实验编号： 3 **四川师大《IOS》实验报告 2018** 年 **9** 月 **19** 日

### **计算机科学学院** 2016 级 4 班 实验名称： Date、String、文件、URL \_

姓名：\_\_\_\_蒋宇童\_\_\_\_\_ 学号：\_2016110419\_\_\_\_\_\_\_\_\_ 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验\_三\_ \_\_\_\_\_\_** Date、String、文件、URL **\_\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 掌握Date、String的定义以及使用；
3. 掌握文件、URL的定义以及使用；
4. 实验要求
5. 认真填写实验报告，要求附加部分运行界面和主要代码；
6. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
7. 实验内容

* （Date、String、文件、URL）

1. 显示当前准确的中文时间，包括北京、东京、纽约、伦敦，格式为（2016年9月28日星期三 上午10:25）
   1. 显示中文需要设置locale
2. 处理字符串
   1. 新建字符串：“Swift is a powerful and intuitive programming language for iOS, OS X, tvOS, and watchOS.”；
   2. 返回字符串从第6个字符到第20个字符的子串；
   3. 删除其中所有的OS字符；
3. 将1、2题的时间和字符串存入一个字典中，并存入沙盒中的Document某文件中；
4. 从网上下载一张照片并保存到本地沙盒的Document的某文件中；
5. 从网上查找访问一个JSon接口文件，并采用JSONSerialization和字典对其进行简单解析；
6. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）

* （Date、String、文件、URL）

1. 显示当前准确的中文时间，包括北京、东京、纽约、伦敦，格式为（2016年9月28日星期三 上午10:25）
   1. 显示中文需要设置locale

* 程序代码：

/\*

1. 显示当前准确的中文时间，包括北京、东京、纽约、伦敦，格式为（2016年9月28日星期三 上午10:25）

a) 显示中文需要设置locale

\*/

func getDate(date: Date, zone: Int = 0) -> String {

let formatter = DateFormatter() //实例化格式化类

formatter.dateFormat = "yyyy年MM月dd日EEEE aa KK:mm" //指定格式化的格式

formatter.locale = Locale.current //设置当前位置，可以将对应的星期数和12小时制的上下午设置为中文

if zone >= 0 { //当传入的为正数时，在东半区

formatter.timeZone = TimeZone(abbreviation: "UTC+\(zone):00")

} else { //当传入的为负数时，在西半区

formatter.timeZone = TimeZone(abbreviation: "UTC\(zone):00")

}

let dateString = formatter.string(from: date) //将传入的日期格式化为字符串

return dateString

}

let now = Date() //获取当前时间日期

let beijing = getDate(date: now, zone: 8) //获取当前北京的时间

print("北京: \(beijing)")

let tokyo = getDate(date: now, zone: 9) //获取当前东京的时间

print("东京: \(tokyo)")

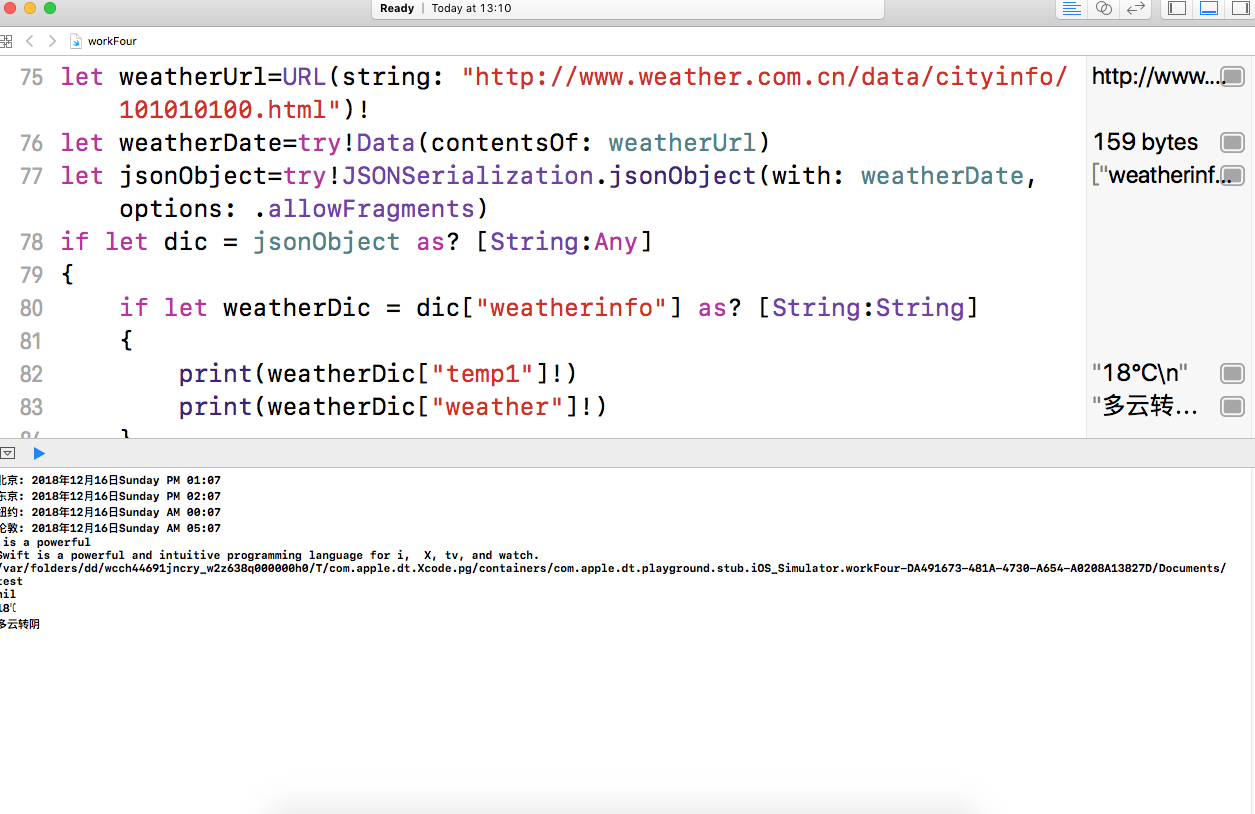
let newYork = getDate(date: now, zone: -5) //获取当前纽约的时间

print("纽约: \(newYork)")

let london = getDate(date: now) ////获取当前伦敦的时间

print("伦敦: \(london)")

* 运行结果：



1. 处理字符串
   1. 新建字符串：“Swift is a powerful and intuitive programming language for iOS, OS X, tvOS, and watchOS.”；
   2. 返回字符串从第6个字符到第20个字符的子串；
   3. 删除其中所有的OS字符；

* 程序代码：

/\*

2. 处理字符串

a) 新建字符串：“Swift is a powerful and intuitive programming language for iOS, OS X, tvOS, and watchOS.”；

b) 返回字符串从第6个字符到第20个字符的子串；

c) 删除其中所有的OS字符；

\*/

var str="Swift is a powerful and intuitive programming language for iOS, OS X, tvOS, and watchOS."

let startIndex=str.index(str.startIndex,offsetBy: 5)

let endIndex=str.index(str.startIndex,offsetBy:19)

var str2=str[startIndex...endIndex]

print(str2)

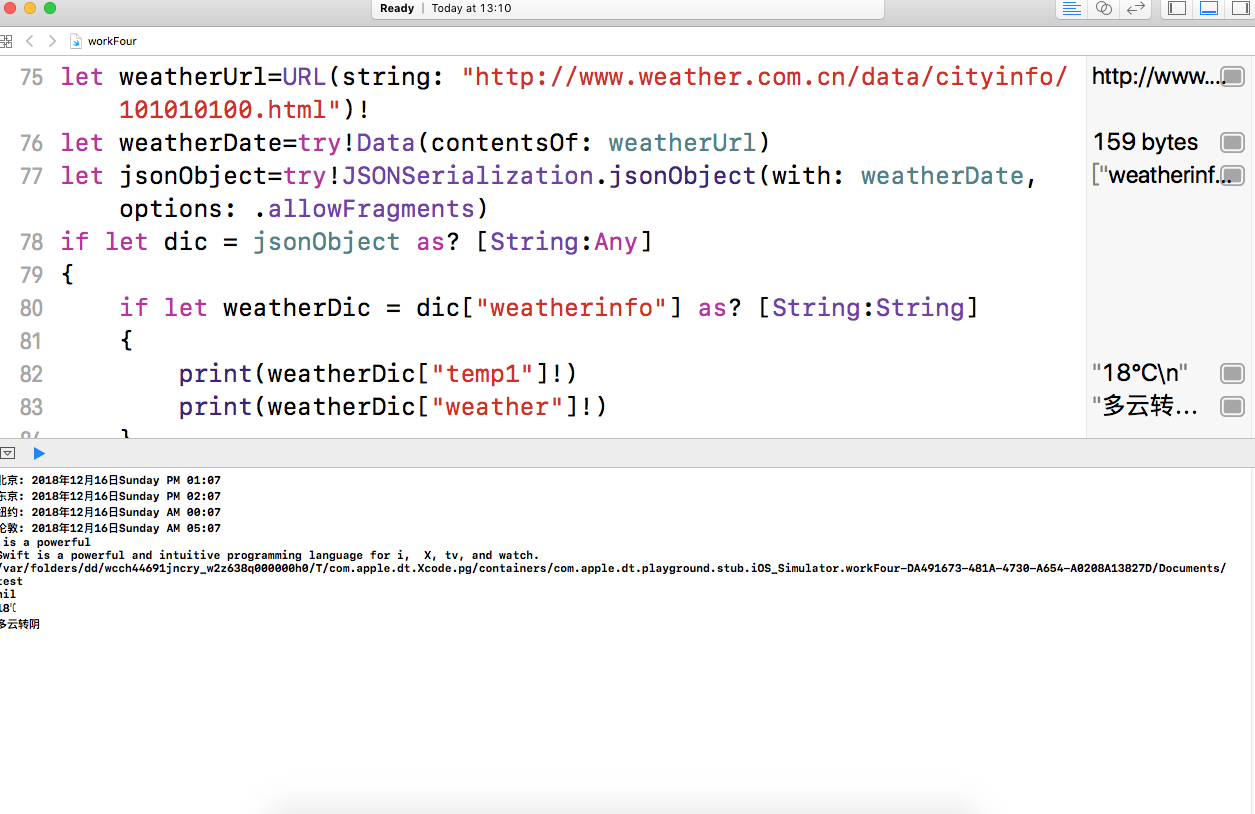
var str3=str.replacingOccurrences(of: "OS", with: "")

print(str3)

print(kk as Any)

}

* 运行结果：



1. 将1、2题的时间和字符串存入一个字典中，并存入沙盒中的Document某文件中；

* 程序代码：

/\*

3. 将1、2题的时间和字符串存入一个字典中，并存入沙盒中的Document某文件中；

\*/

var dictionary = [["时间":beijing,"字符串":str],["时间":tokyo,"字符串":str2],["时间":newYork,"字符串":str3]]as AnyObject

let fileManager=FileManager.default

if var docPath=fileManager.urls(for: .documentDirectory, in: .userDomainMask).first

{

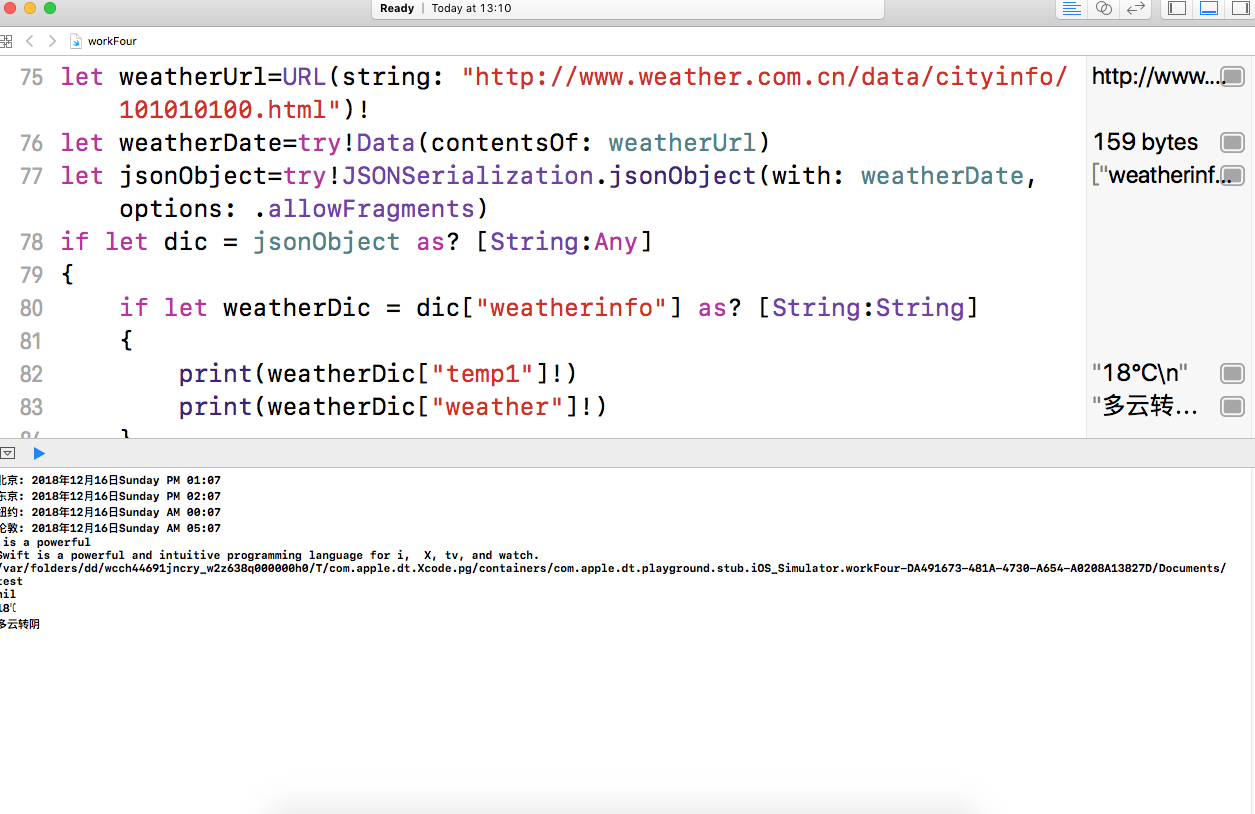
docPath.appendPathComponent("test")

dictionary.write(toFile:docPath.path,atomically:true)

print(docPath.path)

let kk=NSDictionary(contentsOf: docPath)

* 运行结果：



1. 从网上下载一张照片并保存到本地沙盒的Document的某文件中；

* 程序代码：

/\*

4. 从网上下载一张照片并保存到本地沙盒的Document的某文件中；

\*/

let imageUrl=URL(string: "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1538060490247&di=5c1426f416827e7515a18d1efca4413b&imgtype=0&src=http%3A%2F%2Fpic63.nipic.com%2Ffile%2F20150330%2F8993928\_082652755616\_2.jpg")!

let imageDate=try! Data(contentsOf: imageUrl)

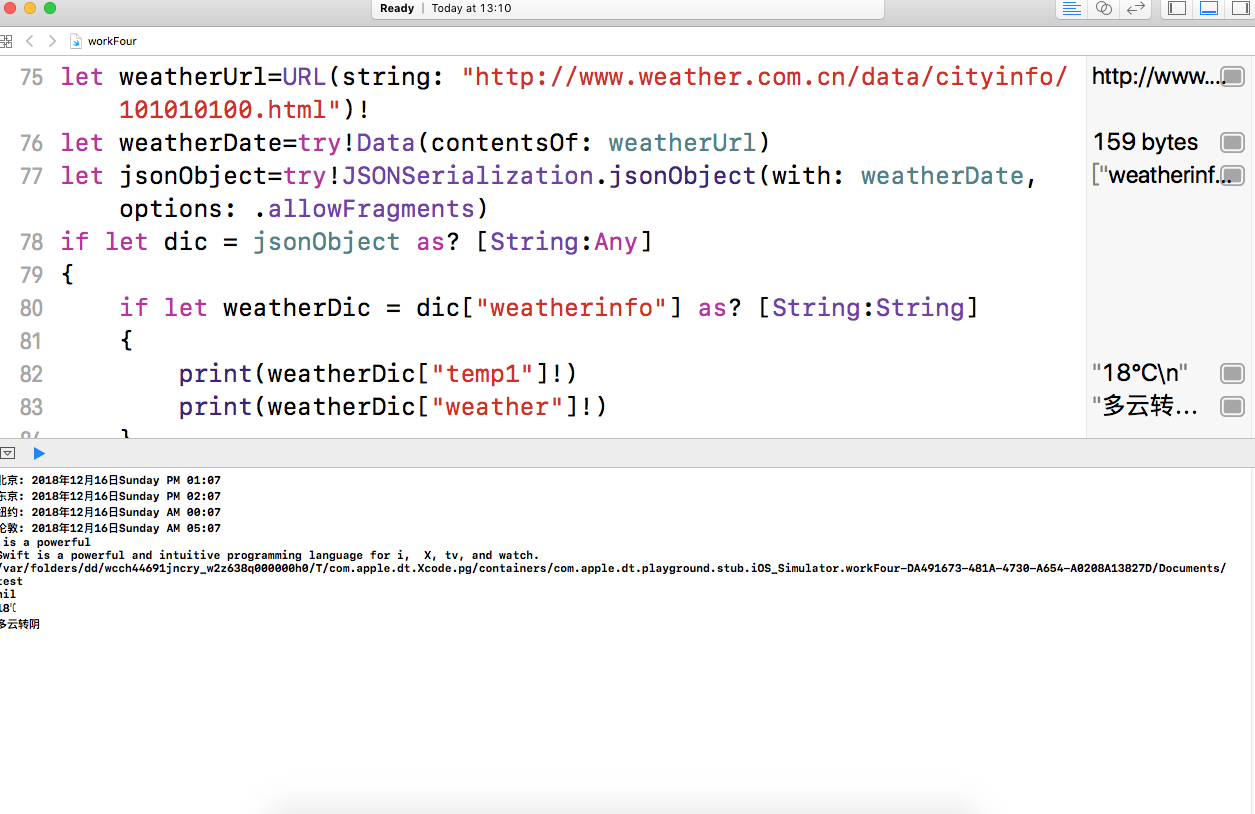
let image=UIImage(data:imageDate)

var imageLocalUrl=fileManager.urls(for: .documentDirectory, in:.userDomainMask).first!

imageLocalUrl.appendPathComponent("1.jpg")

try?imageDate.write(to: imageLocalUrl)

* 运行结果：



1. 从网上查找访问一个JSon接口文件，并采用JSONSerialization和字典对其进行简单解析；

* 程序代码：

/\*

5. 从网上查找访问一个JSon接口文件，并采用JSONSerialization和字典对其进行简单解析；

\*/

let weatherUrl=URL(string: "http://www.weather.com.cn/data/cityinfo/101010100.html")!

let weatherDate=try!Data(contentsOf: weatherUrl)

let jsonObject=try!JSONSerialization.jsonObject(with: weatherDate, options: .allowFragments)

if let dic = jsonObject as? [String:Any]

{

if let weatherDic = dic["weatherinfo"] as? [String:String]

{

print(weatherDic["temp1"]!)

print(weatherDic["weather"]!)

}

}

extension Int {

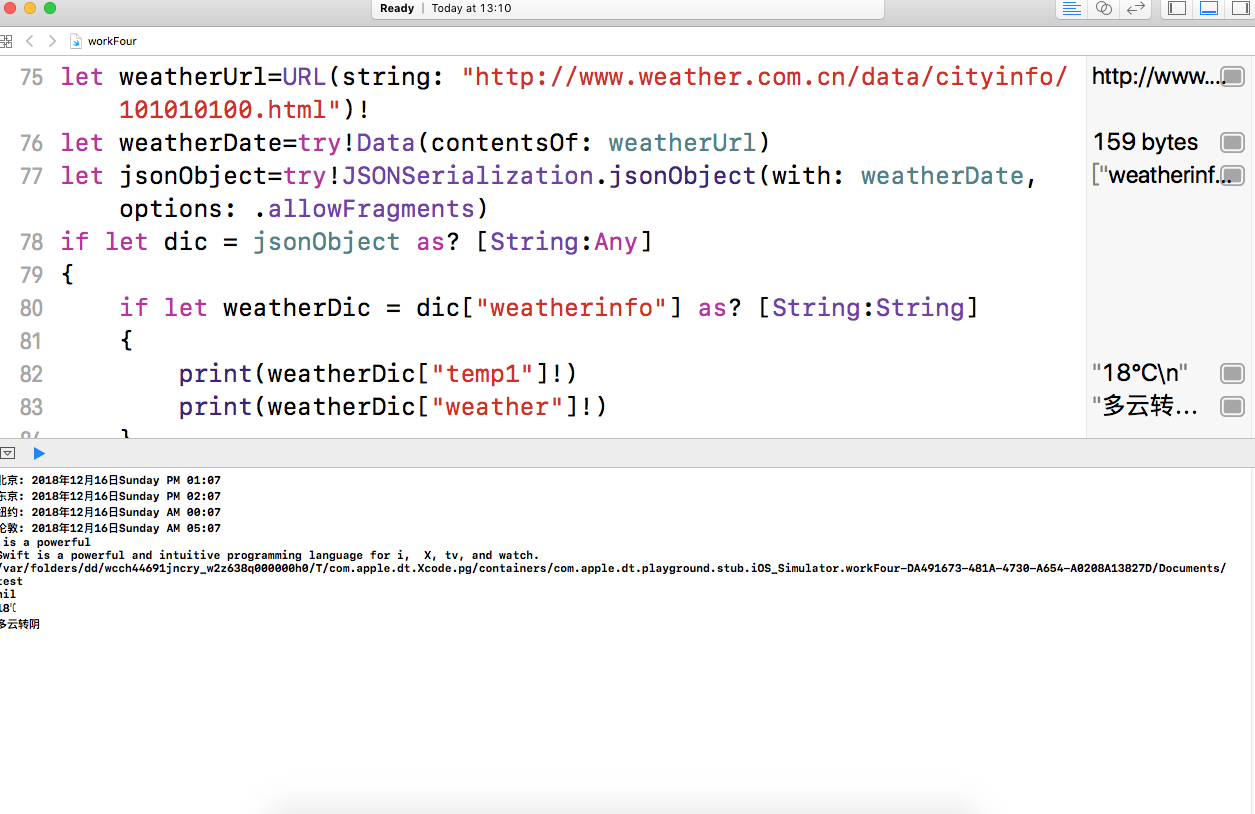
func sqrt() -> Double {

return Darwin.sqrt(Double(self))

}

}

* 运行结果：



1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

**Github地址:** [**https://github.com/jiangyutong/swiftWork/tree/master/代码**](https://github.com/jiangyutong/swiftWork/tree/master/代码)

这次实验主要是时间和沙盒。未来此次实验我还专门去了一下时间究竟是怎么划分的。

我觉得苹果很安全有一个原因就是它有沙盒，当我们的操作只能在沙盒中进行的时候，它当然就会安全许多。

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。