实验编号： 12 **四川师大《IOS》实验报告 2018** 年 **11** 月 **28** 日

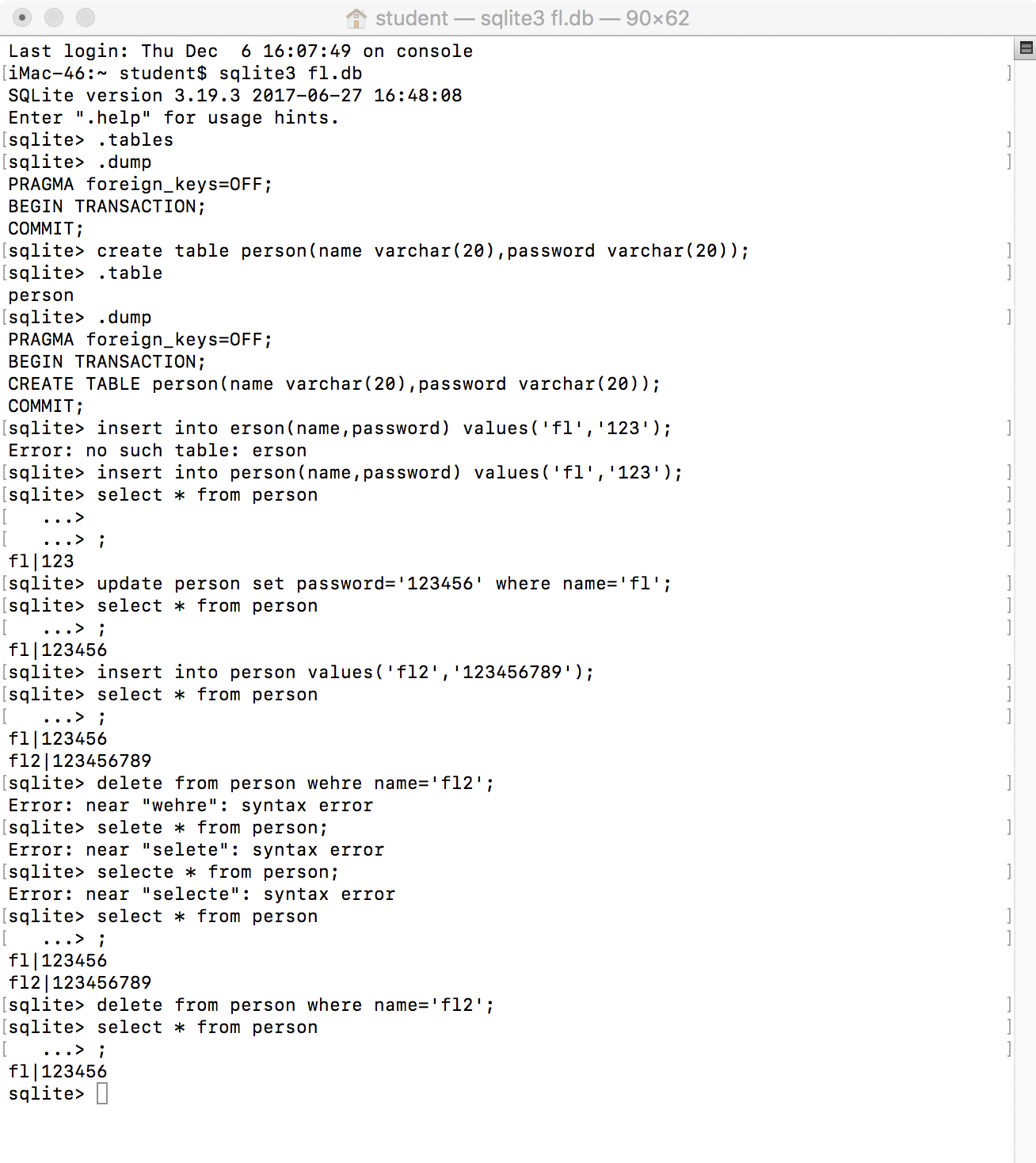
### **计算机科学学院** 2016 级 4 班 实验名称： 数据持久化 \_

姓名：\_\_樊琳\_\_ 学号：\_2016110408\_ 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

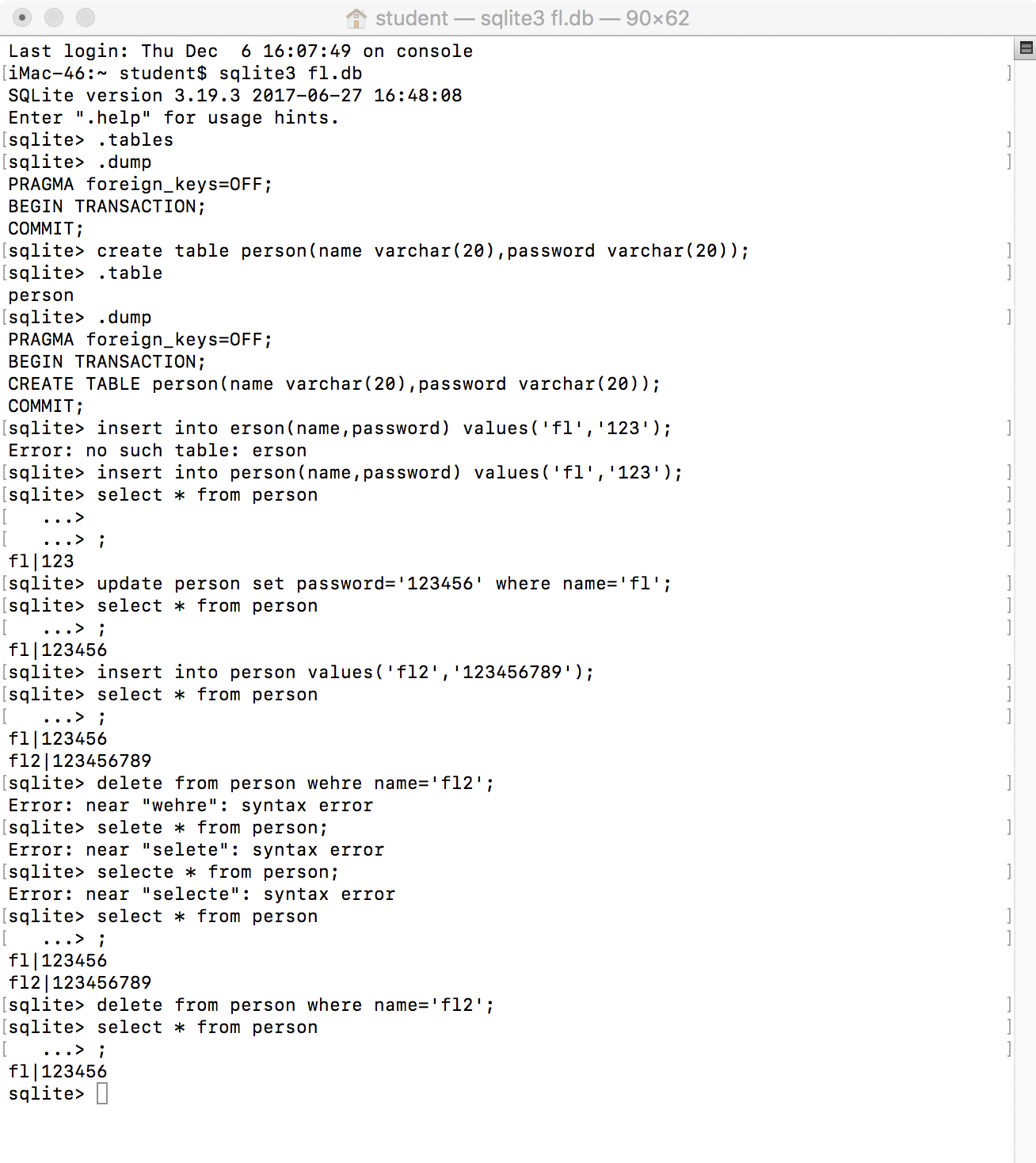
**实验 十二 \_\_\_\_\_\_**数据持久化**\_\_\_\_\_\_\_**

1. 实验目的及要求
2. 掌握iOS中数据持久化的使用；
3. 掌握SQLite和CoreData的原理和使用；
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. 在命令提示符下熟悉并使用SQLite命令
   1. 新建sqlite3数据库；
   2. 新建Person表；
   3. 插入新的数据；
   4. 查询数据；
9. 采用多MVC和SQLite或Core Data构造一个Person信息维护程序，要求：
   1. 可插入新数据；
   2. 可删除修改原数据；
   3. 可查询特定数据；
10. 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）
11. 在命令提示符下熟悉并使用SQLite命令
    1. 新建sqlite3数据库；
    2. 新建Person表；
    3. 插入新的数据；
    4. 查询数据；

* 程序代码：



* 运行结果：



1. 采用多MVC和SQLite或Core Data构造一个Person信息维护程序，要求：
   1. 可插入新数据；
   2. 可删除修改原数据；
   3. 可查询特定数据；
      * 程序代码：

//

// AppDelegate.swift

// 12

//

// Created by student on 2018/12/15.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import CoreData

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

window?.rootViewController = UINavigationController(rootViewController: ViewController())

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

// Saves changes in the application's managed object context before the application terminates.

self.saveContext()

}

// MARK: - Core Data stack

lazy var persistentContainer: NSPersistentContainer = {

/\*

The persistent container for the application. This implementation

creates and returns a container, having loaded the store for the

application to it. This property is optional since there are legitimate

error conditions that could cause the creation of the store to fail.

\*/

let container = NSPersistentContainer(name: "homework")

container.loadPersistentStores(completionHandler: { (storeDescription, error) in

if let error = error as NSError? {

// Replace this implementation with code to handle the error appropriately.

// fatalError() causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it may be useful during development.

/\*

Typical reasons for an error here include:

\* The parent directory does not exist, cannot be created, or disallows writing.

\* The persistent store is not accessible, due to permissions or data protection when the device is locked.

\* The device is out of space.

\* The store could not be migrated to the current model version.

Check the error message to determine what the actual problem was.

\*/

fatalError("Unresolved error \(error), \(error.userInfo)")

}

})

return container

}()

// MARK: - Core Data Saving support

func saveContext () {

let context = persistentContainer.viewContext

if context.hasChanges {

do {

try context.save()

} catch {

// Replace this implementation with code to handle the error appropriately.

// fatalError() causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it may be useful during development.

let nserror = error as NSError

fatalError("Unresolved error \(nserror), \(nserror.userInfo)")

}

}

}

}

//

// ViewController.swift

// 12

//

// Created by student on 2018/12/15.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import CoreData

class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

var tableView: UITableView!

var personArray = [Person]()

let appDelegate = UIApplication.shared.delegate as! AppDelegate

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

self.title = "Persons"

self.view.backgroundColor = UIColor.white

tableView = UITableView(frame: self.view.frame)

tableView.delegate = self

tableView.dataSource = self

self.view.addSubview(tableView)

let searchBtn = UIBarButtonItem(barButtonSystemItem: .search, target: self, action: #selector(search))

self.navigationItem.leftBarButtonItem = searchBtn

let addBtn = UIBarButtonItem(barButtonSystemItem: .add, target: self, action: #selector(add))

self.navigationItem.rightBarButtonItem = addBtn

}

override func viewWillAppear(\_ animated: Bool) {

let context = appDelegate.persistentContainer.viewContext

let request: NSFetchRequest<Person> = Person.fetchRequest()

if let persons = try? context.fetch(request) {

personArray = persons

}

tableView.reloadData()

}

@objc func search() {

let viewController = SearchViewController()

self.navigationController?.pushViewController(viewController, animated: true)

}

@objc func add() {

let viewController = InsertViewController()

self.navigationController?.pushViewController(viewController, animated: true)

}

// MARK: delegate

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

let viewController = InsertViewController()

viewController.row = indexPath.row

self.navigationController?.pushViewController(viewController, animated: true)

}

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

return .delete

}

// MARK: data source

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

return personArray.count

}

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

var cell = tableView.dequeueReusableCell(withIdentifier: "Cell")

if cell == nil {

cell = UITableViewCell(style: .subtitle, reuseIdentifier: "Cell")

cell?.accessoryType = .disclosureIndicator

}

cell?.textLabel?.text = personArray[indexPath.row].name

cell?.detailTextLabel?.text = String(personArray[indexPath.row].age)

return cell!

}

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

if editingStyle == .delete {

personArray.remove(at: indexPath.row)

tableView.reloadData()

DispatchQueue.global().async {

let context = self.appDelegate.persistentContainer.viewContext

let request: NSFetchRequest<Person> = Person.fetchRequest()

if let persons = try? context.fetch(request) {

context.delete(persons[indexPath.row])

self.appDelegate.saveContext()

}

}

}

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

}

//

// InsertViewController.swift

// 12

//

// Created by student on 2018/12/15.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import CoreData

class InsertViewController: UIViewController {

var row: Int?

var nameTextField: UITextField!

var ageTextField: UITextField!

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

self.view.backgroundColor = UIColor.white

nameTextField = UITextField(frame: CGRect(x: 20, y: 100, width: 300, height: 44))

nameTextField.layer.borderWidth = 1

self.view.addSubview(nameTextField)

ageTextField = UITextField(frame: CGRect(x: 20, y: 200, width: 300, height: 44))

ageTextField.layer.borderWidth = 1

self.view.addSubview(ageTextField)

let btn = UIBarButtonItem(barButtonSystemItem: .done, target: self, action: #selector(done))

self.navigationItem.rightBarButtonItem = btn

if row == nil {

self.title = "Add"

} else {

self.title = "Edit"

let appDelegate = UIApplication.shared.delegate as! AppDelegate

let context = appDelegate.persistentContainer.viewContext

let request: NSFetchRequest<Person> = Person.fetchRequest()

if let persons = try? context.fetch(request) {

nameTextField.placeholder = persons[row!].name

ageTextField.placeholder = String(persons[row!].age)

}

}

}

@objc func done() {

let appDelegate = UIApplication.shared.delegate as! AppDelegate

let context = appDelegate.persistentContainer.viewContext

let name = self.nameTextField.text!

let age = Int16(self.ageTextField.text!)

if row == nil {

if !name.isEmpty && age != nil {

let person = Person(context: context)

person.name = name

person.age = age!

}

} else {

let request: NSFetchRequest<Person> = Person.fetchRequest()

if let persons = try? context.fetch(request) {

if !name.isEmpty {

persons[row!].name = name

}

if age != nil {

persons[row!].age = age!

}

}

}

appDelegate.saveContext()

self.navigationController?.popViewController(animated: true)

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

/\*

// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

// Get the new view controller using segue.destinationViewController.

// Pass the selected object to the new view controller.

}

\*/

}

//

// SearchViewController.swift

// 12

//

// Created by student on 2018/12/15.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import CoreData

class SearchViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

var nameTextField: UITextField!

var ageTextField: UITextField!

var tableView: UITableView!

var persons = [Person]()

let appDelegate = UIApplication.shared.delegate as! AppDelegate

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

self.view.backgroundColor = UIColor.white

self.title = "Search"

nameTextField = UITextField(frame: CGRect(x: 10, y: 100, width: 200, height: 44))

nameTextField.layer.borderWidth = 1

self.view.addSubview(nameTextField)

let nameBtn = UIButton(frame: CGRect(x: 250, y: 100, width: 100, height: 44))

nameBtn.setTitle("按名字查找", for: .normal)

nameBtn.setTitleColor(UIColor.cyan, for: .normal)

nameBtn.setTitleColor(UIColor.brown, for: .highlighted)

nameBtn.addTarget(self, action: #selector(searchWithName), for: .touchUpInside)

self.view.addSubview(nameBtn)

ageTextField = UITextField(frame: CGRect(x: 10, y: 150, width: 200, height: 44))

ageTextField.layer.borderWidth = 1

self.view.addSubview(ageTextField)

let ageBtn = UIButton(frame: CGRect(x: 250, y: 150, width: 100, height: 44))

ageBtn.setTitle("按年龄查找", for: .normal)

ageBtn.setTitleColor(UIColor.cyan, for: .normal)

ageBtn.setTitleColor(UIColor.brown, for: .highlighted)

ageBtn.addTarget(self, action: #selector(searchWithAge), for: .touchUpInside)

self.view.addSubview(ageBtn)

tableView = UITableView(frame: CGRect(x: 0, y: 200, width: self.view.frame.width, height: self.view.frame.height - 200))

tableView.delegate = self

tableView.dataSource = self

self.view.addSubview(tableView)

}

@objc func searchWithName() {

ageTextField.text = ""

let context = appDelegate.persistentContainer.viewContext

let name = nameTextField.text

let request: NSFetchRequest<Person> = Person.fetchRequest()

if name != nil {

request.predicate = NSPredicate(format: "name = %@", name!)

}

if let persons = try? context.fetch(request) {

self.persons = persons

tableView.reloadData()

}

}

@objc func searchWithAge() {

nameTextField.text = ""

let context = appDelegate.persistentContainer.viewContext

let age = Int16(ageTextField.text!)

let request: NSFetchRequest<Person> = Person.fetchRequest()

if age != nil {

request.predicate = NSPredicate(format: "age = %d", age!)

}

if let persons = try? context.fetch(request) {

self.persons = persons

tableView.reloadData()

}

}

// MARK: data source

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

return persons.count

}

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

var cell = tableView.dequeueReusableCell(withIdentifier: "Cell")

if cell == nil {

cell = UITableViewCell(style: .subtitle, reuseIdentifier: "Cell")

}

cell?.textLabel?.text = persons[indexPath.row].name

cell?.detailTextLabel?.text = String(persons[indexPath.row].age)

return cell!

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

/\*

// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

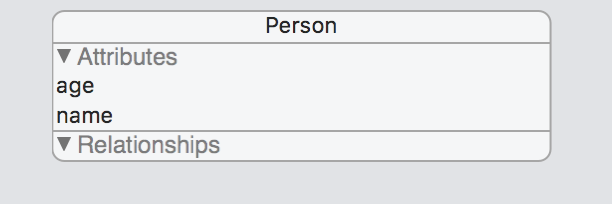
// Get the new view controller using segue.destinationViewController.

// Pass the selected object to the new view controller.

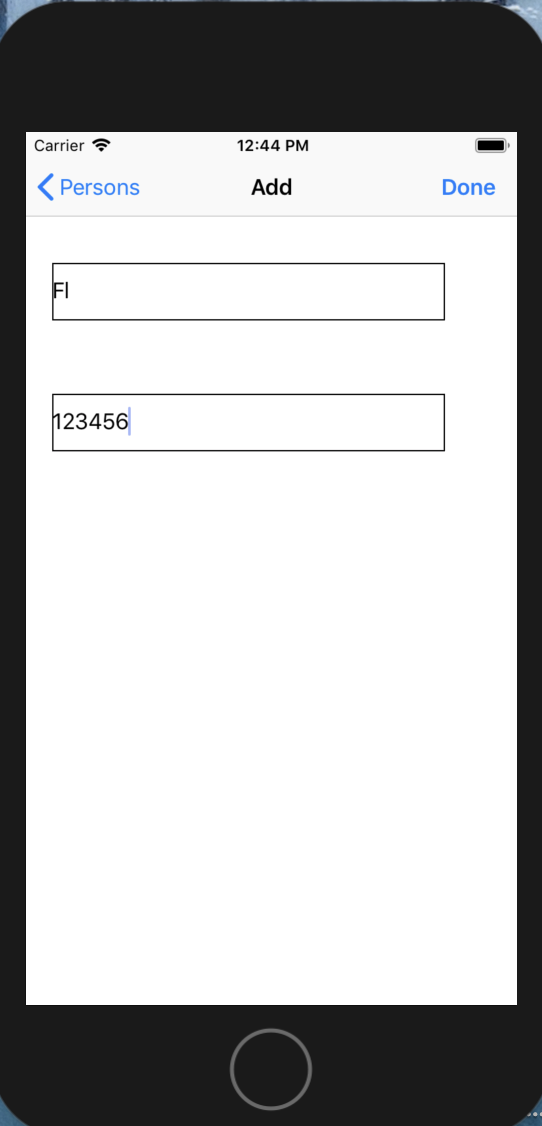
}

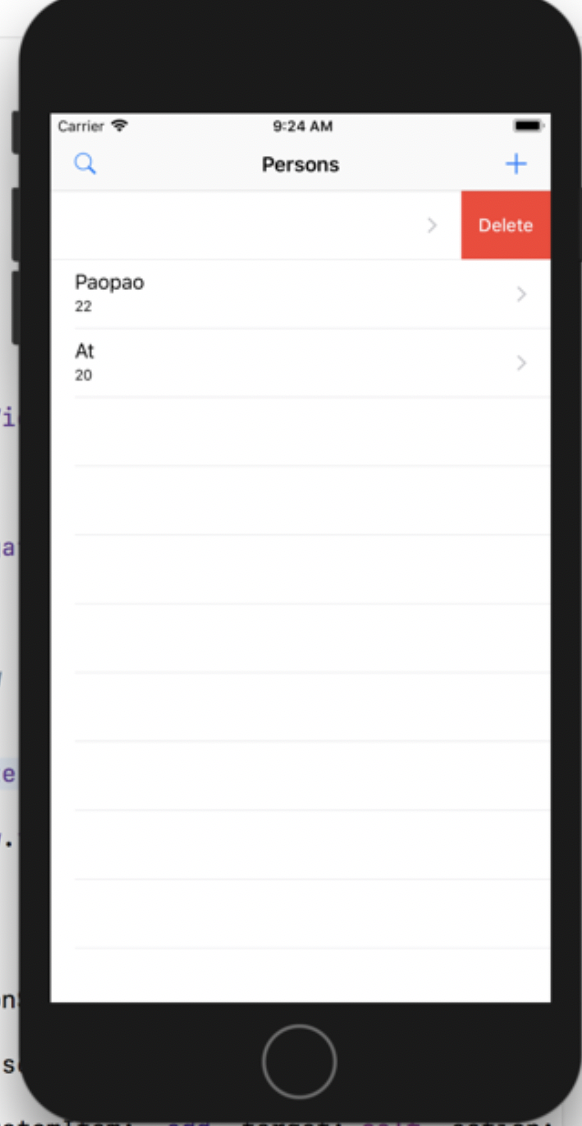
\*/

}



* + - 运行结果：





实验结果的分析与评价（该部分如不够填写，请另加附页）

Coredata运用起来真的很简单。比Java那些还有简单。

Github地址：

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。