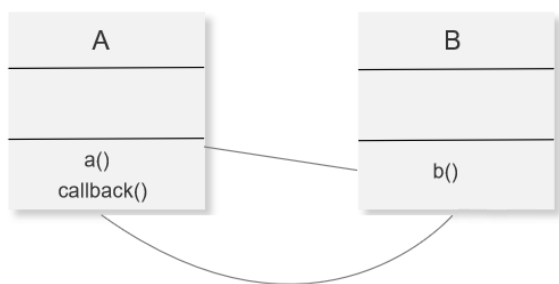


## 1. 定义

在计算机程序设计中，回调函数，或简称回调（Callback 即call then back 被主函数调用运算后会返回主函数），是指通过函数参数传递到其它代码的，某一块可执行代码的引用。这一设计允许了底层代码调用在高层定义的程序。

在java的回调函数，以类A调用类B方法为例

在类A中方法a调用了类B中的b方法，b方法在执行完成之后，会执行类A中的callback()方法，这个callback方法就是回调函数，表明类B的b方法执行完成。



## 2.同步回调

同步回调的话会等待回调函数执行完成之后才执行其他的业务。

以儿子要坐火车出远门，跟妈妈告别，到了学校之后打电话告诉自己到学校这件事为例：

接口：

```
public interface Callable {
    /**
     * 回调函数
     */
    public void call();
}
```

妈妈：

//定义类Mother实现了Callable接口，实现了call()回调函数的具体内容

```
public class Mother implements Callable {
    Son son;
    public Mother(Son son)
    {
        this.son=son;
    }
    //表示妈妈和儿子的分别函数，儿子在这期间搭乘火车离开
    public void parting()
    {
        System.out.println("开始执行同步回调函数");
        son.rideTrain(this);
        System.out.println("同步回调函数执行完成");
    }
}
```

```

@Override
public void call() {
    System.out.println("儿子到学校了");
}
}

```

儿子:

```

public class Son {
    public void rideTrain(Callable callable)
    {
        try {
            //模拟坐火车
            Thread.sleep(5000);
        } catch (Exception e) {
            e.printStackTrace();
        }
        callable.call();//到了打电话给妈妈
    }
}

```

测试类:

```

public class Test {
    public static void main(String[] args) {
        Son jack=new Son();
        Mother mother=new Mother(jack);
        mother.parting();
    }
}

```

运行结果:

开始执行同步回调函数  
 儿子到学校了  
 同步回调函数执行完成

### 3 异步回调

异步调用发生在某些业务十分耗时，又需要立即返回结果，这个时候需要开辟新的线程处理耗时业务，业务处理完了，再去回调回调函数，处理其他业务。

还是以妈妈和儿子为例, 区别是妈妈中的parting函数不同实现，开启了线程处理函数。

```

public class Mother implements Callable {
    Son son;
    public Mother(Son son)
    {
        this.son=son;
    }

    public void parting()
    {
        System.out.println("开始执行异步回调函数");
    }
}

```

```
new Thread(new Runnable() {  
    @Override  
    public void run() {  
        son.rideTrain(Mother.this);  
    }  
}).start();  
System.out.println("异步回调函数执行完成");//开启线程处理儿子坐火车的函数，这条打印语句会在回调函数执行完成前执行。  
}
```

```
@Override  
public void call() {  
    System.out.println("儿子到学校了");  
}  
}
```

最后的运行结果：

开始执行异步回调函数

异步回调函数执行完成

儿子到学校了

#### **4 总结**

- 1、 回调函数可以用于事件监听，当发生相对应的事件时系统可以自动调用对应的函数
- 2 、回调函数可以用于中断处理。