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

**计算机科学学院** 2016级4班 实验名称： 作业一 \_

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

**实验\_一\_ \_\_\_\_\_作业一\_\_\_\_\_**

1. 实验目的及要求

1.用 Swift语言编写类和子类

2.掌握类和子类的定义及使用

3.掌握git的基本使用命令

1. 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、协议）

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：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)；

实验代码：

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

import UIKit

var primes = [Int]()

for i in 2...10000{

var isprime = true

for j in 2..<i{

if i%j == 0 {

isprime = false

}

}

if isprime{

primes.append(i)

}

}

//print(primes)

////对数组进行正反排序并输出所有结果

////方法1：

////primes.reverse()

////print(primes)

////方法2：

//func onSort(s1:Int,s2:Int)->Bool{

// return s1>s2

//}

//primes.sort(by:onSort)

//print(primes)

//

////方法3：

//primes.sort(by:{(s1:Int,s2:Int)->Bool in return s1>s2})

//方法4

//primes.sort(by:{(s1,s2) in return s1>s2})

//方法5

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

//方法6

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

//方法7

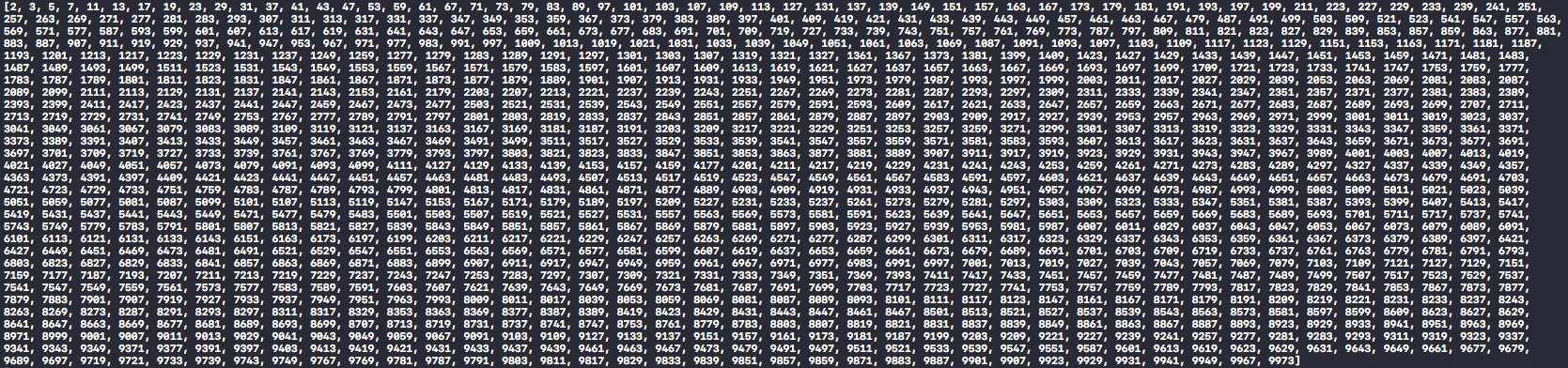
primes.sort(){

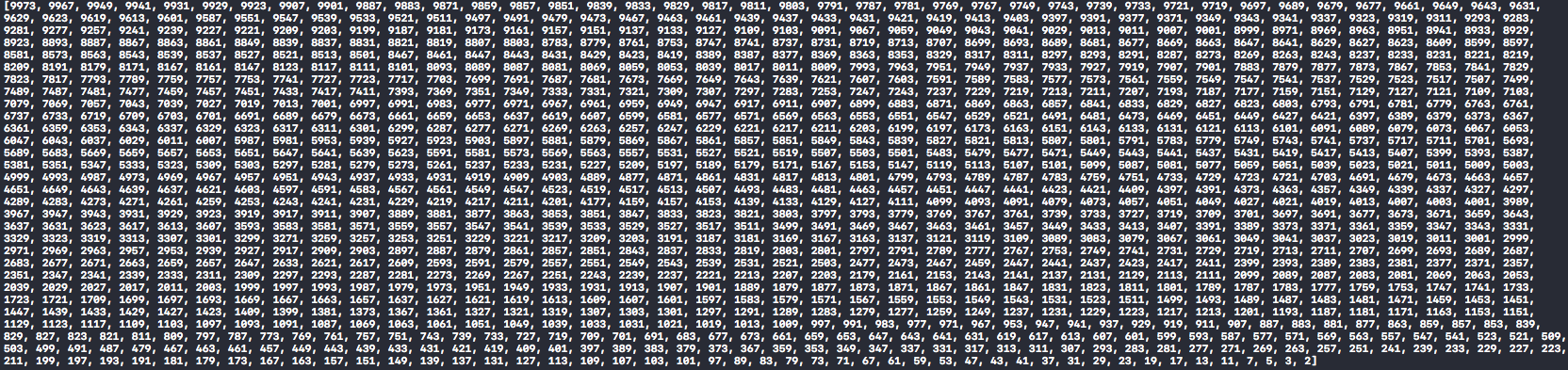
$0>$1

}

print(primes)

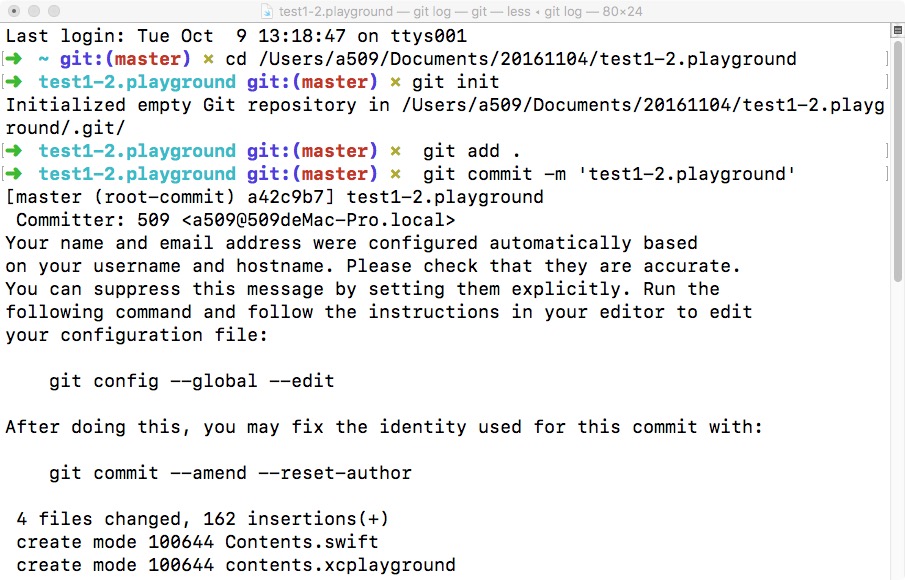
运行截图：

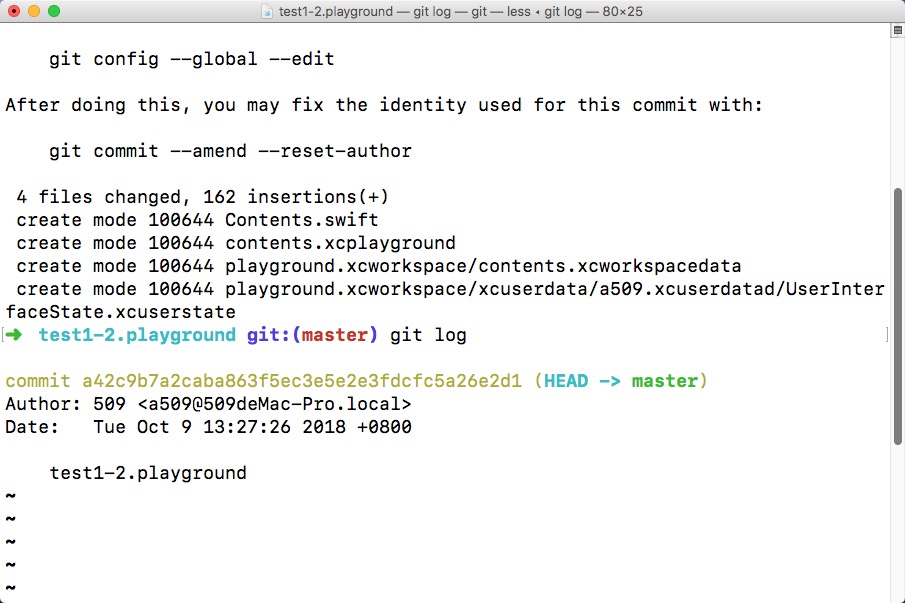




掌握版本控制git的基本使用

* 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、协议）

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；

实验代码：

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

import UIKit

var str = "Hello, playground"

enum Gender:Int{ //gender是枚举类型

case male

case female

//重载>操作符，便于后面排序使用

static func >(Ihs:Gender,rhs:Gender)->Bool{

return Ihs.rawValue < rhs.rawValue

}

}

class Person{//Person类(父类)

var firstName:String //存储属性：名

var lastName:String //存储属性：姓

var age:Int //存储属性：年龄

var gender:Gender //存储属性：性别

var fullName:String{ //计算属性：全名

get{ //get方法

return firstName+""+lastName

}

}

static func == (p1:Person,p2:Person)->Bool {

return p1.firstName == p2.firstName

}

init(firstName:String,lastName:String,age:Int,gender:Gender){//初始构造函数

self.firstName = firstName

self.lastName = lastName

self.age = age

self.gender = gender

}

convenience init(name:String){//便利构造函数

self.init(firstName:name,lastName:"",age:18,gender:Gender.female)

}

var description:String{//实现Person实例可以直接用print输出

return "Name:\(fullName) Age:\(age) Gender:\(gender) "

}

}

class Student:Person{//由Person派生出Student子类

var stuNo:String //增加了stuNo(学号)存储属性

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

self.stuNo = stuNo

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

//调用父类的构造函数

}

convenience init(name:String){

self.init(firstName:name,lastName:"",age:18,gender:Gender.male,stuNo:"201611001")

}

override var description:String{//重写父类方法实现Student实例可以直接用print输出

return super.description + " stuNo:\(stuNo)"

}

}

class Teacher:Person{ //由Person派生出Teacher子类

var title:String //增加了title(科目)存储属性

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

self.title = title

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

//调用父类的构造函数

}

convenience init(name:String){

self.init(firstName:name,lastName:"",age:36,gender:Gender.male,title:"math")

}

override var description:String{//重写父类方法实现Teacher实例可以直接用print输出

return super.description + " title:\(title)"

}

}

//实例化对象并输出

let p1 = Person(firstName:"Li",lastName:"si",age:36,gender:Gender.male)

let p2 = Person(name:"zhangsan")

print(p1.description)

print(p2.description)

let s1 = Student(firstName:"xiao",lastName:"ming",age:19,gender:Gender.male,stuNo:"201611002")

let s2 = Student(name:"xiaohong")

print(s1.description)

print(s2.description)

let t1 = Teacher(firstName:"zhiao",lastName:"liu",age:32,gender:Gender.male,title:"English")

let t2 = Teacher(name:"wangsiwu")

print(t1.description)

print(t2.description)

//两个Person实例对象可以用==和!=进行比较

let p3 = p1 //直接比较会出错，需要重载==操作符号

if p1 == p3 {

print("is the same people")

}else{

print("is not the same people")

}

//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；

var people = [Person]()//初始化一个空的Person类型数组

people.append(p1)//将不同对象存入数组

people.append(p2)

people.append(s1)

people.append(s2)

people.append(t1)

people.append(t2)

//生成3个Person对象

for i in 1...3 {

let p = Person(firstName: "张", lastName: "\(i)", age: 30, gender:Gender.male)

people.append(p)

}

//生成3个Student对象

for i in 1...3 {

let s = Student(firstName:"刘",lastName: "\(i)",age:18,gender:Gender.female,stuNo:"20161100\(i)")

people.append(s)

}

//生成3个Teacher对象

for i in 1...3 {

let t = Teacher(firstName:"李",lastName:"\(i)",age:20,gender:Gender.male,title:"计算机")

people.append(t)

}

print(people)

//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；

////定义一个字典，用于统计每个类的对象个数

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

for item in people {

if item is Student {

dictionary["Teacher"]! += 1

}else if item is Teacher {

dictionary["Student"]! += 1

}else{

dictionary["Person"]! += 1

}

}

//输出字典值

for (key,value) in dictionary {

print("\(key) has \(value) items")

}

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

//原数组

for item in people{

print(item.description)

}

//按照age值从小到大排序

people.sort{

return $0.age < $1.age

}

for item in people {

print(item.description)

}

//按照fullName从小到大排序

people.sort{

return $0.fullName < $1.fullName

}

for item in people {

print(item.description)

}

//按照gender+age从小到大排序

people.sort{

return ($0.gender > $1.gender) && ($0.age > $1.age)

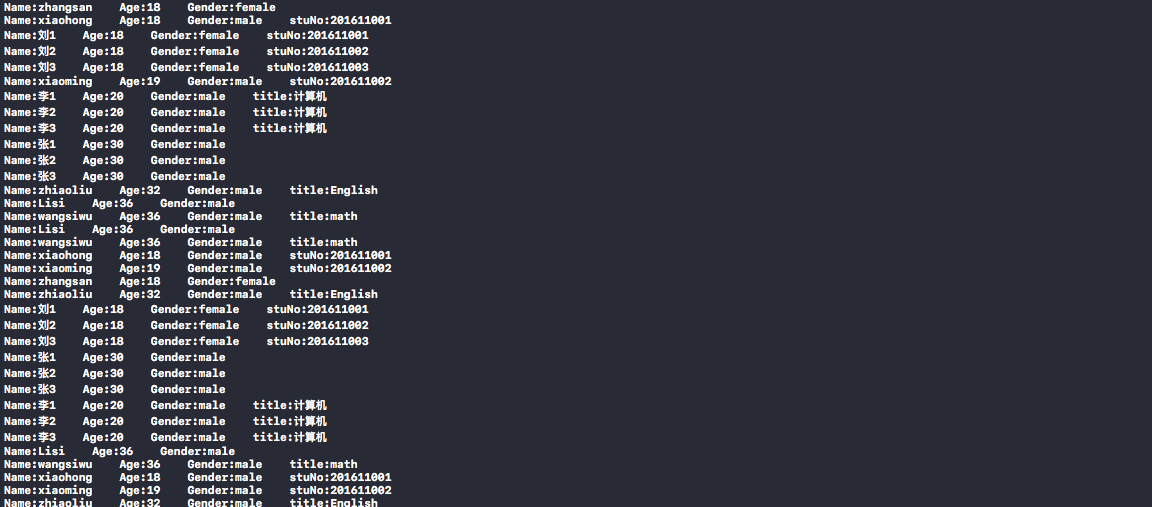
}

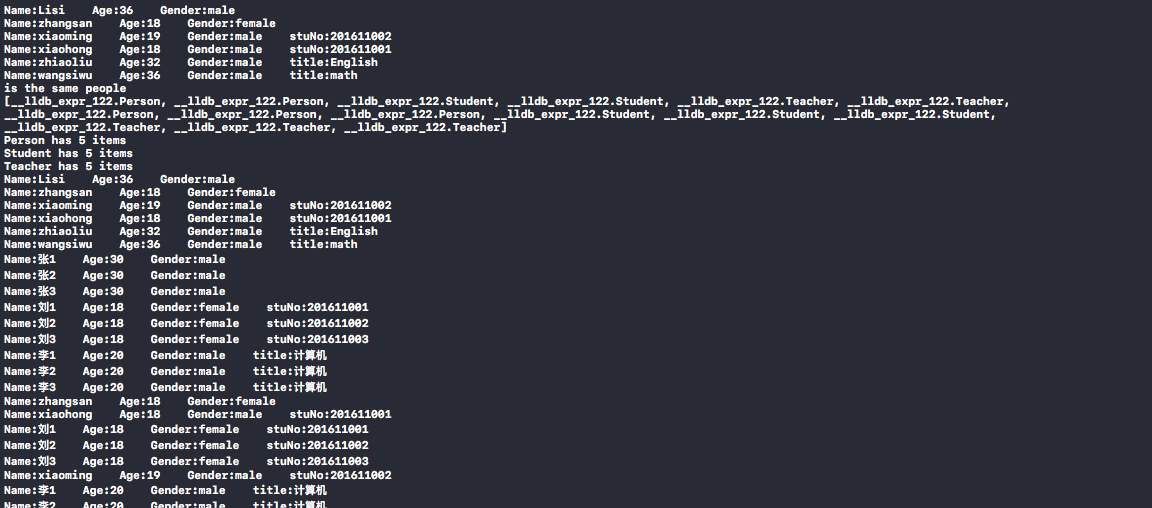
for item in people {

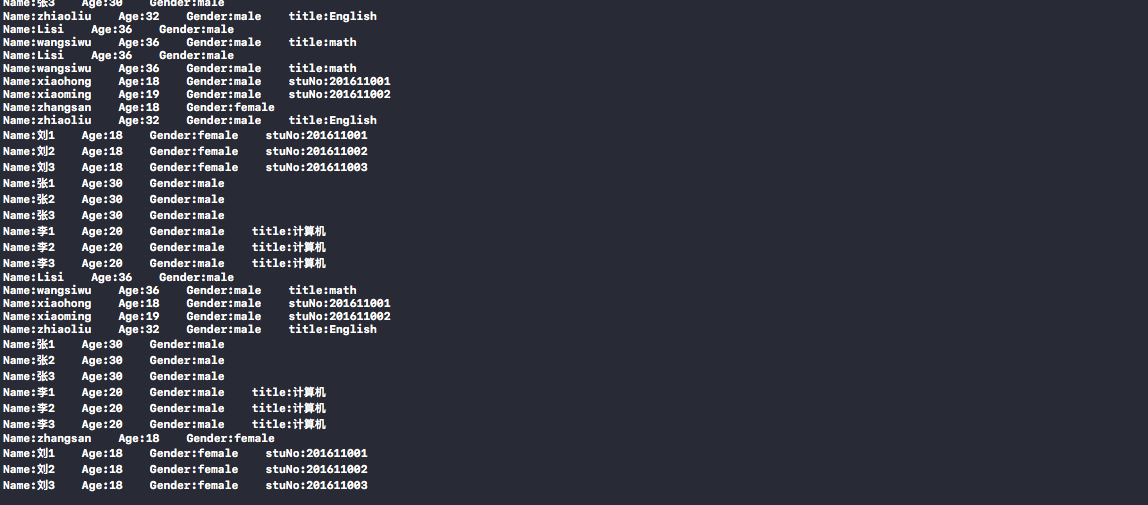
print(item.description)

}

运行截图：







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字符串的相关属性，字典，数组的相关定义用法以及类的定义，构造函数及对象的比较和重载函数的用法等相关的知识，体会到了swift语言相较于其他编程语言的不同，熟悉了IOS系统的相关操作。了解了git版本库的使用。

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。