实验编号：03 **四川师大《IOS高级开发技术》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： 作业三 \_

姓名：\_\_邹琳\_\_ 学号：\_2016110458\_\_\_\_ 指导老师：\_李贵洋\_ 实验成绩:\_\_\_\_\_

**实验\_三\_ \_\_\_\_\_作业三\_\_\_\_\_\_**

一．实验目的及要求

1、实验目的

1. 掌握Date、String的定义以及使用；
2. 掌握文件、URL的定义以及使用；

2、实验要求

1. 认真填写实验报告，要求附加部分运行界面和主要代码；
2. 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；

二. 实验内容

**作业1（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）

代码：

//: Playground - noun: a place where people can play

import UIKit

//第1题

let currentDate = Date()

var dateFormatter = DateFormatter()

dateFormatter.dateFormat = "yyyy年M月dd日E a HH:mm"

dateFormatter.locale = Locale(identifier: "zh\_CN")

let timeBeiJing = dateFormatter.string(from:currentDate)

print("beijing："+timeBeiJing)

dateFormatter.timeZone = TimeZone(abbreviation:"UTC+9:00")

let timeTokyo = dateFormatter.string(from: currentDate)

print("dongjing："+timeTokyo)

dateFormatter.timeZone = TimeZone(secondsFromGMT:-4\*3600)

let timeNY = dateFormatter.string(from: currentDate)

print("niuyue："+timeNY)

dateFormatter.timeZone = TimeZone(secondsFromGMT:1\*3600)

let timeLondon = dateFormatter.string(from: currentDate)

print("london："+timeLondon)

//第2题

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

let index1 = str.index(str.startIndex, offsetBy: 6)

let index2 = str.index(str.startIndex, offsetBy: 26)

let str1 = str[index1..<index2]

print(str1)

let str2 = str.replacingOccurrences(of: "OS", with: "")

print(str2)

//第3题

//var dictionary = ["beijingtime":timeBeiJing,"dongjingtime":timeTokyo,"niuyuetime":timeNY,"londontime":timeLondon,"str":str]

//let filename = FileManager.default

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

//url.appendPathComponent("test3.txt")

//print(url)

//try? dictionary.write(to: url, atomically: true)

let dic = ["date": ["beijing": timeBeiJing, "tokyo": timeTokyo, "newYork": timeNY, "london": timeLondon], "string": str] as AnyObject

let fileManager = FileManager.default

print(fileManager.currentDirectoryPath)

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

path.append("/test3.txt") //在文件夹路径下增加一个test.txt

print(dic.write(toFile: path, atomically: true)) //新建上面指定的文件，并将数据写入(输出true)

}

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

//urls.appendPathComponent("test3-3.txt")

//print(dic.write(toFile:urls,atomically:true,encoding:.utf8)

//第4题

let image = try Data(contentsOf: URL(string: "http://t2.hddhhn.com/uploads/tu/201707/200/2.jpg")!) //通过指定的url获取图片，并转换为二进制数据

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

url.appendPathComponent("image.png")

try image.write(to: url) //将转换后的二进制数据存储为png图片

}

//第5题

enum MyError:Error {

case Zero

case NotURL

}

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

// throw MyError.NotURL

//}

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

//let strs = try String(contentsOf: url)

//print(strs) //显示json数据内容

//let data = try Data(contentsOf: url) //将json转换为二进制数据

//let json = try JSONSerialization.jsonObject(with: data, options: .allowFragments) //序列化json

////解析json数据

//if let dics = json as? [String: Any] {

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

// let city = weather["city"]!

// let temp = weather["temp"]!

// let wd = weather["WD"]!

// let ws = weather["WS"]!

// print("城市: \(city), 温度: \(temp), 风向: \(wd), 风力: \(ws)")

// //输出: 城市: 西安, 温度: 20, 风向: 西南风, 风力: 1级

// }

//}

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

throw MyError.NotURL

}

let jsondata = try! Data(contentsOf: postUrl) //将json转换为二进制数据

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

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

throw MyError.NotURL

}

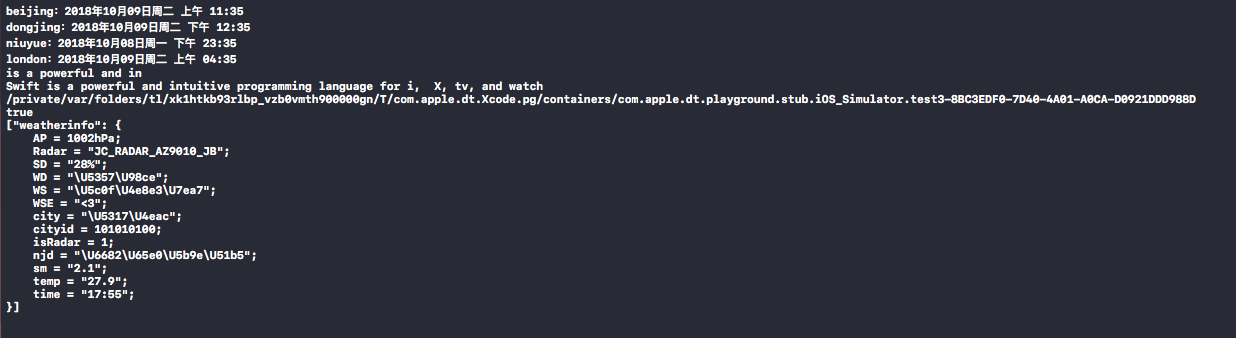
print(dics)

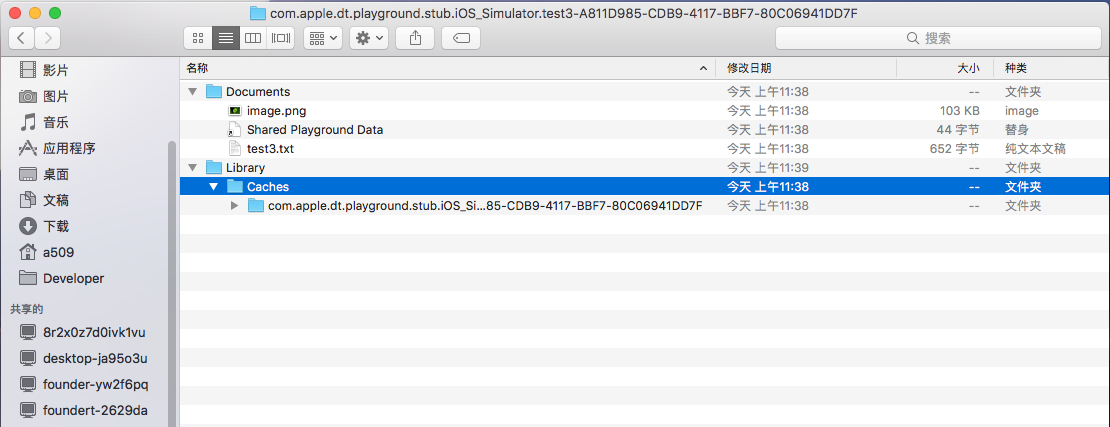
guard let informationInfo = dics["message"] as? [String:String] else {

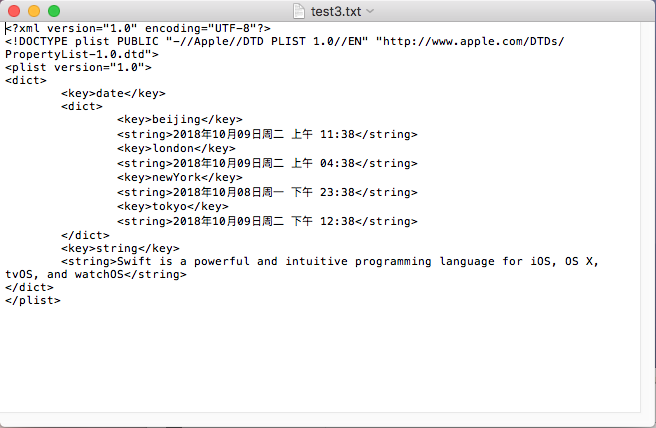
throw MyError.NotURL

}

print(informationInfo)









1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

本次实验主要考察了对时间的获取，对String对象的操作，对Json文件的解析等文件方面的操作，对沙盒的操作充分体现了IOS系统的安全性，展示了其系统的优势。也体现了swift开发的便捷性和优势。

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。