# 计算器实验报告

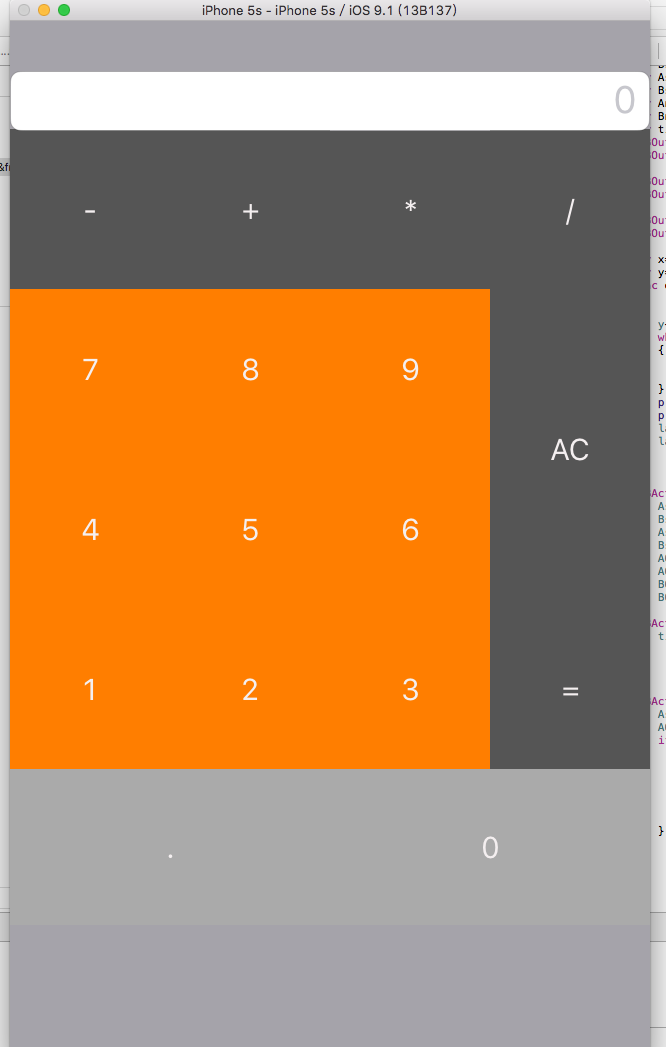
## 实验目的

为了熟悉swift编程，将所学的理论知识付诸于实践，我们将在mac操作系统下开发一个简易的计算器。

## 实验要求

此计算器可以实现基本的加、减、乘、除等数学运算。

1. **程序主界面**

****

1. **功能实现代码**

**4.1结构体**

本计算器将加减乘除四则基本运算的算法放在结构体calcula内，具体实现代码如下：

import Foundation  
  
struct calcula {  
 //设置当前运算算法为未选择

private var NowArithmetic: arithmetic = .unchoose  
   
 mutating func setNowarithmetic( provisional:arithmetic ){  
 NowArithmetic = provisional  
 }  
 //定义方法获取当前算法。  
 func getNowarithmetic() -> arithmetic{  
 return NowArithmetic  
 }  
 //设置加减乘除四则运算  
 enum arithmetic {  
 case add  
 case subtraction  
 case multiplication  
 case division  
 case unchoose  
 }  
 //定义获取结果方法，其中的两个参数分别是第一个输入的数和第二个输入的数。  
 func getResult(firstin firstin:Double, secondin:Double) -> String {  
 var back = ""  
 switch NowArithmetic {  
 case .add:  
 back = "\(firstin + secondin)"  
 case .subtraction:  
 back = "\(firstin - secondin)"  
 case .multiplication:  
 back = "\(firstin \* secondin)"  
 case .division where secondin != 0:  
 back = "\(firstin / secondin)"  
 case .unchoose:  
 back = "错误：未选择算法"  
 default:  
 back = "0 不能做被除数！"  
 }  
 return back  
 }  
}

**4.2 程序主体代码**

计算器的主要代码放在ViewController内，它包括了按钮的绑定，屏幕显示等功能。

import UIKit  
  
class ViewController: UIViewController {  
  
 override func viewDidLoad() {  
 super.viewDidLoad()  
 // Do any additional setup after loading the view, typically from a nib.  
 }  
  
 override func didReceiveMemoryWarning() {  
 super.didReceiveMemoryWarning()  
 // Dispose of any resources that can be recreated.  
 }  
   
 var calculator = calcula() //计算  
   
 var displaycache = "" //显示缓存  
   
 var cache = 0.0 //缓存  
   
 var cache2:Double?  
   
 var pointcount = 0  
 //此处解决了小数点可以无限输入的Bug。  
 func loadResult(number:String){  
 if number == "."{  
 pointcount++  
 }  
 if number == "." && pointcount > 1{  
 return  
 }  
   
 displaycache += number  
 OutPutResult.text = displaycache  
 }  
 /\*  
 求值  
 \*/  
 func getResult(){  
 var Result = ""  
   
 pointcount = 0  
   
 if !displaycache.isEmpty {  
 let temporary = displaycache as NSString //临时  
 cache = temporary.doubleValue  
 displaycache = "" //显示缓存清空  
 }  
   
 let temporary = cache  
   
 if let previousNumber = cache2 {  
   
 Result = calculator.getResult(firstin: previousNumber, secondin: temporary)  
 OutPutResult.text = Result  
 let tem = Result as NSString  
 cache2 = tem.doubleValue  
   
 }else{  
 cache2 = cache  
 cache = 0.0  
 }  
 }  
 //绑定按钮  
 @IBAction func Button\_1(sender: UIButton) {  
 loadResult("1")  
 }  
 @IBAction func Button\_2(sender: UIButton) {  
 loadResult("2")  
 }  
 @IBAction func Button\_3(sender: UIButton) {  
 loadResult("3")  
 }  
 @IBAction func Button\_4(sender: UIButton) {  
 loadResult("4")  
 }  
 @IBAction func Button\_5(sender: UIButton) {  
 loadResult("5")  
 }  
 @IBAction func Button\_6(sender: UIButton) {  
 loadResult("6")  
 }  
 @IBAction func Button\_7(sender: UIButton) {  
 loadResult("7")  
 }  
 @IBAction func Button\_8(sender: UIButton) {  
 loadResult("8")  
 }  
 @IBAction func Button\_9(sender: UIButton) {  
 loadResult("9")  
 }  
 @IBAction func Button\_0(sender: UIButton) {  
 loadResult("0")  
 }  
 @IBAction func Button\_point(sender: UIButton) {  
 loadResult(".")  
 }  
 @IBAction func Button\_add(sender: UIButton) {  
 calculator.setNowarithmetic(.add)  
 getResult()  
 }  
 @IBAction func Button\_subtraction(sender: UIButton) {  
 calculator.setNowarithmetic(.subtraction)  
 getResult()  
 }  
 @IBAction func Button\_multiplication(sender: UIButton) {  
 calculator.setNowarithmetic(.multiplication)  
 getResult()  
 }  
 @IBAction func Button\_division(sender: AnyObject) {  
 calculator.setNowarithmetic(.division)  
 getResult()  
 }  
 @IBAction func Button\_equal(sender: UIButton) {  
 getResult()  
 }  
 @IBAction func Button\_eliminate(sender: UIButton) {  
 pointcount = 0  
 displaycache = ""  
 cache = 0.0  
 cache2 = nil  
 calculator.setNowarithmetic(.unchoose)  
 OutPutResult.text = "0"  
 }

//绑定输出文本框  
 @IBOutlet weak var OutPutResult: UITextField!  
  
  
}

1. **总结**

通过此次实验，我掌握了一门新的编程语言，成功的运用所学的知识解决了一些常见的问题，例如小数点的控制和零不能做被除数等。但是在此次实验中还有很多不足的地方，例如，功能太少，像百分号、平方等还没有做。