

# Callable

Callable 是三种线程创建最强的一种方式,它不但拥有异常抛出,还具有返回值,这是之前两种[Thread Runnable]不具备的.

但Callable 不仅仅只能做这些,Callable还可以依靠其他类[FutureTask]做出更强大的事情.

FutureTask

常用方法:

```
public boolean cancel(boolean mayInterruptIfRunning)
```

尝试取消任务

```
public V get()
```

等待计算完成,然后检索其结果。

```
public boolean isCancelled()
```

如果此任务在正常完成之前被取消,则返回 true 。

```
public boolean isDone()
```

返回true如果任务已完成。 完成可能是由于正常终止,异常或取消 - 在所有这些情况下,此方法将返回true 。

若创建Callable 不使用futureTask则异常并不会显示出来

推测:

```
public class Main {  
    public static void main(String[] args) {  
        Test1 test1 = new Test1();  
        test1.show();  
        test1.get();  
    }  
}
```

```
class Test1{  
    Exception ex;  
    public void show(){ //FutureTask 调用完show之后将发生的异常保存起来,然后调用  
get再将异常打印
```

```
        //若不调用则不打印
```

```
        try {  
            Test2.show();  
        } catch (Exception e) {  
            this.ex = e;  
        }  
    }
```

```
    public void get(){  
        ex.printStackTrace();  
    }
```

```
    }  
}  
class Test2{  
    public static void show() throws Exception {  
        throw new Exception("抛出");  
    }  
}
```