实验编号： 1 **四川师大《IOS》实验报告 2018** 年 **9** 月 **5** 日

### **计算机科学学院** 级 4 班 实验名称： Swift类和子类 \_

姓名：蒋星 学号：2016110418 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验\_一\_ \_\_\_\_\_\_** Swift类和子类**\_\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 用 Swift语言编写类和子类
3. 掌握类和子类的定义及使用
4. 掌握git的基本使用命令
5. 实验要求
6. 认真填写实验报告，要求附加部分运行界面和主要代码；
7. 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；
8. 实验内容

* 作业1:（重点：闭包）

1. 找出从1-10000的所有质数，将找出的质数存入可变数组，对数组进行正反排序并输出结果(采用5种以上的方法实现)；
2. 掌握版本控制git的基本使用
   1. git init 初始化版本库
   2. git add . 添加当前目录下的所有文件到暂存区
   3. git commit -m 提交暂存区的文件到版本库

* 作业2:（枚举、类、派生）

1. 实现Person类：
   1. 要求具有firstName, lastName，age，gender等存储属性,fullName计算属性；其中gender是枚举类型（male，female）；
   2. 具有指定构造函数和便利构造函数；
   3. 两个Person实例对象可以用==和!=进行比较；
   4. Person实例可以直接用print输出；
2. 从Person分别派生Teacher类和Student类：
   1. Teacher类增加属性title，实例可以直接用print输出；
   2. Student类增加属性stuNo，实例可以直接用print输出；
3. 分别构造多个Person、Teacher和Student对象，并将这些对象存入同一个数组中；
4. 对数组执行以下要求：
   1. 分别统计Person、Teacher和Student对象的个数并放入一字典中，统计完后输出字典内容；
   2. 对数组按以下要求排序并输出：age、fullName、gender+age；
5. 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）

* 作业1:（重点：闭包）

1. 找出从1-10000的所有质数，将找出的质数存入可变数组，对数组进行正反排序并输出结果(采用5种以上的方法实现)；

* 程序代码：

import UIKit

#if os(Linux)

import Glibc

#else

import Darwin

#endif

var emptyArray:Array = [Int]()

emptyArray.append(1)

emptyArray.append(2)

emptyArray.append(3)

var index = 0;

for count in 4...100 {

let d\_count:Double = Double(count);

let result:Double = sqrt(d\_count);

let boundary = Int(result);

var ishave:Bool = false

for var i in 2...boundary{

let d\_re = Double(count)/Double(i);

let i\_re = count / i;

if d\_re == Double(i\_re) {

ishave = true

break

}

}

if !ishave {

emptyArray.append(count)

index = index + 1

}

}

emptyArray.sort();

emptyArray.reverse();

emptyArray.sort(by:{$0 > $1 })

func sortself (one:Int,two:Int) -> Bool{

return one > two

}

emptyArray.sort(by:sortself)

emptyArray.sort(by:{(one:Int,two:Int) in return one > two })

print(emptyArray)

emptyArray.sort(by: > )

var array = [10,4,2,11,65]

let temp = array.reduce(

(max:Int.min,min:Int.max,sum:0)

)

{ (temp,i) in

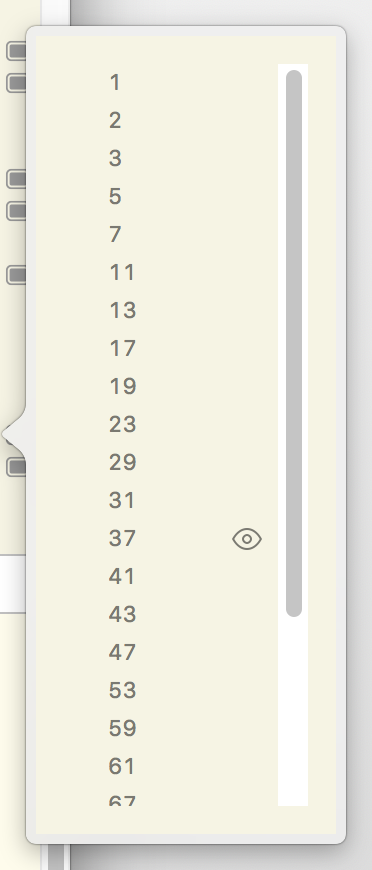
print(temp)

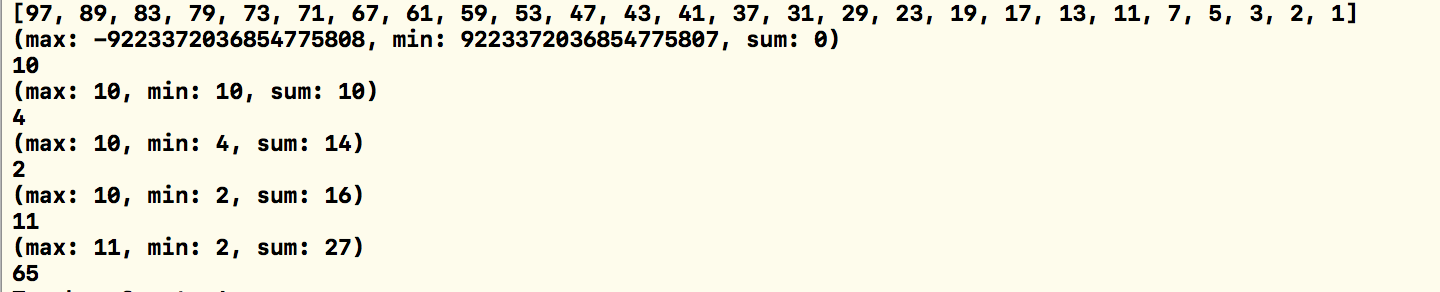
print(i)

return (max(temp.max,i),min(temp.min,i),temp.sum+i)

}

* 运行结果：





1. 掌握版本控制git的基本使用
   1. git init 初始化版本库
   2. git add . 添加当前目录下的所有文件到暂存区
   3. git commit -m 提交暂存区的文件到版本库

* 程序代码：

Github 地址为证 已经学习完成：

https://github.com/kickgod/Md-Linux/tree/master/Git

* 运行结果：
* 作业2:（枚举、类、派生）

1. 实现Person类：
   1. 要求具有firstName, lastName，age，gender等存储属性,fullName计算属性；其中gender是枚举类型（male，female）；
   2. 具有指定构造函数和便利构造函数；
   3. 两个Person实例对象可以用==和!=进行比较；
   4. Person实例可以直接用print输出；

* 程序代码：

enum department:Int{

case sicnu\_jisuanjikexuexueyuan = 1

case sicnu\_gongxueyuan = 2

case sicnu\_shangxueyuan = 4

case sicnu\_wenxueyuan = 8

}

protocol SchoolProtocal{

var department:department{ get set }

func lendBook(data:String) ->()

}

enum Gender:Int{

case male = 1

case female = 4

}

class Person: CustomStringConvertible{

var firstName:String

var lastName:String

var age:Int

var gender:Gender

init(firstName:String,lastName:String,age:Int,gender:Gender) {

self.age = age

self.lastName = lastName

self.firstName = firstName

self.gender = gender

}

var fullName:String{

get{

return firstName + " " + lastName

}

}

func run(){

print("Person \(fullName) is running")

}

convenience init(firstName:String){

self.init(firstName: firstName,lastName:"king",age:18,gender: Gender.male)

}

static func == (one:Person,two:Person)-> Bool{

if(one.age == two.age) && (one.firstName == two.firstName) && (one.lastName == two.lastName)

&& (one.gender == two.gender){

return true;

}else{

return false;

}

}

static func != (one:Person,two:Person)-> Bool{

if(one.age == two.age) && (one.firstName == two.firstName) && (one.lastName == two.lastName)

&& (one.gender == two.gender){

return false;

}else{

return true;

}

}

var description:String {

return " name: \(firstName) \(lastName) age:\(age) sex:\(gender) "

}

}

1. 从Person分别派生Teacher类和Student类：
   1. Teacher类增加属性title，实例可以直接用print输出；
   2. Student类增加属性stuNo，实例可以直接用print输出；
2. 分别构造多个Person、Teacher和Student对象，并将这些对象存入同一个数组中；
3. 对数组执行以下要求：
   1. 分别统计Person、Teacher和Student对象的个数并放入一字典中，统计完后输出字典内容；
   2. 对数组按以下要求排序并输出：age、fullName、gender+age；

* 程序代码：

var p = Person(firstName:"kicker0",lastName:"king",age:21,gender:Gender.male)

var p0 = Person(firstName:"kicker10",lastName:"king",age:21,gender:Gender.male)

var p1 = Teacher(firstName:"kicker1",lastName:"king",age:21,gender:Gender.male,title:"banzhuren0")

var p3 = Teacher(firstName:"kicker2",lastName:"king",age:21,gender:Gender.male,title:"banzhuren1")

var p4 = Teacher(firstName:"kicker3",lastName:"king",age:21,gender:Gender.male,title:"banzhuren2")

var p5 = Teacher(firstName:"kicker4",lastName:"king",age:21,gender:Gender.male,title:"banzhuren3")

var p2 = Student(firstName:"kicker5",lastName:"king",age:21,gender:Gender.male,stuNo:"2016110418")

var plist:[Person] = [Person]()

plist.append(p)

plist.append(p0)

plist.append(p1)

plist.append(p2)

plist.append(p3)

plist.append(p4)

plist.append(p5)

var tcount:Int = plist.filter({ $0 is Teacher}).count

var scount:Int = plist.filter({ $0 is Student}).count

var pcount:Int = plist.count - tcount - scount

print("Teacher Count: \(tcount)")

print("Student Count: \(scount)")

print("Person Count: \(pcount)")

var dic:[String:Int] = [String:Int]()

dic["Teacher Count"] = 0

dic["Student Count"] = 0

dic["Person Count"] = 0

plist.map(){

one in

if(one is Teacher){

dic["Teacher Count"] = dic["Teacher Count"]! + 1

}

if(one is Student){

dic["Student Count"] = dic["Student Count"]! + 1

}

dic["Person Count"] = plist.count - dic["Student Count"]! - dic["Student Count"]!

}

print(dic)

plist.forEach({

$0.run()

if $0 is Teacher{

($0 as! Teacher).lendBook(data: "wo shi laoshi")

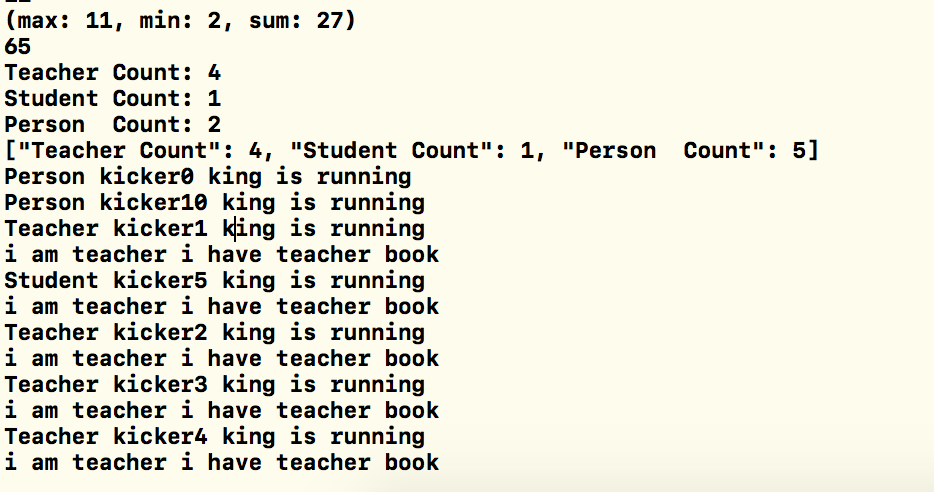
} else if $0 is Student{

($0 as! Student).lendBook(data: "wo shi xuesheng")

}

})

* 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: 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。