  
 **if**(number==1 || number==2){  
 **return** 1;  
 }**else**{  
 **return** *getNumber*(number-1)+*getNumber*(number-2);  
 }  
 }  
}