## 1.基本原理

//Runtime.getRuntime().exec("calc.exe");

## 2.一些无用的语句

//System.out.println(method.getName().toString());

//method.invoke(null, "calc.exe");

//method.in(, "calc.exe");

//method.invoke(new Class[] {method.getClass()}, new Object[] {"exec"});

//method.invoke(Runtime.class, "exec");

//runtime.getMethod(name, parameterTypes)

## 3. 分步实现 反射

/\* Class runtime= Runtime.class;

Method method= runtime.getMethod("getRuntime",null);

System.out.println(method.invoke(null, null));

Object resobj=method.invoke(null, null);

Class relruntime=resobj.getClass();

Method resmethod= relruntime.getMethod("exec",new Class[] {String.class});

resmethod.invoke(resobj,"calc.exe");\*/

## 4.使用1行java代码进行整合

// Runtime.class.getMethod("getRuntime",null).invoke(null, null).getClass().getMethod("exec",new Class[] {String.class}).invoke(Runtime.class.getMethod("getRuntime",null).invoke(null, null), "cmd.exe /c echo 123>E:/1.txt");

## 说明

1. Runtime.class.getMethod("getRuntime",null).invoke(null, null) 的作用是创建一个Runtime 对象 相当于 Runtime.getRuntime();
2. Runtime.class.getMethod("getRuntime",null).invoke(null, null).getClass().getMethod("exec",new Class[] {String.class}) 的作用是 反射获取 exec方法

3.Runtime.class.getMethod("getRuntime",null).invoke(null, null).getClass().getMethod("exec",new Class[] {String.class}).invoke(Runtime.class.getMethod("getRuntime",null).invoke(null, null), "cmd.exe /c echo 123>E:/1.txt"); 的作用相当于Runtime.getRuntime().exec("calc.exe");