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：\_樊琳\_ 学号：\_2016110408\_ 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：

var count=0

var ispreimnum = true

var preimnumber=[2]

for item in 3...1000

{

for j in 2..<item

{

if item % j == 0

{

ispreimnum = false

break

}

}

if ispreimnum{

count += 1

preimnumber.append(item)

}

ispreimnum = true

}

print("数量总计是\(count)个素数")

print(preimnumber)

///开始排序，从小到大

///1

func descending(x: Int, y: Int) -> Bool {

return x > y

}

preimnumber.sort(by: descending)

///2

preimnumber.sort { (x: Int, y: Int) -> Bool in

return x > y

}

///3

preimnumber.sort { (x, y) -> Bool in

return x > y

}

///4

preimnumber.sort { (x, y) in

return x > y

}

///5

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

///6

preimnumber.sort { $0 > $1 }

///7

preimnumber.sort(by: >)

print(preimnumber)

///开始排序，从大到小

///1

func descending2(x: Int, y: Int) -> Bool {

return x < y

}

preimnumber.sort(by: descending2)

///2

preimnumber.sort { (x: Int, y: Int) -> Bool in

return x < y

}

///3

preimnumber.sort { (x, y) -> Bool in

return x < y

}

///4

preimnumber.sort { (x, y) in

return x < y

}

///5

preimnumber.sort(by: { return $0 < $1 })

///6

preimnumber.sort { $0 < $1 }

///7

preimnumber.sort(by: <)

print(preimnumber)

* 运行结果：

数量总计是167个素数

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]

[997, 991, 983, 977, 971, 967, 953, 947, 941, 937, 929, 919, 911, 907, 887, 883, 881, 877, 863, 859, 857, 853, 839, 829, 827, 823, 821, 811, 809, 797, 787, 773, 769, 761, 757, 751, 743, 739, 733, 727, 719, 709, 701, 691, 683, 677, 673, 661, 659, 653, 647, 643, 641, 631, 619, 617, 613, 607, 601, 599, 593, 587, 577, 571, 569, 563, 557, 547, 541, 523, 521, 509, 503, 499, 491, 487, 479, 467, 463, 461, 457, 449, 443, 439, 433, 431, 421, 419, 409, 401, 397, 389, 383, 379, 373, 367, 359, 353, 349, 347, 337, 331, 317, 313, 311, 307, 293, 283, 281, 277, 271, 269, 263, 257, 251, 241, 239, 233, 229, 227, 223, 211, 199, 197, 193, 191, 181, 179, 173, 167, 163, 157, 151, 149, 139, 137, 131, 127, 113, 109, 107, 103, 101, 97, 89, 83, 79, 73, 71, 67, 61, 59, 53, 47, 43, 41, 37, 31, 29, 23, 19, 17, 13, 11, 7, 5, 3, 2]

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]

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

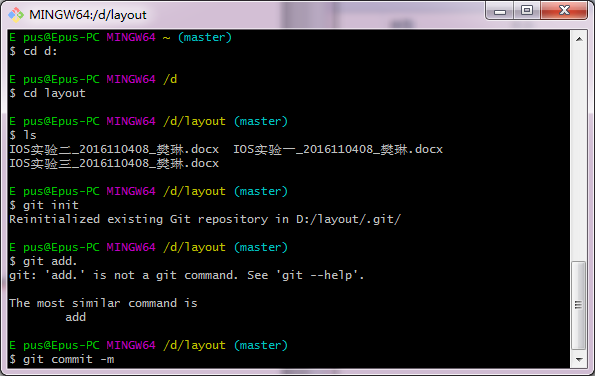
* 程序代码：

git init

git add .

git commit -m

* 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 Gender{

case male

case female

}

///person类

class Person {

var firstName: String

var lastName: String

var age: Int

var gender: Gender

var fullName: String{ return firstName + lastName }

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

self.firstName = firstName

self.lastName = lastName

self.age = age

self.gender = gender

}

convenience init(firstName: String){

self.init(firstName: firstName, lastName: "", age: 43, gender: Gender.male)

}

func description() -> String{

return "name:\(fullName) age:\(age) gender:\(gender)"

}

static func ==(x:Person,y:Person) ->Bool{

return x.fullName==y.fullName&&x.age==y.age&&x.gender==y.gender

}

static func != (x:Person,y:Person) ->Bool

{

return !(x == y)

}

}

///teacher类

class Teacher: Person{

var title : String

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

self.title = title

super.init(firstName: firstName, lastName: lastName, age: age, gender: gender)

}

override func description() -> String{

return "title:\(title) name:\(fullName) age:\(age) gender:\(gender)"

}

}

///student类

class Student: Person{

var stuNo: String

init(stuNo:String, firstName: String,

lastName: String, age: Int, gender: Gender){

self.stuNo = stuNo

super.init(firstName: firstName, lastName: lastName, age: age, gender: gender)

}

override func description() -> String{

return "stuNo:\(stuNo) name:\(fullName) age:\(age) gender:\(gender) "

}

}

var person1 = Person(firstName: "Li", lastName: " guiyang", age: 18, gender: Gender.female)

print(person1.description())

var person2 = Person(firstName: "gao")

print(person2.description())

print("person1==person2:")

print(person1==person2)

print("person1!=person2:")

print(person1 != person2)

var teacher = Teacher(title: "Math",firstName: "1", lastName: "2", age: 23, gender: Gender.male)

print(teacher.description())

var student = Student(stuNo:"2016110001", firstName: "han",

lastName: "meimei", age: 32, gender: Gender.female)

print(student.description())

var teacher1 = Teacher(title:"Chinese",firstName:"jiang ",lastName:"hong",age:19,gender:Gender.female)

var student1 = Student(stuNo: "2016110002",firstName: "Wang ",lastName: "qian", age: 20,gender: Gender.male)

//对象数组

var arry = [person1,person2,teacher,student,teacher1,student1]

for i in 0...2

{

let temp = Student(stuNo:"2016110003",firstName: "wang",lastName:"san",age:20,gender:Gender.male)

arry.append(temp)

}

//声明一个字典并对其初始化

var countDate = ["Person": 0,"Teacher":0,"Student":0]

//判断各种类的个数

for item in arry {

if item is Teacher { //是否是Teacher类

countDate["Teacher"]! += 1

} else if item is Student { //是否是Student

countDate["Student"]! += 1

} else { //Person类

countDate["Person"]! += 1

}

}

//输出字典内容

print(countDate)

//对数组按以下要求排序并输出：age、fullName、gender+age

//按照age

arry.sort{return $0.age>$1.age}

print("按照age排序")

for i in 0...arry.count-1

{

print(arry[i].description())

}

//按照fullName

arry.sort{return $0.fullName>$1.fullName}

print("按照fullName排序")

for i in 0...arry.count-1

{

print(arry[i].description())

}

//按照gender+age

func compareGender(x:Gender,y:Gender) ->Bool

{

return x.hashValue > y.hashValue

}

func compare(x:Person,y:Person) ->Bool {

return compareGender(x:x.gender,y:y.gender) && x.age>y.age

}

arry.sort(by:compare)

print("按照gender+age排序")

for i in 0...arry.count-1

{

print(arry[i].description())

}

* 运行结果：

name:Li guiyang age:18 gender:female

name:gao age:43 gender:male

person1==person2:

false

person1!=person2:

true

title:Math name:12 age:23 gender:male

stuNo:2016110001 name:hanmeimei age:32 gender:female

["Person": 2, "Teacher": 2, "Student": 5]

按照age排序

name:gao age:43 gender:male

stuNo:2016110001 name:hanmeimei age:32 gender:female

title:Math name:12 age:23 gender:male

stuNo:2016110002 name:Wang qian age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

title:Chinese name:jiang hong age:19 gender:female

name:Li guiyang age:18 gender:female

按照fullName排序

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110002 name:Wang qian age:20 gender:male

name:Li guiyang age:18 gender:female

title:Chinese name:jiang hong age:19 gender:female

stuNo:2016110001 name:hanmeimei age:32 gender:female

name:gao age:43 gender:male

title:Math name:12 age:23 gender:male

按照gender+age排序

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110002 name:Wang qian age:20 gender:male

name:Li guiyang age:18 gender:female

title:Chinese name:jiang hong age:19 gender:female

stuNo:2016110001 name:hanmeimei age:32 gender:female

name:gao age:43 gender:male

title:Math name:12 age:23 gender:male

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：

enum Gender{

case male

case female

}

///person类

class Person {

var firstName: String

var lastName: String

var age: Int

var gender: Gender

var fullName: String{ return firstName + lastName }

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

self.firstName = firstName

self.lastName = lastName

self.age = age

self.gender = gender

}

convenience init(firstName: String){

self.init(firstName: firstName, lastName: "", age: 43, gender: Gender.male)

}

func description() -> String{

return "name:\(fullName) age:\(age) gender:\(gender)"

}

static func ==(x:Person,y:Person) ->Bool{

return x.fullName==y.fullName&&x.age==y.age&&x.gender==y.gender

}

static func != (x:Person,y:Person) ->Bool

{

return !(x == y)

}

}

///teacher类

class Teacher: Person{

var title : String

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

self.title = title

super.init(firstName: firstName, lastName: lastName, age: age, gender: gender)

}

override func description() -> String{

return "title:\(title) name:\(fullName) age:\(age) gender:\(gender)"

}

}

///student类

class Student: Person{

var stuNo: String

init(stuNo:String, firstName: String,

lastName: String, age: Int, gender: Gender){

self.stuNo = stuNo

super.init(firstName: firstName, lastName: lastName, age: age, gender: gender)

}

override func description() -> String{

return "stuNo:\(stuNo) name:\(fullName) age:\(age) gender:\(gender) "

}

}

var person1 = Person(firstName: "Li", lastName: " guiyang", age: 18, gender: Gender.female)

print(person1.description())

var person2 = Person(firstName: "gao")

print(person2.description())

print("person1==person2:")

print(person1==person2)

print("person1!=person2:")

print(person1 != person2)

var teacher = Teacher(title: "Math",firstName: "1", lastName: "2", age: 23, gender: Gender.male)

print(teacher.description())

var student = Student(stuNo:"2016110001", firstName: "han",

lastName: "meimei", age: 32, gender: Gender.female)

print(student.description())

var teacher1 = Teacher(title:"Chinese",firstName:"jiang ",lastName:"hong",age:19,gender:Gender.female)

var student1 = Student(stuNo: "2016110002",firstName: "Wang ",lastName: "qian", age: 20,gender: Gender.male)

//对象数组

var arry = [person1,person2,teacher,student,teacher1,student1]

for i in 0...2

{

let temp = Student(stuNo:"2016110003",firstName: "wang",lastName:"san",age:20,gender:Gender.male)

arry.append(temp)

}

//声明一个字典并对其初始化

var countDate = ["Person": 0,"Teacher":0,"Student":0]

//判断各种类的个数

for item in arry {

if item is Teacher { //是否是Teacher类

countDate["Teacher"]! += 1

} else if item is Student { //是否是Student

countDate["Student"]! += 1

} else { //Person类

countDate["Person"]! += 1

}

}

//输出字典内容

print(countDate)

//对数组按以下要求排序并输出：age、fullName、gender+age

//按照age

arry.sort{return $0.age>$1.age}

print("按照age排序")

for i in 0...arry.count-1

{

print(arry[i].description())

}

//按照fullName

arry.sort{return $0.fullName>$1.fullName}

print("按照fullName排序")

for i in 0...arry.count-1

{

print(arry[i].description())

}

//按照gender+age

func compareGender(x:Gender,y:Gender) ->Bool

{

return x.hashValue > y.hashValue

}

func compare(x:Person,y:Person) ->Bool {

return compareGender(x:x.gender,y:y.gender) && x.age>y.age

}

arry.sort(by:compare)

print("按照gender+age排序")

for i in 0...arry.count-1

{

print(arry[i].description())

}

* 运行结果：

name:Li guiyang age:18 gender:female

name:gao age:43 gender:male

person1==person2:

false

person1!=person2:

true

title:Math name:12 age:23 gender:male

stuNo:2016110001 name:hanmeimei age:32 gender:female

["Person": 2, "Teacher": 2, "Student": 5]

按照age排序

name:gao age:43 gender:male

stuNo:2016110001 name:hanmeimei age:32 gender:female

title:Math name:12 age:23 gender:male

stuNo:2016110002 name:Wang qian age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

title:Chinese name:jiang hong age:19 gender:female

name:Li guiyang age:18 gender:female

按照fullName排序

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110002 name:Wang qian age:20 gender:male

name:Li guiyang age:18 gender:female

title:Chinese name:jiang hong age:19 gender:female

stuNo:2016110001 name:hanmeimei age:32 gender:female

name:gao age:43 gender:male

title:Math name:12 age:23 gender:male

按照gender+age排序

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110003 name:wangsan age:20 gender:male

stuNo:2016110002 name:Wang qian age:20 gender:male

name:Li guiyang age:18 gender:female

title:Chinese name:jiang hong age:19 gender:female

stuNo:2016110001 name:hanmeimei age:32 gender:female

name:gao age:43 gender:male

title:Math name:12 age:23 gender:male

1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

经过本次实验,我大致熟练的掌握闭包,枚举,派生,类的内容,达到了实验的要求.

在做实验的过程中,还涉及到字典,数组等的运用,巩固了前面所学的关于swift的基础知识.

这次实验的难点在于闭包的内容,这是一个全新的知识.闭包运用灵活,需要个非常熟练才能运用的好.关于类和继承这些,是我们在学Java语言的时候就涉及到的,不算很难理解.

要多练习闭包.

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。