实验编号： 2 **四川师大《IOS》实验报告 2018** 年 **9** 月 **12** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 闭包、扩展、泛型、协议 \_

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

**实验\_二\_ \_\_\_\_\_\_**闭包、扩展、泛型、协议**\_\_\_\_\_\_\_\_**

1. 实验目的
2. 掌握闭包、扩展、泛型的定义;
3. 掌握排序等采用闭包方法的实现；
4. 掌握类的派生、协议；
5. 掌握版本控制git的进阶使用；
6. 实验要求
7. 认真填写实验报告，要求附加部分运行界面和主要代码；
8. 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；
9. 实验内容

* 作业1（闭包、扩展、泛型）：

1. 给定一个Dictionary，Dictionary包含key值name和age，用map函数返回age字符串数组;
2. 给定一个String数组，用filter函数选出能被转成Int的字符串
3. 用reduce函数把String数组中元素连接成一个字符串，以逗号分隔
4. 用 reduce 方法一次求出整数数组的最大值、最小值、总数和
5. 新建一个函数数组，函数数组里面保存了不同函数类型的函数，要求从数组里找出参数为一个整数，返回值为一个整数的所有函数；
6. 扩展Int，增加sqrt方法，可以计算Int的Sqrt值并返回浮点数，进行验证；
7. 实现一个支持泛型的函数，该函数接受任意个变量并返回最大和最小值，分别传入整数值、浮点数值、字符串进行验证。
8. 掌握版本控制git的进阶使用
   1. git reset 恢复到之前修改的版本；
   2. git log 看提交记录；
   3. git branch 新建分支；
   4. git checkout 切换分支；
   5. git branch -d 删除分支；

* 作业2:（枚举、类、派生、协议）(红色字体为新增内容)

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；
   5. Person增加run方法(方法里面直接print输出Person XXX is running;
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. Teacher和Student重载run方法(方法里面直接print输出Teacher XXX is running和Student XXX is running)
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. 新建一个协议SchoolProtocol，协议包括一个department属性(Enum，自己实现enum的定义)和lendBook方法（随便写点内容，能区隔即可）；
5. 修改Teacher和Student，让这两个类实现该协议；
6. 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；
   3. 对数组进行穷举，调用每个对象的run方法，同时调用满足协议SchoolProtocol对象的lendBook方法；
7. 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（闭包、扩展、泛型）：

1. 给定一个Dictionary，Dictionary包含key值name和age，用map函数返回age字符串数组;
2. 给定一个String数组，用filter函数选出能被转成Int的字符串
3. 用reduce函数把String数组中元素连接成一个字符串，以逗号分隔
4. 用 reduce 方法一次求出整数数组的最大值、最小值、总数和
5. 新建一个函数数组，函数数组里面保存了不同函数类型的函数，要求从数组里找出参数为一个整数，返回值为一个整数的所有函数；
6. 扩展Int，增加sqrt方法，可以计算Int的Sqrt值并返回浮点数，进行验证；
7. 实现一个支持泛型的函数，该函数接受任意个变量并返回最大和最小值，分别传入整数值、浮点数值、字符串进行验证。
8. 掌握版本控制git的进阶使用
   1. git reset 恢复到之前修改的版本；
   2. git log 看提交记录；
   3. git branch 新建分支；
   4. git checkout 切换分支；
   5. git branch -d 删除分支；

* 程序代码：

1.(1)

var Dictionary:[String:String] = ["a":"12","b":"13","c":"56","d":"75","e":"42"]

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

ages = Dictionary.map({$1})

print(ages)

1.(2)

var contoin

= ["asdas","asd23","5421","12","456.442","0","asd","-984"]

var result = contoin.filter( {Int($0) != nil} )

print(result)

1.(3)

var contoin = ["asdas","asd23","5421","12","456.442","0","asd","-984"]

var result = contoin.reduce("",{

if $0 == ""{

return $0 + $1

}else{

return $0 + "," + $1

}

})

1.(4)

var vals = [1,50,-90,-50,555,23]

var result\_val = vals.reduce((max:Int.min,min:Int.max,sum:0),{

(result,the\_val\_of\_vals\_input)

in

return (max(result.0, the\_val\_of\_vals\_input),min(result.1, the\_val\_of\_vals\_input),result.2 + the\_val\_of\_vals\_input)

})

print(result\_val)

/\*字典\*/

var vals = [1,50,-90,-50,555,23]

var result\_dic = ["max":Int.min,"min":Int.max,"sum":0]

var result\_val = vals.reduce(result\_dic,{

(result,val\_ok)

in

return ["max":max(result["max"]!,val\_ok),"min":min(result["min"]!, val\_ok),"sum":result["sum"]! + val\_ok];

})

print(result\_val)

1.(5)

func getvals(val:Double){

print("你真是的够了")

}

func getin(val:Int) ->Int{

return val \* 20

}

var funcs = [getvals,getin] as [Any]

//判断类型

var result\_func = funcs.filter({ $0 is (Int) ->Int }).map({ $0 as! (Int) ->Int })

1.(6)

extension Int{

func sqrt()->Double{

return Darwin.sqrt(Double(self))

}

}

var val:Int = 4

print(val.sqrt())

1.(7)

func get\_min\_max<T:Comparable>(Elements:T...)->(max:T,min:T){

return (Elements.max()!,Elements.min()!)

}

var rsl = get\_min\_max(Elements:56,56,918,12,23,456)

print(rsl)

1.(8)

已经学会

* 运行结果：
* 作业2:（枚举、类、派生、协议）(红色字体为新增内容)

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；
   5. Person增加run方法(方法里面直接print输出Person XXX is running;

* 程序代码：

func run(){

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

}

* 运行结果：

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；
   3. Teacher和Student重载run方法(方法里面直接print输出Teacher XXX is running和Student XXX is running)

* 程序代码：

override func run() {

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

}

override func run() {

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

}

* 运行结果：

1. 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；
2. 新建一个协议SchoolProtocol，协议包括一个department属性(Enum，自己实现enum的定义)和lendBook方法（随便写点内容，能区隔即可）；
3. 修改Teacher和Student，让这两个类实现该协议；
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；
   3. 对数组进行穷举，调用每个对象的run方法，同时调用满足协议SchoolProtocol对象的lendBook方法；

* 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) ->()

}

class Teacher:Person,SchoolProtocal{

var title:String

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

self.title = title

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

}

override var description:String{

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

}

override func run() {

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

}

var department:department{ get{ return department } set{ department = newValue} }

func lendBook(data:String) {

print("i am teacher i have teacher book")

}

}

class Student:Person,SchoolProtocal{

var stuNo:String

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

self.stuNo = stuNo

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

}

override var description:String{

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

}

override func run() {

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

}

var department:department{ get{ return department } set{ department = newValue} }

func lendBook(data:String) {

print("i am teacher i have teacher book")

}

}

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)

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

无敌是多么 多么寂寞

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。