一.File 类

java.io.File 类是文件、目录(文件夹)路径名的抽象表示,就是把硬盘中的文件、文件夹的路径抽象封装成一个 File 类,主要用于文件和目录的创建、查找和删除等操作。并且 File 类是一个和系统无关的类,任何操作系统下都可以使用这个类。

- 1.File 类的四个静态成员变量
- 2.File 类的构造方法
- 3.File 类的常用成员方法
 - 获取功能的方法
 - 判断功能的方法
 - 创建、删除功能的方法
 - File 类遍历文件夹(目录)功能的方法

二.递归

递归:指在当前方法内自己调用自己。递归分为以下两种:直接递归和间接递归。

• 直接递归: 方法自身调用自己

```
private static void A() {
    System.out.println("A方法的直接递归调用!");
    A();
}
```

• 间接递归: A 方法调用 B 方法, B 方法调用 C 方法, C 方法再调用 A 方法

```
private static void A() {
    System.out.println("A方法中调用B方法");
    B();
}

private static void B() {
    System.out.println("B方法中调用C方法");
    C();
}

private static void C() {
    System.out.println("C方法中调用A方法");
    A();
}
```

#.注意事项:

- (1).递归一定要有条件限定,保证递归能够停止下来,并且递归调用次数不能太多,否则会发生栈内存溢出异常(java.lang.StackOverflowError)
- (2).构造方法,禁止递归。构造方法一直递归会导致内存中创建无数个对象,直接编译报错。

1.递归累加求和

```
栈内存:
public class RecursiveSum {
                                                  sum(1):
   public static void main(String[] args) {
       System.out.println(sum( n: 5));
                                                          return 1;
                                          回溯(
                                                  sum(2):
   private static int sum(int n) {
                                                      return 2 + sum(1);
                                                  sum(3):
       // 递归结束条件: 获取到1的时候结束
       if (n == 1) {
                                                      return 3 + sum(2);
           return 1;
                                                  sum(4):
                                                      return 4 + sum(3);
       return n + sum( n: n - 1);
                                                                  调用
                                                  main:
}
                                                      return 5 + sum(4);
```

- 2.递归求阶乘
- 3.递归打印多级目录

三.综合案例

- 1.文件搜索
- 2.文件过滤器 FileFilter