实验编号： 8 **四川师大《IOS》实验报告 2019** 年 **11** 月 **4**日

### **计算机科学学院** 2017 级 4 班 实验名称： TextField和TableView \_

姓名：\_李晋\_\_\_ 学号：\_2017110517\_\_ 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验\_八\_ \_\_\_\_\_\_** TextField和TableView **\_\_\_\_\_\_\_**

1. **实验目的及要求**
2. 掌握TextField和Delegate、Notification的使用；
3. 掌握表视图TableView以及Datasource、Delegate等的使用；
4. **实验要求**
5. 认真填写实验报告，要求附加部分运行界面和主要代码；
6. 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；
7. **实验内容**
8. 将之前作业实现的Person 类和子类导入新项目，新建多个 Student 和 Teacher 对象并放入一数组中；
9. 数组排序后采用 TableView 的形式显示所有Student和Teacher对象的信息，要求每一个对象一个Cell，同时学生和教师显示不同的信息，选择一个Cell 后给出选择反馈；
10. 修改界面，增加添加新学生的部分(键盘要正确显示和消失)，要求实现TableView的增删改;

说明：键盘显示消失需要基于Notification 实现自适应键盘的界面布局;注册KeyboardDidShow和KeyBoardWillHide通知，并处理相应的action，进行界面的布局调整；

1. **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）**

（1）将之前作业实现的Person 类和子类导入新项目，新建多个 Student 和 Teacher 对象并放入一数组中；

**class** Person{

**var** name:String

**var** age:Int

**var** gender:Bool

**init**(name:String,age:Int,gender:Bool) {

**self**.name = name

**self**.age = age

**self**.gender = gender

}

}

**class** Student:Person {

**var** stuClass:String

**init**(name:String,age:Int,gender:Bool,stuClass:String) {

**self**.stuClass = stuClass

**super**.init(name: name, age: age, gender: gender)

}

}

**class** Teacher: Person {

**var** major:String

**init**(name:String,age:Int,gender:Bool,major:String) {

**self**.major = major

**super**.init(name: name, age: age, gender: gender)

}

}

**let** s1 = Student(name: "li", age: 10, gender: **false**, stuClass: "1班")

**let** s2 = Student(name: "liu", age: 30, gender: **false**, stuClass: "2班")

**let** s3 = Student(name: "张", age: 30, gender: **true**, stuClass: "12班")

**let** t1 = Teacher(name: "王", age: 44, gender: **true**, major: "数学")

**let** t2 = Teacher(name: "秦", age: 44, gender: **true**, major: "语文")

**let** t3 = Teacher(name: "广", age: 44, gender: **true**, major: "英语")

**let** t4 = Teacher(name: "柳", age: 44, gender: **true**, major: "历史")

**let** t5 = Teacher(name: "钧", age: 44, gender: **true**, major: "政治")

persons = [s1,s2,s3,t1,t2,t3,t4,t5]

（2）数组排序后采用 TableView 的形式显示所有Student和Teacher对象的信息，要求每一个对象一个Cell，同时学生和教师显示不同的信息，选择一个Cell 后给出选择反馈；

*//*按照*age*排序

persons.sort(by:{

$0.age < $1.age

})

*//*行数

**func** tableView(**\_** tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

**return** persons.count

}

*//cell*的内容

**func** tableView(**\_** tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

**let** cell:UITableViewCell?

**let** person = persons[indexPath.row]

**if** **let** student = person **as**? Student{

**let** studentCell = tableView.dequeueReusableCell(withIdentifier: "green") **as**! StudentTableViewCell

studentCell.nameLabel.text = student.name

studentCell.ageLabel.text = String(student.age)

studentCell.classLabel.text = student.stuClass

**if** student.gender{

studentCell.genderImageView.image = UIImage(named: "xingbienan")

}**else**{

studentCell.genderImageView.image = UIImage(named: "xingbienv")

}

cell = studentCell

}**else** {

**let** teacher = person **as**? Teacher

**let** teacherCell = tableView.dequeueReusableCell(withIdentifier: "pink") **as**! TeacherTableViewCell

teacherCell.nameLabel.text = teacher!.name

teacherCell.ageLabel.text = String(teacher!.age)

teacherCell.majorLabel.text = teacher?.major

**if** teacher!.gender{

teacherCell.genderImageView.image = UIImage(named: "xingbienan")

}**else**{

teacherCell.genderImageView.image = UIImage(named: "xingbienv")

}

cell = teacherCell

}

**return** cell!

}

*//MARK:UITableViewDelegate*代理

*//*设置行高

**func** tableView(**\_** tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {

**return** 80

}

*//*选中时：显示选中的信息

**func** tableView(**\_** tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

**var** message:String = " "

**if** **let** stucell = tableView.cellForRow(at: indexPath) **as**? StudentTableViewCell{

**let** name = stucell.nameLabel.text

**let** age = stucell.ageLabel.text

**let** stuClass = stucell.classLabel.text

message = "学生\n姓名：\(name!)\n年龄：\(age!)\n班级：\(stuClass!)"

}**else**{

**let** teacell = tableView.cellForRow(at: indexPath) **as**? TeacherTableViewCell

**let** name = teacell!.nameLabel.text

**let** age = teacell!.ageLabel.text

**let** major = teacell!.majorLabel.text

message = "老师\n姓名：\(name!)\n年龄：\(age!)\n教学科目：\(major!)"

}

messageLabel.text = message

}

（3）修改界面，增加添加新学生的部分(键盘要正确显示和消失)，要求实现TableView的增删改;

说明：键盘显示消失需要基于Notification 实现自适应键盘的界面布局;注册KeyboardDidShow和KeyBoardWillHide通知，并处理相应的action，进行界面的布局调整；

//设置cell的编辑模式

func tableView(\_ tableView: UITableView, editingStyleForRowAt indexPath: IndexPath) -> UITableViewCell.EditingStyle {

return .delete

}

//设置可以编辑的行

func tableView(\_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {

return true

}

//设置删除操作

func tableView(\_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {

if editingStyle == .delete{

persons.remove(at: indexPath.row)

tableView.deleteRows(at: [indexPath], with: .fade)

}

}

//设置可以移动的行

func tableView(\_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool {

return true

}

//设置移动操作

func tableView(\_ tableView: UITableView, moveRowAt sourceIndexPath: IndexPath, to destinationIndexPath: IndexPath) {

persons.insert(persons.remove(at: sourceIndexPath.row), at: destinationIndexPath.row)

}

//MARK:按钮触发事件

//添加学生

@IBAction func addStudent(\_ sender: Any) {

guard let name = nameTF.text,let ageString = ageTF.text,let age = Int(ageString),let stuClass = classTF.text else{

return

}

let gender = genderSC.selectedSegmentIndex

let theStudent = Student(name: name, age: age, gender: gender == 0 ? true : false, stuClass: stuClass

)

persons.append(theStudent)

myTV.reloadData()

}

//编辑学生

@IBAction func editStudent(\_ sender: UIButton) {

//将UITableView设置为编辑模式

//myTV.isEditing = true

myTV.setEditing(!myTV.isEditing, animated: true)

sender.setTitle(myTV.isEditing ? "正在编辑" : "编辑学生", for: .normal)

}

//MARK:通知(键盘出现，消失遮挡)

NotificationCenter.default.addObserver(self, selector: #selector(keyboardShouldAppear), name: UIResponder.keyboardWillShowNotification, object: nil)

NotificationCenter.default.addObserver(self, selector: #selector(keyboardShouldDisappear), name: UIResponder.keyboardWillHideNotification, object: nil)

}

//键盘出现，视图上移

@objc func keyboardShouldAppear(\_ notification:Notification){

let rect = notification.userInfo!["UIKeyboardBoundsUserInfoKey"] as! CGRect

let height = rect.size.height

self.view.frame.origin.y = -height

}

//键盘消失，视图下移

@objc func keyboardShouldDisappear(){

self.view.frame.origin.y = 0

 }

1. **实验结果的分析与评价（该部分如不够填写，请另加附页）**
2. 能够熟练地设置textfield的键盘出现消失，熟练掌握了textfield的代理
3. 学会了TableView的使用，能熟练地设置数据源和代理。使用数据源设置tableview的section数目和其中的行数，以及每行的内容
4. 学会了自己定制TableViewCell，制作符合自己需求的TableViewCell。

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。