

```
/**
 摘要:
 完成私人通讯录案例，灵活使用导航控制器
 并掌握本地数据存储方式：如plist文件，用户偏好设置，归档
*/
```

一、私人通讯录

"【登录界面】"

1. 掌握UIStoryboardSegue对象

- (1) 在storyboard中，可以通过连线完成多个界面间的跳转，每一连线我们称为segue
- (2) 掌握storyboard中的segue连线有两种，一是自动跳转、二是手动跳转
- (3) 掌握如何使用代码跳转页面手动segue --> `[[self performSegueWithIdentifier:@"contactSegue" sender:nil];]`
- (4) 掌握UIStoryboardSegue对象的源控制器和目标控制器

2. 掌握MBProgressHUD的使用，给用户一个友好的提示

3. 掌握控制器的数据传递

- (1) 在控制器的 `-(void)prepareForSegue(sender:)` 的方法中获取目标控制器
- (2) 设置目标控制器的属性，可以实现数据的传递

4. 掌握文件框文字改变的监听

方法1 使用通知中心

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(textChange) name:UITextFieldTextDidChangeNotification
object:self.passwordField];
```

方法2 往文本输入框添加事件

```
【addTarget: action: forControlEvents:UIControlEventEditingChanged】
```

方法3 storyboard连线

5. 掌握记住密码与自动登录逻辑

- (1) 记住密码为关闭的情况下，如果自动登录为开启状态，应该关闭
- (2) 如果自动登录为关闭状态，记住密码必需为开启状态

"【添加联系人页面】"

1. 掌握数据的逆向传递

- (1) 如果想向上一个控制器传递数据，可以使用代理的方法
- (2) 向上个控制器传递数据时，可以传递字符串或者模型

2. 掌握表格的局部刷新，性能更好，使用 `self.tableView insertRowsAtIndexPaths:withRowAnimation:` 方法

"【编辑联系人】"

1. 使用代理的方法通知上一个控制器联系个编辑完成，并保存
2. 刷新数据使用局部刷新 `self.tableView reloadRowsAtIndexPaths:withRowAnimation`

二、数据存储

"iOS数据存储的方式有哪些"

1. plist
2. preference 用户偏好设置
3. NSKeyedArchiver(归档)
4. sqlite
5. coredata

"plist文件存储"

1. 字符串 数组 字典 可以直接存储数据在一个文件
2. 掌握沙盒目录的作用以及目录路径获取方式

```
// Document [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) lastObject]
// 缓存 [NSSearchPathForDirectoriesInDomains(NSCachesDirectory, NSUserDomainMask,
YES) lastObject];
// 临时 NSTemporaryDirectory();
// 主目录 NSHomeDirectory();
```
3. 不是所有对象都可以保存到plist文件中，要有实现writeFile方法才可以
4. 学会使用SimPholders2打开沙盒目录

"用户偏好设置"

1. ios中有个NSUserDefaults对象有可保存数据，我们称为用户偏好设置
2. 通过[NSUserDefaults standardUserDefaults]可以获取用户偏好设置对象，保存字符串 布尔值 int等数据
3. 保存数据时，一定要调用synchronize，因为数据要及时保存到沙盒的文件中

```
/* NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];

[defaults setObject:@"zhangsan" forKey:@"username"];
[defaults setObject:@"123" forKey:@"password"];
[defaults setBool:YES forKey:@"autoLogin"];
[defaults setBool:YES forKey:@"rememberPwd"];

//同步
[defaults synchronize];
*/
```
4. 掌握用户偏好设置数据的获取，更改，删除

"NSKeyedArchiver归档"

- ```
/*什么叫归档 归档就是把数据保存到一个文件中*/
```
1. 使用NSKeyedArchiver可以将NSArray NSDictionary NSString等对象归档到一个文件
  2. 只有实现了NSCoding协议的对象才可使用NSKeyedArchiver进行归档
  3. 将模型对象保存到一个文件时，对象要遵守NSCoding协议，并实现NSKeyedArchiver的encodeWithCoder方法，
  4. 从归档文件里读取对象时要实现NSCoding的initWithCoder方法

如下图，如果想把Person对象保存到文件并读了，要实现下面两个方法

## @implementation CZPerson

```
/**
 * 将某个对象写入文档时会调用
 * 这个方法清楚的告诉哪些属性要存储
 */
-(void)encodeWithCoder:(NSCoder *)encoder{
 [encode encodeObject:self.name forKey:@"name"];
 [encode encodeInt:self.age forKey:@"age"];
 [encode encodeDouble:self.height forKey:@"height"];
}

//从文件解析对象时会调用
//这个方法告诉解析出来的数据放到对应的属性中去
-(id)initWithCoder:(NSCoder *)decoder{

 if (self =[super init]) {
 self.name = [decoder decodeObjectForKey:@"name"];
 self.age = [decoder decodeIntForKey:@"age"];
 self.height = [decoder decodeDoubleForKey:@"height"];
 }

 return self;
}
```

### 三、完善通讯录的登录页面数据存储与联系人信息存储

- 1.登录页面使用 "用户偏好" 来存储数据
- 2.联系人页面使用 "归档" 来存储数据

### 四、删除联系人/随机添加联系人

#### 1.掌握表格的编辑状态

- \*表格有个editing属性，可以打开cell的删除或者添加状态
- \*指定表格的编辑状态，可以实现代理方法

```
/**
 表格的编辑状态
 -(UITableViewCellStyle)tableView:(UITableView *)tableView
 editingStyleForRowAtIndexPath:(NSIndexPath *)indexPath{
 //第一行返回的编辑状态为添加
 if (indexPath.row == 0) {
 return UITableViewCellEditingStyleInsert;
 }
 //其它行返回的编辑状态为删除
 return UITableViewCellEditingStyleDelete;
 }
 */
```



\*如果要监听表格绿色与红色按钮的点击，只要实现一个代理方法即可，如下

```
/*
-(void)tableView:(UITableView *)tableView commitEditingStyle:
(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:(NSIndexPath
*)indexPath{
 if (editingStyle == UITableViewCellEditingStyleInsert) {//添加

 }else{//删除

 }
}
*/
```

五、实现联系人表格cell的分隔线

(1)系统的分隔线不好看，去除表格自带的分隔线

```
self.tableView.separatorStyle = UITableViewCellSeparatorStyleNone;
```

(2)自定义cell，在awakeFromNib方法中添加分隔线

"为什么要在awakeFromNib中添加分隔，因为在类方法不能调用实例方法"

(3)在layoutSubviews中设置分隔线的frm