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：蒋星 学号：2016110418 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：

 var now = Date()

  var formmatter =  DateFormatter()

  formmatter.locale

  formmatter.dateFormat = "yyyy年MM月dd日 EEEE aaa hh:mm"

 print("北京时间："+formmatter.string(from: now))

  formmatter.timeZone = TimeZone.init(secondsFromGMT: 3600\*9)

  print("东京时间：" + formmatter.string(from: now))

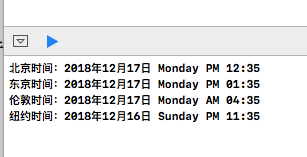
 formmatter.timeZone = TimeZone.init(secondsFromGMT: 3600\*0)

 print("伦敦时间：" + formmatter.string(from: now))

 formmatter.timeZone = TimeZone.init(secondsFromGMT: 3600 \* -5)

 print("纽约时间：" + formmatter.string(from: now))

* 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.

//截取字符串 6 ～20

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

let index\_1 = str.index(str.startIndex, offsetBy: 6)

let index\_2 = str.index(str.startIndex, offsetBy: 20)

let sub = str[index\_1...index\_2]

print(sub)

let re\_str = str.replacingOccurrences(of: "OS", with: "")

print(re\_str)

* 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.

var dicData:[String:String] = [String:String]()

let time\_beijing = formmatter.string(from: now)

formmatter.timeZone = TimeZone.init(secondsFromGMT: 3600 \* 9)

let time\_lundun = formmatter.string(from: now)

formmatter.timeZone = TimeZone.init(secondsFromGMT: 3600 \* 0)

let time\_dongjing = formmatter.string(from: now)

formmatter.timeZone = TimeZone.init(secondsFromGMT: 3600 \* -5)

let time\_newyu = formmatter.string(from: now)

dicData["Beijing time"] = time\_beijing //北京时间

dicData["London time"] = time\_lundun //伦敦时间

dicData["Tokyo time"] = time\_dongjing //东京时间

dicData["NewYork time"] = time\_newyu //纽约时间

dicData["str"] = str

print(dicData)

// 1、获得沙盒的根路径

let home = NSHomeDirectory() as NSString

print(home)

// 2、获得Documents路径，使用NSString对象的stringByAppendingPathComponent()方法拼接路径

let docPath = home.appendingPathComponent("Documents") as NSString;

// 3.获得文本文件路径

let filePath = docPath.appendingPathComponent("data.plist")

print(filePath)

let data\_store = dicData as NSDictionary

data\_store.write(toFile: filePath, atomically: true)

* 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：

//存储路径

//////// 4.

let url = URL(string:"https://goss.veer.com/creative/vcg/veer/800water/veer-145757956.jpg")

let data = try! Data(contentsOf:url!)

let img = UIImage(data:data)

func saveImage(currentImage: UIImage, persent: CGFloat, imageName: String){

if let imageData = UIImageJPEGRepresentation(currentImage, persent) as NSData? {

let fullPath = NSHomeDirectory().appending("/Documents/").appending(imageName)

imageData.write(toFile: fullPath, atomically: true)

print("fullPath=\(fullPath)") //打印存储路径

}

}

saveImage(currentImage: img!, persent: 1, imageName: "999.jpg")

* 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.

enum MyError:Error {

case Zero

case NotURL

}

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

throw MyError.NotURL

}

let jsondata = try! Data(contentsOf: weatherUrl)

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

guard let dic = json as? [String:Any] else{

throw MyError.NotURL

}

guard let weather = dic["weatherinfo"] as? [String:String] else{

throw MyError.NotURL

}

let temp1 = weather["temp1"]

let temp2 = weather["temp2"]

print(weather)

* 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/KikcerGoer/study-swift/tree/master/homework

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。