**联系人管理4（File’s owner）**

**ViewController.h**

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

//添加

- (IBAction)add:(UIBarButtonItem \*)sender;

//删除

- (IBAction)remove:(UIBarButtonItem \*)sender;

-(IBAction)deleteClick:(UIButton \*)btn;

@property (weak, nonatomic) IBOutlet UIBarButtonItem \*removeItem;

@end

**ViewController.m**

#import "ViewController.h"

#define krowH 50

#define kNameTag 10

//类扩展(class entension,匿名分类)

@interface ViewController ()

{

NSArray \*\_allNames;

}

@end

@implementation ViewController

#pragma mark 控制器的view加载完毕的时候调用一次

- (void)viewDidLoad {

[super viewDidLoad];

\_allNames = @[@"Mary",@"Jack",@"Rose",@"Jessica",@"Lily",@"Anna"];

}

#pragma mark 添加一行

- (IBAction)add:(UIBarButtonItem \*)sender {

//0、取出最后一个子控件

UIView \*last = [self.view.subviews lastObject];

//这行的Y值=最后一个子控件的Y值 + 最后一个子控件的高度 + 间距

CGFloat rowY = last.frame.origin.y + last.frame.size.height + 1;

//1、创建一行

// UIView \*rowView = [[UIView alloc] init];

// rowView.backgroundColor = [UIColor redColor];

UIView \*rowView = [self creatRowView];

//2、添加一行到控制器的view中

[self.view addSubview:rowView];

//3、让删除Item有效

\_removeItem.enabled = YES;

//4、执行动画

rowView.frame = CGRectMake(320, rowY, 320, krowH);

rowView.alpha = 0;

[UIView animateWithDuration:1.0 animations:^{

rowView.frame = CGRectMake(0, rowY, 320, krowH);

rowView.alpha = 1;

}];

}

#pragma mark 创建一行（从xib中加载一行的view）

-(UIView \*)creatRowView

{

//xib==nib

//1、加载RowView.xib文件，创建objects下面的所有控件，并且按顺序放到数组中返回

NSArray \*views = [[NSBundle mainBundle] loadNibNamed:@"RowView" owner:self options:nil];

//2、取出一行的红色view

UIView \*rowView = views[0];

//3、设置头像

UIButton \*icon = (UIButton \*)[rowView viewWithTag:1];

NSString \*iconName = [NSString stringWithFormat:@"01%d.png",arc4random\_uniform(9)];

[icon setImage:[UIImage imageNamed:iconName] forState:UIControlStateNormal];

//4、设置姓名

UILabel \*name = (UILabel \*)[rowView viewWithTag:2];

name.text = \_allNames[arc4random\_uniform(\_allNames.count)];

// //5、监听删除按钮

// UIButton \*delete = (UIButton \*)[rowView viewWithTag:3];

// [delete addTarget:self action:@selector(deleteClick:) forControlEvents:UIControlEventTouchUpInside];

return rowView;

}

#pragma mark 监听删除按钮点击

-(void)deleteClick:(UIButton \*)btn

{

[UIView animateWithDuration:1.0 animations:^{

CGRect tempF = btn.superview.frame;

tempF.origin.x = 320;

btn.superview.frame = tempF;

btn.superview.alpha = 0;

} completion:^(BOOL finished) {

//1、获得即将删除的这行在数组中的位置

int startIndex = [self.view.subviews indexOfObject:btn.superview];

//2、删除当前行

[btn.superview removeFromSuperview];

[UIView animateWithDuration:1.0 animations:^{

//3、遍历后面的子控件

for (int i =startIndex; i < self.view.subviews.count; i++) {

UIView \*child = self.view.subviews[i];

CGRect tempF = child.frame;

tempF.origin.y -= krowH + 1;

child.frame = tempF;

}

}];

//3、遍历后面的子控件

// for (int i =startIndex; i < self.view.subviews.count; i++) {

// UIView \*child = self.view.subviews[i];

// CGRect tempF = child.frame;

// tempF.origin.y -= krowH + 1;

// child.frame = tempF;

// }

//4、判断垃圾桶

if (self.view.subviews.count == 1) {

\_removeItem.enabled = NO;

}

}];

}

-(void)iconClick:(UIButton \*)btn

{

//1、先取得按钮的父控件（因为label和btn处在同一个父控件中）

//btn.superview;

//2、取得文本标签

UILabel \*label = (UILabel \*)[btn.superview viewWithTag:kNameTag];

//3、打印

NSLog(@"名字是：%@",label.text);

}

#pragma mark 删除一行

- (IBAction)remove:(UIBarButtonItem \*)sender {

//1、取出最后一个控件

UIView \*last = [self.view.subviews lastObject];

[UIView animateWithDuration:1.0 animations:^{

CGRect tempF = last.frame;

tempF.origin.x = 320;

last.frame = tempF;

last.alpha = 0;

} completion:^(BOOL finished) {

[last removeFromSuperview];

//3、判断剩下的子控件个数

if (self.view.subviews.count == 1) {

\_removeItem.enabled = NO;

}

}];

}

@end

**联系人管理5（创建RowView类，表示xib）**

**RowView.h**

#import <UIKit/UIKit.h>

@interface RowView : UIView

+(id)rowViewWithIcon:(NSString \*)icon name:(NSString \*)name;

@property(nonatomic,weak) IBOutlet UIButton \*iconBtn;

@property(nonatomic,weak) IBOutlet UILabel \*nameLabel;

@end

**RowView.m (可以通过tag找到控件，也可以直接连线)**

#import "RowView.h"

@implementation RowView

+(id)rowViewWithIcon:(NSString \*)icon name:(NSString \*)name

{

RowView \*view = [[NSBundle mainBundle] loadNibNamed:@"RowView" owner:nil options:nil][0];

// //1、设置图标

// UIButton \*iconBtn = (UIButton \*)[view viewWithTag:1];

// [iconBtn setImage:[UIImage imageNamed:icon] forState:UIControlStateNormal];

[view.iconBtn setImage:[UIImage imageNamed:icon] forState:UIControlStateNormal];

//

// //2、设置姓名

// UILabel \*nameLabel = (UILabel \*)[view viewWithTag:2];

// nameLabel.text = name;

view.nameLabel.text = name;

return view;

}

@end

**ViewController.h**

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

//添加

- (IBAction)add:(UIBarButtonItem \*)sender;

//删除

- (IBAction)remove:(UIBarButtonItem \*)sender;

-(IBAction)deleteClick:(UIButton \*)btn;

@property (weak, nonatomic) IBOutlet UIBarButtonItem \*removeItem;

@end

**ViewController.m**

#import "ViewController.h"

#import "RowView.h"

#define krowH 50

#define kNameTag 10

//类扩展(class entension,匿名分类)

@interface ViewController ()

{

NSArray \*\_allNames;

}

@end

@implementation ViewController

#pragma mark 控制器的view加载完毕的时候调用一次

- (void)viewDidLoad {

[super viewDidLoad];

\_allNames = @[@"Mary",@"Jack",@"Rose",@"Jessica",@"Lily",@"Anna"];

}

#pragma mark 添加一行

- (IBAction)add:(UIBarButtonItem \*)sender {

//0、取出最后一个子控件

UIView \*last = [self.view.subviews lastObject];

//这行的Y值=最后一个子控件的Y值 + 最后一个子控件的高度 + 间距

CGFloat rowY = last.frame.origin.y + last.frame.size.height + 1;

//1、创建一行

// UIView \*rowView = [[UIView alloc] init];

// rowView.backgroundColor = [UIColor redColor];

UIView \*rowView = [self creatRowView];

//2、添加一行到控制器的view中

[self.view addSubview:rowView];

//3、让删除Item有效

\_removeItem.enabled = YES;

//4、执行动画

rowView.frame = CGRectMake(320, rowY, 320, krowH);

rowView.alpha = 0;

[UIView animateWithDuration:1.0 animations:^{

rowView.frame = CGRectMake(0, rowY, 320, krowH);

rowView.alpha = 1;

}];

}

#pragma mark 创建一行（从xib中加载一行的view）

-(UIView \*)creatRowView

{

NSString \*icon = [NSString stringWithFormat:@"01%d.png",arc4random\_uniform(9)];

NSString \*name = \_allNames[arc4random\_uniform(\_allNames.count)];

RowView \*rowView = [RowView rowViewWithIcon:icon name:name];

return rowView;

}

#pragma mark 监听删除按钮点击

-(void)deleteClick:(UIButton \*)btn

{

[UIView animateWithDuration:1.0 animations:^{

CGRect tempF = btn.superview.frame;

tempF.origin.x = 320;

btn.superview.frame = tempF;

btn.superview.alpha = 0;

} completion:^(BOOL finished) {

//1、获得即将删除的这行在数组中的位置

int startIndex = [self.view.subviews indexOfObject:btn.superview];

//2、删除当前行

[btn.superview removeFromSuperview];

[UIView animateWithDuration:1.0 animations:^{

//3、遍历后面的子控件

for (int i =startIndex; i < self.view.subviews.count; i++) {

UIView \*child = self.view.subviews[i];

CGRect tempF = child.frame;

tempF.origin.y -= krowH + 1;

child.frame = tempF;

}

}];

//3、遍历后面的子控件

// for (int i =startIndex; i < self.view.subviews.count; i++) {

// UIView \*child = self.view.subviews[i];

// CGRect tempF = child.frame;

// tempF.origin.y -= krowH + 1;

// child.frame = tempF;

// }

//4、判断垃圾桶

if (self.view.subviews.count == 1) {

\_removeItem.enabled = NO;

}

}];

}

-(void)iconClick:(UIButton \*)btn

{

//1、先取得按钮的父控件（因为label和btn处在同一个父控件中）

//btn.superview;

//2、取得文本标签

UILabel \*label = (UILabel \*)[btn.superview viewWithTag:kNameTag];

//3、打印

NSLog(@"名字是：%@",label.text);

}

#pragma mark 删除一行

- (IBAction)remove:(UIBarButtonItem \*)sender {

//1、取出最后一个控件

UIView \*last = [self.view.subviews lastObject];

[UIView animateWithDuration:1.0 animations:^{

CGRect tempF = last.frame;

tempF.origin.x = 320;

last.frame = tempF;

last.alpha = 0;

} completion:^(BOOL finished) {

[last removeFromSuperview];

//3、判断剩下的子控件个数

if (self.view.subviews.count == 1) {

\_removeItem.enabled = NO;

}

}];

}

@end

**多组数据展示**

**ViewController.m**

#import "ViewController.h"

@interface ViewController ()<UITableViewDataSource>

{

NSArray \*\_gdcities;

NSArray \*\_hncities;

NSArray \*\_hbcities;

}

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

//1、添加tableView

UITableView \*tableView = [[UITableView alloc] initWithFrame:self.view.bounds style:UITableViewStyleGrouped];

tableView.dataSource = self;

//2、初始化数据

//广东

\_gdcities = @[@"广州",@"深圳",@"梅州",@"东莞"];

\_hncities = @[@"长沙",@"益阳",@"岳阳"];

\_hbcities = @[@"武汉",@"黄冈"];

[self.view addSubview:tableView];

}

#pragma mark - 数据源方法

#pragma mark 一共有多少组

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

{

return 3;

}

#pragma mark 第section组有多少行

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

{

if(section == 0){

return \_gdcities.count;

}else if(section == 1){

return \_hncities.count;

}else{

return \_hbcities.count;

}

}

#pragma mark 返回每一行显示的内容（每行显示怎样的cell）

-(UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

{

UITableViewCell \*cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:nil];

//indexPath标识唯一的一行，第section组的第row行

NSString \*text = nil;

if (indexPath.section ==0) {

text = \_gdcities[indexPath.row];

}else if(indexPath.section == 1){

text = \_hncities[indexPath.row];

}else{

text = \_hbcities[indexPath.row];

}

cell.textLabel.text = text;

//cell.textLabel.text = @"678";

//NSLog(@"section = %d,row = %d",indexPath.section,indexPath.row);

return cell;

}

@end