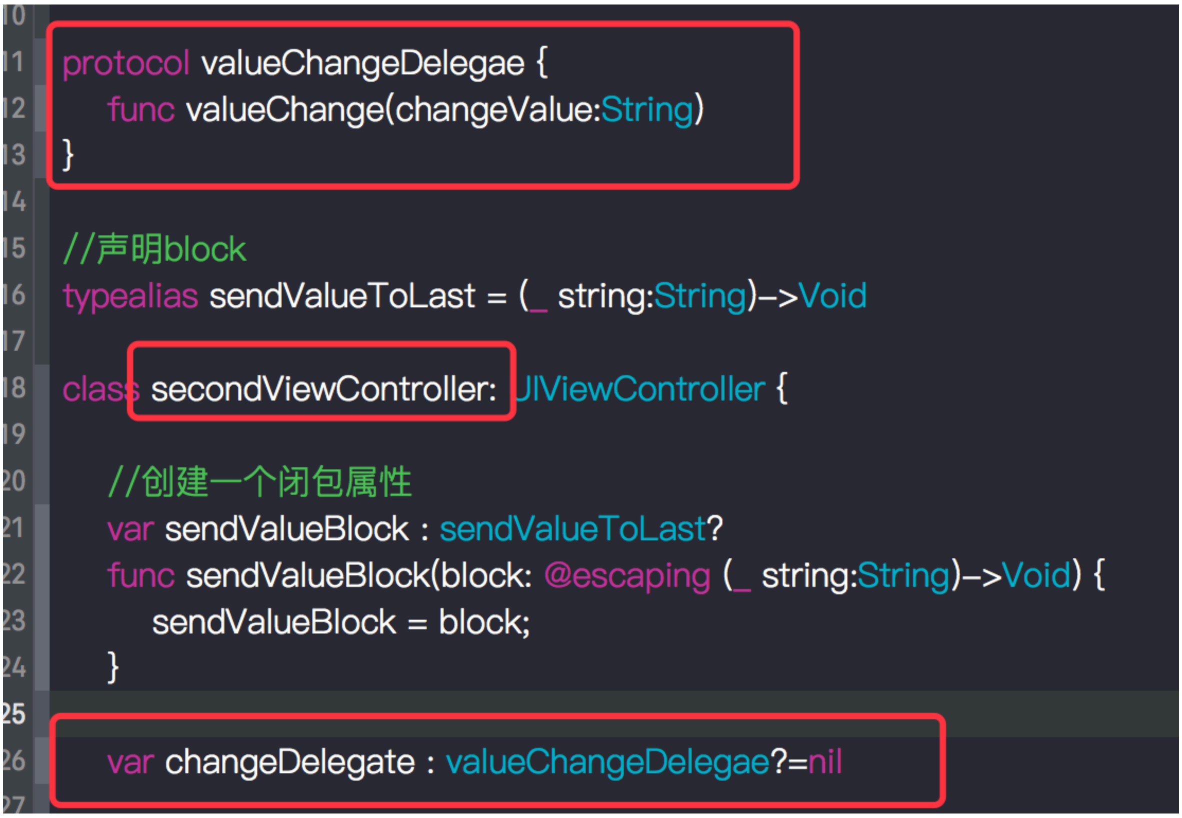
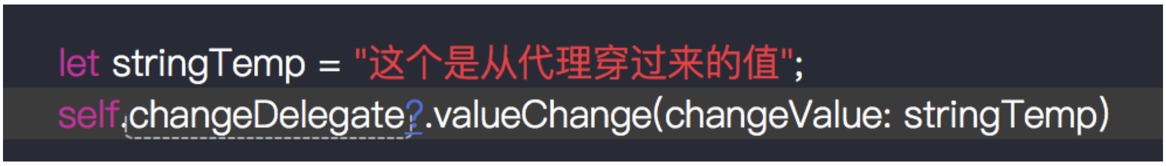


```
protocol valueChangeDelegae {
    func valueChange(changeValue:String)
}

//声明block
typealias sendValueToLast = (_ string:String)->Void

class secondViewController: UIViewController {

    //创建一个闭包属性
    var sendValueBlock : sendValueToLast?
    func sendValueBlock(block: @escaping (_ string:String)->Void) {
        sendValueBlock = block;
    }

    var changeDelegate : valueChangeDelegae?=nil
```

```
protocol valueChangeDelegae {
    func valueChange(changeValue:String)
}
```

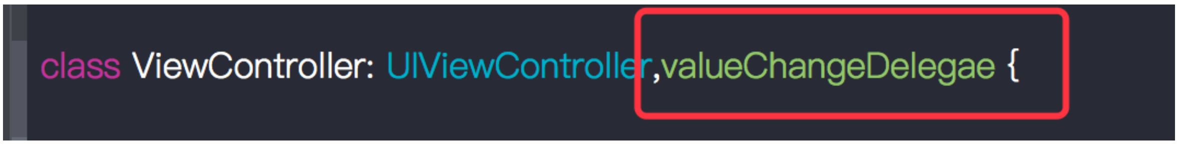
```
 var changeDelegate : valueChangeDelegae?=nil
```



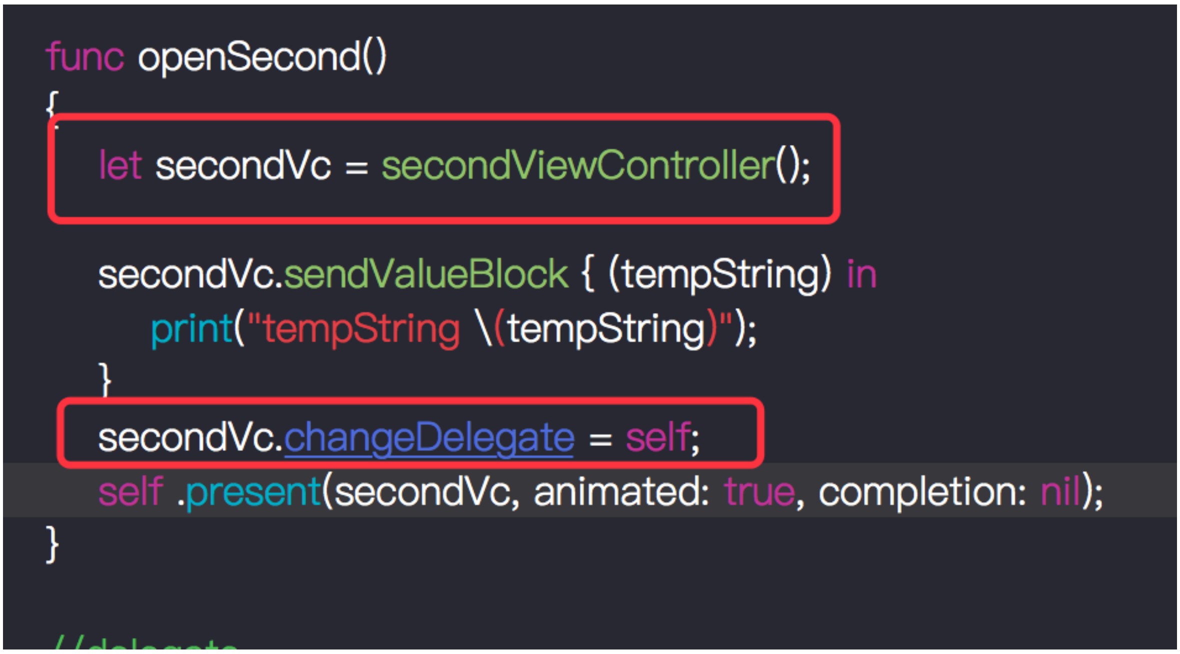
```
let stringTemp = "这个是从代理穿过来的值";
self.changeDelegate?.valueChange(changeValue: stringTemp)
```

```
 let stringTemp = "这个是从代理穿过来的值";
        self.changeDelegate?.valueChange(changeValue:
stringTemp)
```

ViewController

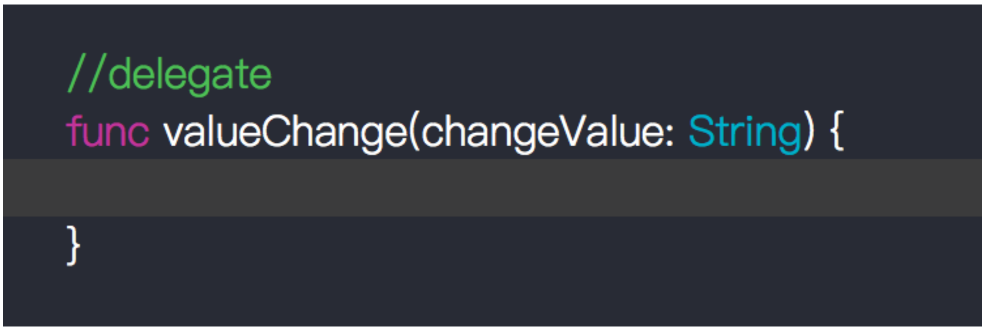


```
class ViewController: UIViewController,valueChangeDelegae {
```



```
func openSecond()
{
    let secondVc = secondViewController();

    secondVc.sendValueBlock { (tempString) in
        print("tempString \(tempString)");
    }
    secondVc.changeDelegate = self;
    self .present(secondVc, animated: true, completion: nil);
}

//delegate
```

```
let secondVc = secondViewController();

        secondVc.sendValueBlock { (tempString) in
            print("tempString \(tempString)");
        }
        secondVc.changeDelegate = self;
```



```
//delegate
func valueChange(changeValue: String) {

}
```

```
  //delegate
```

```
func valueChange(changeValue: String) {

}
```