**联系人管理2**

**ViewController.h**

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

//添加

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

//删除

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

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

@end

**ViewController.m**

#import "ViewController.h"

#define krowH 50

#define kNameTag 10

//类扩展(class extension,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

-(UIView \*)creatRowView

{

//1、创建红色的父控件

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

rowView.backgroundColor = [UIColor redColor];

//2、添加名字（标签）

UILabel \*name = [[UILabel alloc] init];

name.frame = CGRectMake(0, 0, 320, krowH);

int nameIndex = arc4random\_uniform(\_allNames.count);

name.text = \_allNames[nameIndex];

//name.center = CGPointMake(160, krowH \* 0.5);

name.backgroundColor = [UIColor clearColor];

//设置文字居中

name.textAlignment = NSTextAlignmentCenter;

name.tag = kNameTag;

[rowView addSubview:name];

//3、添加头像

//UIButton \*icon = [[UIButton alloc] init];

UIButton \*icon = [UIButton buttonWithType:UIButtonTypeCustom];

icon.frame = CGRectMake(20, 0, 50, krowH);

//随机产生文件名

//int randowIndex = arc4random() % 9;

int randomIndex = arc4random\_uniform(9);

NSString \*iconName = [NSString stringWithFormat:@"01%d.png",randomIndex];

//设置图片

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

//添加监听器

[icon addTarget:self action:@selector(iconClick:) forControlEvents:UIControlEventTouchUpInside];

[rowView addSubview:icon];

//删除按钮

UIButton \*delete = [UIButton buttonWithType:UIButtonTypeRoundedRect];

delete .frame = CGRectMake(250, 5, 60, 40);

[delete setTitle:@"删除" forState:UIControlStateNormal];

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

[rowView addSubview:delete];

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3（使用xib）**

**ViewController.h**

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

//添加

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

//删除

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

@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:nil 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