作业

**1.**

**APP应用程序的状态**

Not Running：app还没运行

Inactive：app运行在前台但没有接收事件 即将活跃

Active：app运行在前台和正在接收事件 活跃中

Background：运行在后台和正在执行代码 后台

Suspended：运行在后台但没有执行代码 挂起

AppDelegate中的方法

响应app运行时重要事件，状态之间的转换

程序第一次启动后执行 didFinishLaunchingWithOptions

程序将要失去焦点时执行 applicationWillResignActive

程序得到焦点时执行 applicationDidBecomeActive

程序进入后台执行 applicationDidEnterBackground

程序进入前台执行 applicationDidEnterForeground

程序即将退出时调用 applicationWillTerminate

程序发生内存警告的时候调用 applicationDidReceiveMemoryWarning

应用程序状态切换

启动一个App 从Not running -> inactive -> active

此时调用didFinishLaunchingWithOptions方法，然后inactive切换到active，切换完成调用applicationDidBecomeActive

当App发生中断时，由active状态切换到inactive状态，此时调用applicationWillResignActive方法

当切换App时，由状态active切换到inactive，此时调用applicationWillResignActive方法；然后从inactive状态切换到running状态，此时调用applicationDidEnterBackground方法。

而当切换回本来的app时，由running状态切换到inactive状态，此时调用applicationWillEnterForeground方法，然后由inactive状态切换到active状态，调用applicationDidBecomeActive方法。

当手机锁屏时，由状态active切换到inactive，此时调用applicationWillResignActive；然后再由inactive状态切换到running状态，此时调用applicationDidEnterBackground方法。

系统常常是为其他app启动时由于内存不足而回收内存最后需要终止应用程序，但有时也会是由于app很长时间才响应而终止。如果app当时运行在后台并且没有暂停，系统会在应用程序终止之前调用applicationWillTerminate来保存用户的一些重要数据以便下次启动时恢复到app原来的状态。

iOS App基于MVC

View与用户交互事件，通过Action-Target的方式来响应View的事件。

当View需要处理一些特殊UI逻辑或获取数据源时，通过delegate或data source方式交给Controller来处理。

Model的更新通过Notification和KVO来通知Controller更新View

**ViewController的生命周期**

init

如果通过xib或storyboard创建，会首先调用该方法

loadView

此时VC还未初始化，可以重写该方法自定义控制器的View，若重写逻辑，就不能调用[super loadView]

viewDidLoad

在VC的生命周期中，只会被调用一次。此时view已经初始化好，可以做一些页面初始化任务，由于此时view的bounds尚未确定，所以不适合写frame类型的布局代码，但是给视图添加约束没有影响。

viewWillAppear

VC的视图将要出现时调用，可能会调用多次，适合做一些与视图相关联的任务，适合做一些与视图出现相关联的任务，例如改变状态栏的方向、风格。

viewWillLayoutSubviews

view的子视图即将布局，触发条件是

view 被添加到视图层次中

view 的 bounds 发生了改变

调用了 - setNeedsLayout方法

调用了 - layoutIfNeeded方法

viewDidLayoutSubviews

当 view 的子视图已经布局时，此方法会被调用，触发条件同上。

viewWillDisappear

视图即将从屏幕中消失时触发。

viewDidAppear

视图从屏幕中消失后触发。

**2**.

UIButton 按钮

UILabel 文本标签

UITextField 文本输入框

UIImageView 图片显示

UIScrollView 滚动的控件

UITableView 表格

UICollectionView 九宫格

UIWebView 网页显示控件

UIAlertView 对话框（中间弹框）

**3.**

1）(NSInteger)numberOfSectionsInTableView:(UITableView \*)tableView;

这个方法返回 tableview 有多少个section。

[cpp] view plain copy

//返回有多少个Sections

- (NSInteger)numberOfSectionsInTableView:(UITableView \*)tableView

{

return 1;

}

2）（NSInteger)tableView:(UITableView \*)table numberOfRowsInSection:(NSInteger)section;

这个方法返回 对应的section有多少个元素，也就是多少行。

[cpp] view plain copy

(NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section

{

return 10；

}

3)(CGFloat)tableView:(UITableView\*)tableViewheightForRowAtIndexPath:(NSIndexPath\*)indexPath;

这个方法返回指定的 row 的高度。

(CGFloat)tableView:(UITableView \*)tableView heightForHeaderInSection:(NSInteger)section;

这个方法返回指定的 section的header view 的高度。

(CGFloat)tableView:(UITableView \*)tableView heightForFooterInSection:(NSInteger)section;

这个方法返回指定的 section的footer view 的高度。

4)(UITableViewCell\*)tableView:(UITableView\*)tableViewcellForRowAtIndexPath:(NSIndexPath\*)indexPath;

返回指定的row 的cell。这个地方是比较关键的地方，一般在这个地方来定制各种个性化的 cell元素。这里只是使用最简单最基本的cell 类型。其中有一个主标题 cell.textLabel 还有一个副标题cell.detailTextLabel, 还有一个 image在最前头 叫 cell.imageView. 还可以设置右边的图标，通过cell.accessoryType 可以设置是饱满的向右的蓝色箭头，还是单薄的向右箭头，还是勾勾标记。