实验编号： 3 **四川师大《IOS》实验报告 2019** 年 **12** 月 **2** 日

### **计算机科学学院** 2017 级 4 班 实验名称： Date、String、文件、URL \_

姓名：\_李晋\_ 学号：\_\_2017110517\_\_\_ 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：

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(identifier: "zh\_cn")

    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: now)  //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)")  //输出北京时间

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

print("东京: \(tokyo)")  //输出东京时间

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

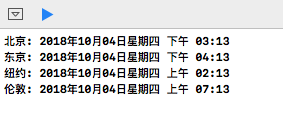
print("纽约: \(newYork)")  //输出纽约时间

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

print("伦敦: \(london)")  //输出伦敦时间

)

* 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：

var str1:String = "Swift is a powerful and intutive programming language for iOS,OS X,tvOS,and watch OS."

var str2=(str1 as NSString).substring(with: NSMakeRange(6,20))

print(str2)

var str3 = str1.replacingOccurrences(of: "OS", with: "")

print(str3)

* 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：

let fileManager = FileManager.default

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

    docPath.appendPathComponent("沙盒.txt")

    print(docPath.path)

    try str2.write(to:docPath,atomically:true,encoding:.utf8)

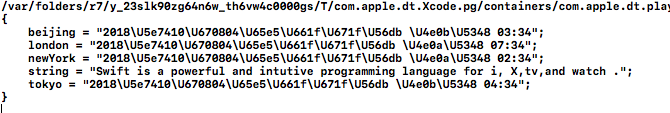
    let dic = ["beijing":beijing,"tokyo":tokyo,"newYork":newYork,"london":london,"string":str3] as AnyObject

    print(dic)

    dic.write(to : docPath,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：

var str4:URL?

do{

    let url = URL(string:"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1538649435720&di=d48cc90254747e46ff9b47afc0d0a838&imgtype=0&src=http%3A%2F%2Fimg2015.zdface.com%2F20171214%2Fe44e18258dfe540d6591c67ab44d15a1.jpg")

    let data = try Data(contentsOf:url!)

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

        docPath.appendPathComponent("沙盒.txt")

        str4 = docPath

        let picData = data as AnyObject

        picData.write(to : docPath,atomically:true)

    }

}catch{

}

* 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：

var str4:URL?

do{

    let url = URL(string : "http://www.weather.com.cn/adat/sk/101270101.html")

    let data = try Data(contentsOf: url!)

    let object = try JSONSerialization.jsonObject(with: data, options: .allowFragments)

    if var dic = object as? [String:Any] {

        print(dic["weatherinfo"])

        if let weather = dic["weatherinfo"] as? [String:Any] {

            print(weather["city"])

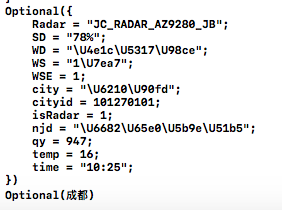
        }

    }

}catch{

}

* 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）

了解沙盒机制，每个应用程序都有自己独立的存储空间，应用程序只能对自己创建的文件系统读取文件，这个独立、封闭、安全的空间，它一般存放着程序包含的文件（即可执行文件）/图片/音频/视频及其他文件。

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。