

# iOSアプリ作成

## —電卓—



佐野 尚之

# 本ドキュメントのライセンスについて

この作品は、クリエイティブ・コモンズのAttribution 3.0 Unportedライセンスの下でライセンスされています。

この使用許諾条件を見るには、<http://creativecommons.org/licenses/by/3.0/>をチェックするか、クリエイティブ・コモンズに郵便にてお問い合わせください。

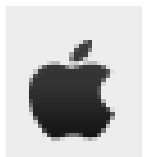
住所は：171 Second Street, Suite 300, San Francisco, California 94105, USA です。



原作者のクレジット（氏名、作品タイトルとURL）を表示することを守れば、改変はもちろん、営利目的での二次利用も許可される最も自由度の高いCCライセンス。

<http://creativecommons.jp/>





# 変更履歴

## ■第1版

2012/1/15

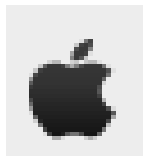
このドキュメントは、オープンソースの「LibreOffice 3.4.4」を使用して作成。



# アジェンダ

- ・動作確認環境 . . . . . 5
- ・作業フォルダについて . . . . . 6
- ・電卓の機能について . . . . . 7
- ・電卓作成 . . . . . 8





# 動作確認環境

## ■ OS

Mac OS X Lion 10.7.2

## ■ PC

Mac mini Mid 2011 (2.3GHz Intel Core i5、メモリ2GB)

## ■ 開発環境

iOS SDK 4.3

Xcode 3.2.6



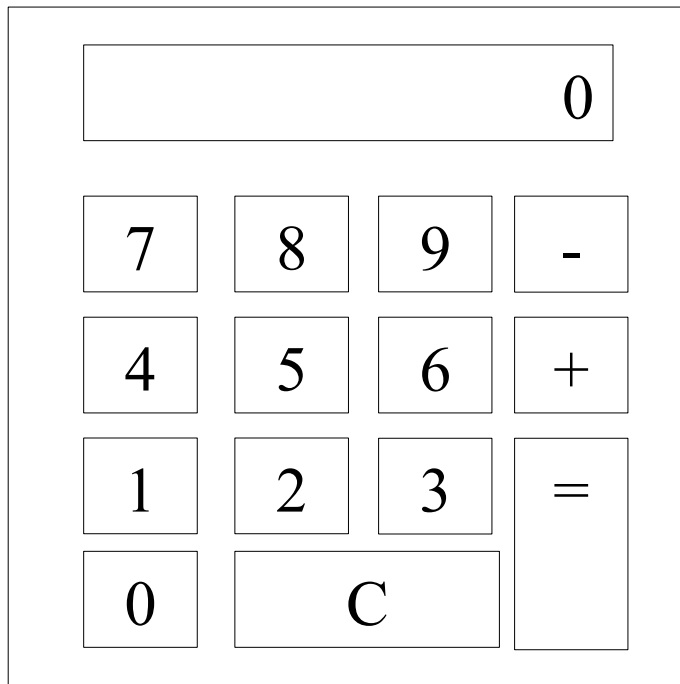
## 作業フォルダについて

本ドキュメントの指定通りの場所ではなくても問題はありません。



# 電卓の機能について

- ・ 足し算、引き算、計算結果、クリアのみを実装する。
- ・ 数字は、0～9までのボタンを用意する。
- ・ 画面イメージは以下の通りとする。

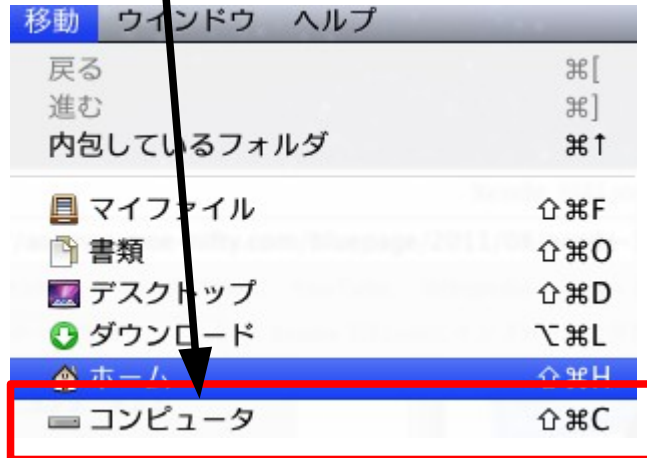


以下のサイトの「アプリ：電卓を作る」を見て、本当に動くものが作れるのかの確認のなので、ソースコードについての説明は以下のサイトを参照してください。  
[http://iphone.moo.jp/app/?page\\_id=2](http://iphone.moo.jp/app/?page_id=2)



## 電卓作成(1/24)

- ・「Finder」の「コンピュータ」を選択します。

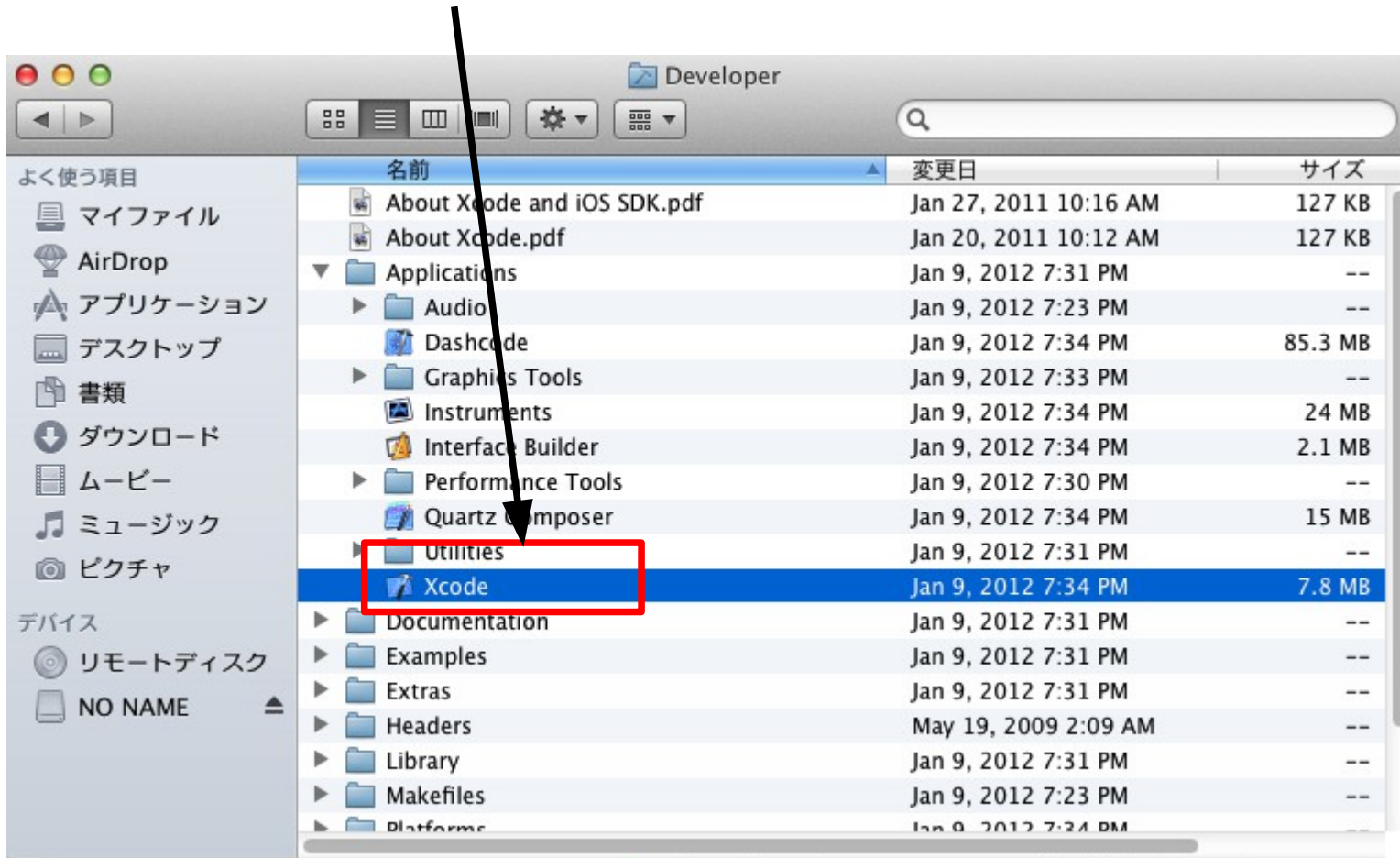






## 電卓作成(2/24)

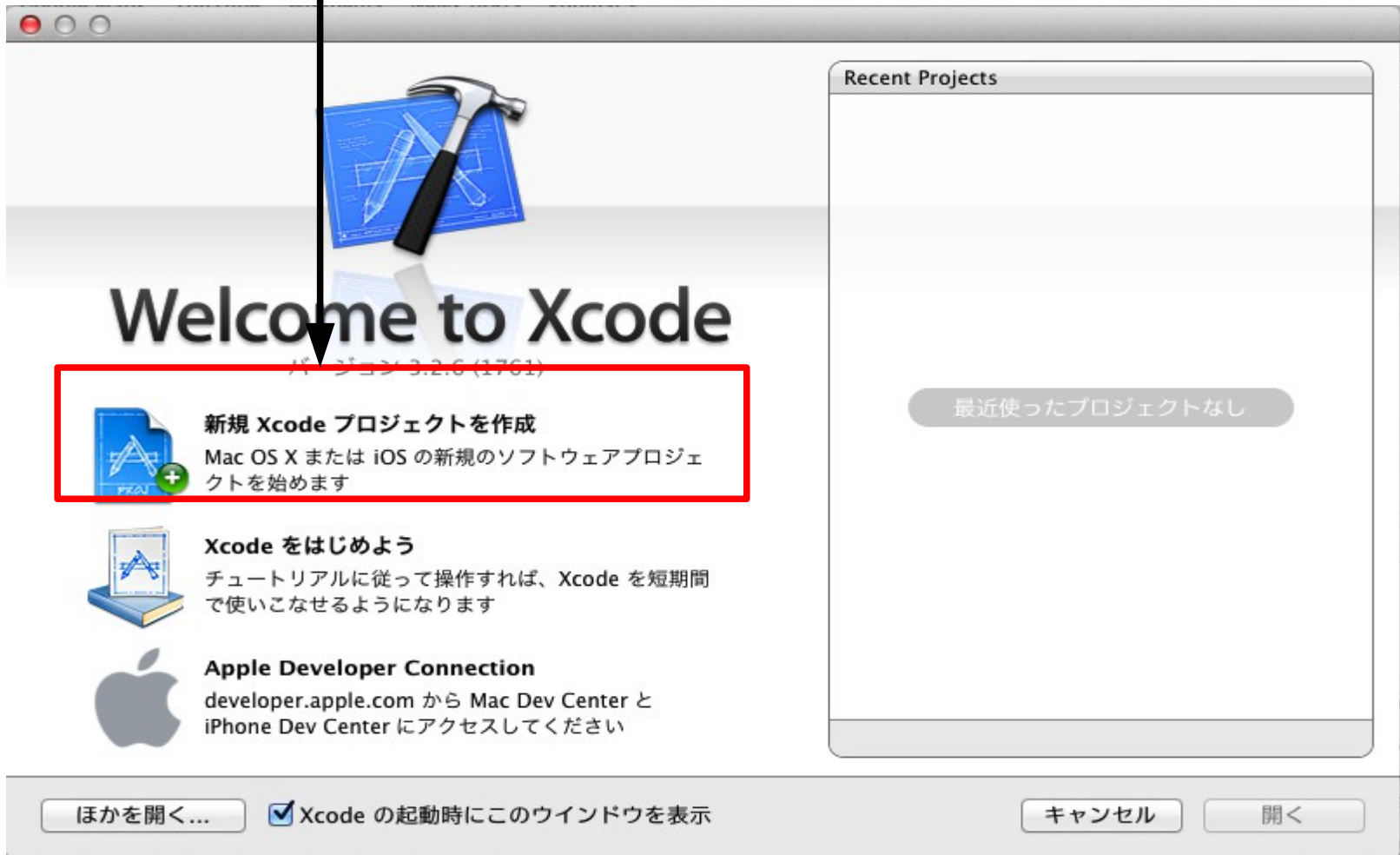
- 「Macintosh HD」－「Developer」－「Applications」－「Xcode」をダブルクリックします。





## 電卓作成(3/24)

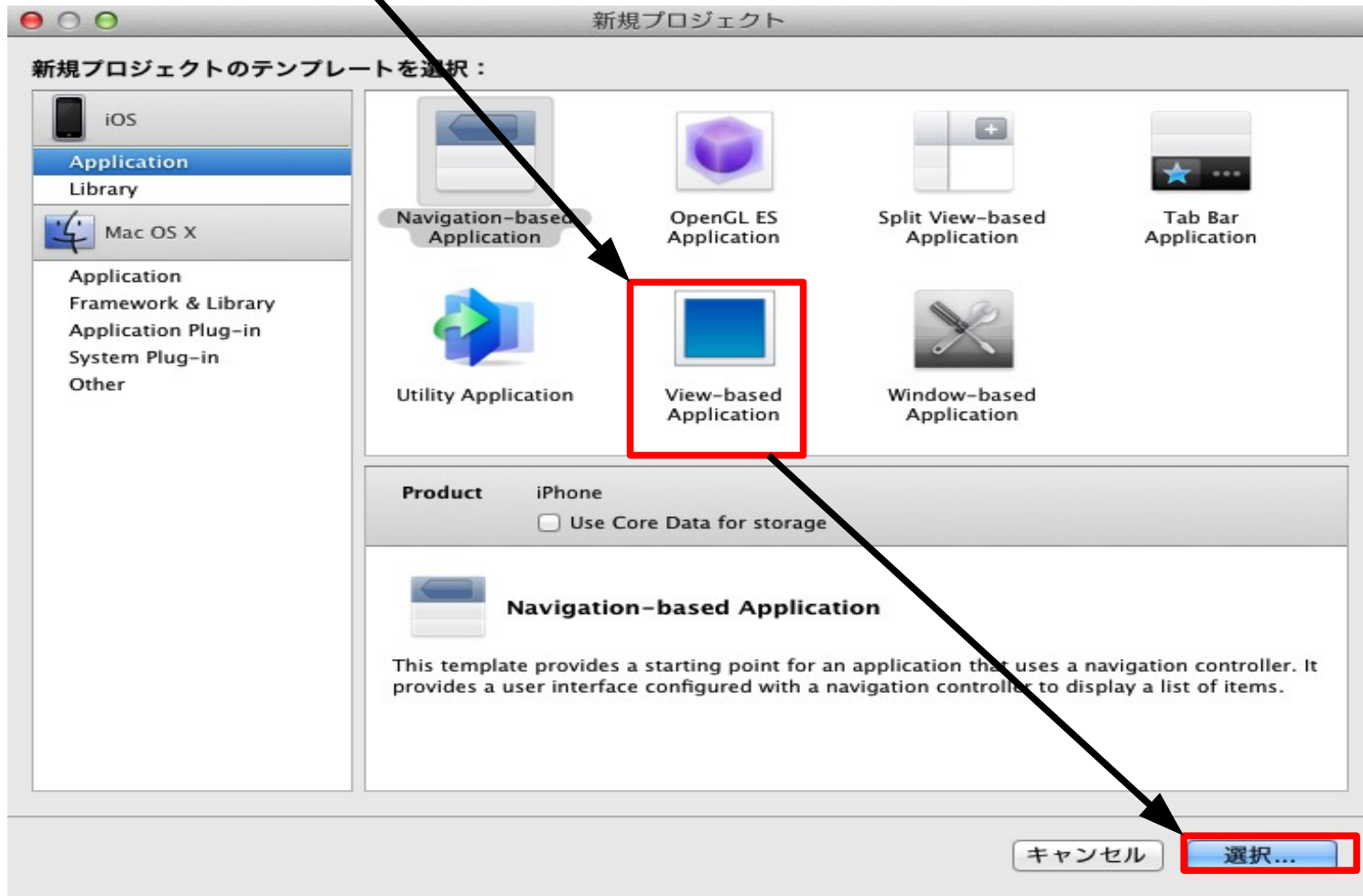
- ・「新規 Xcode プロジェクトを作成」をクリックします。





## 電卓作成(4/24)

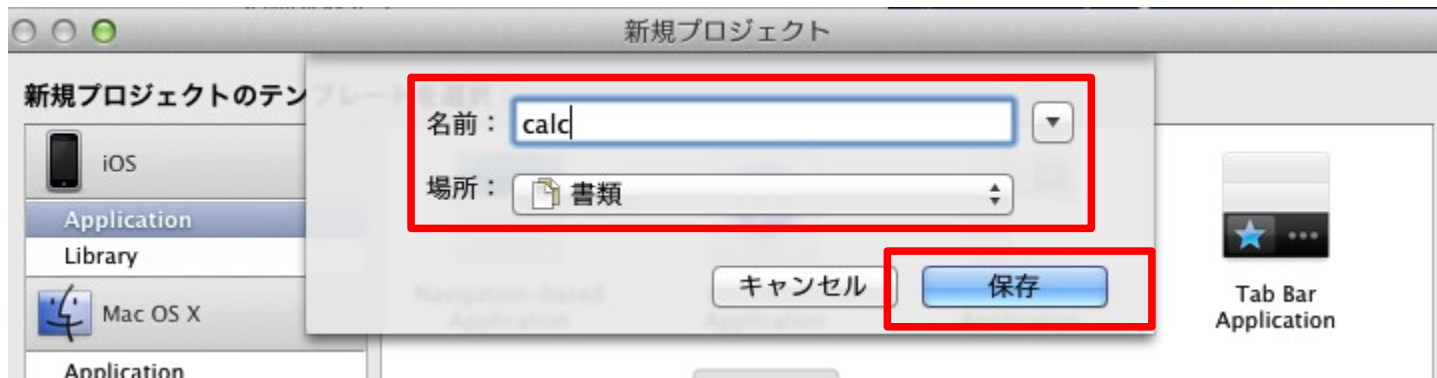
- ・「View-based Application」を選択し、「選択」ボタンをクリックします。





## 電卓作成(5/24)

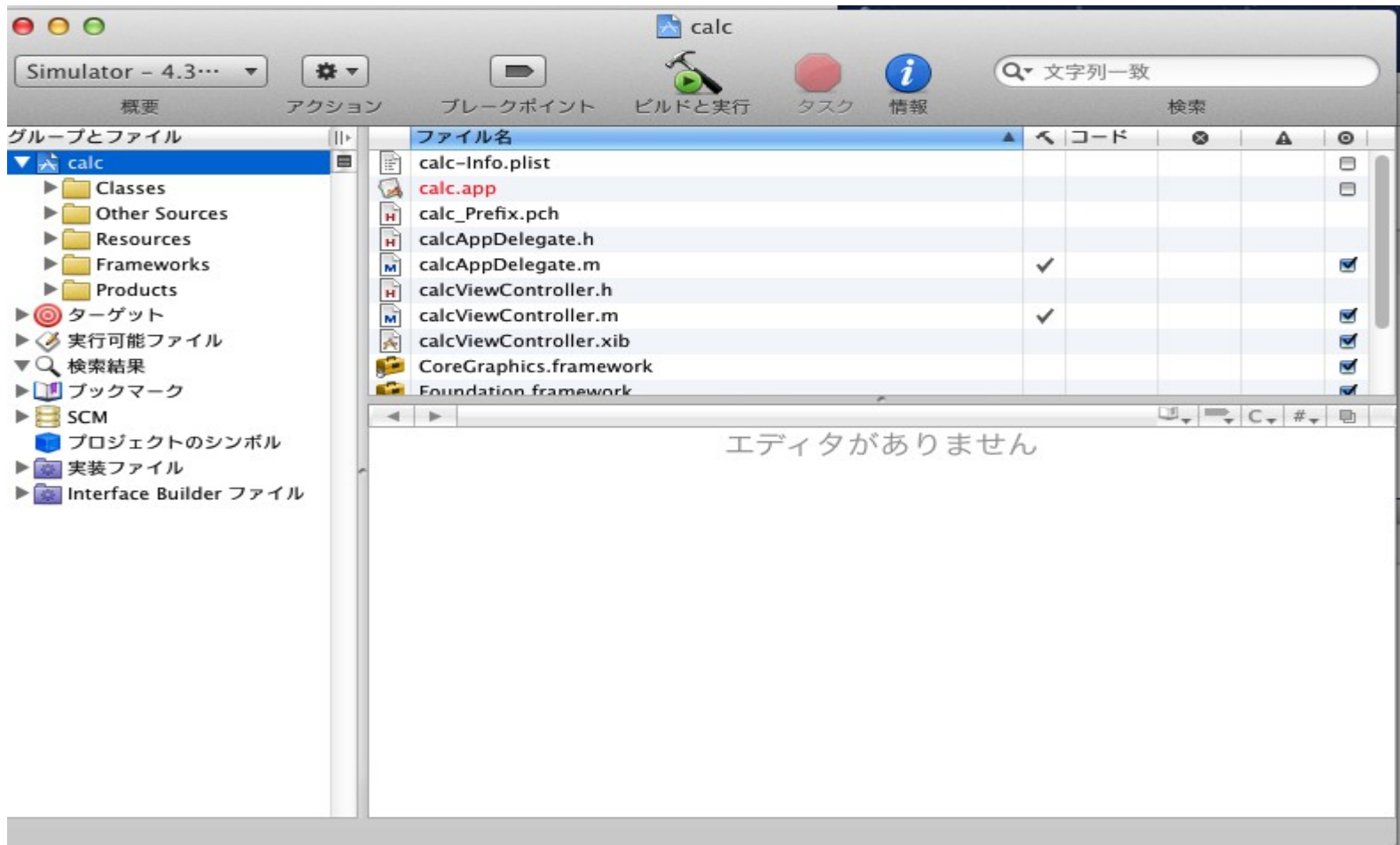
- ・保存先を聞かれるので名前と場所を指定して、「保存」ボタンをクリックしてください。





## 電卓作成(6/24)

- ・以下の画面が表示します。





## 電卓作成(7/24)

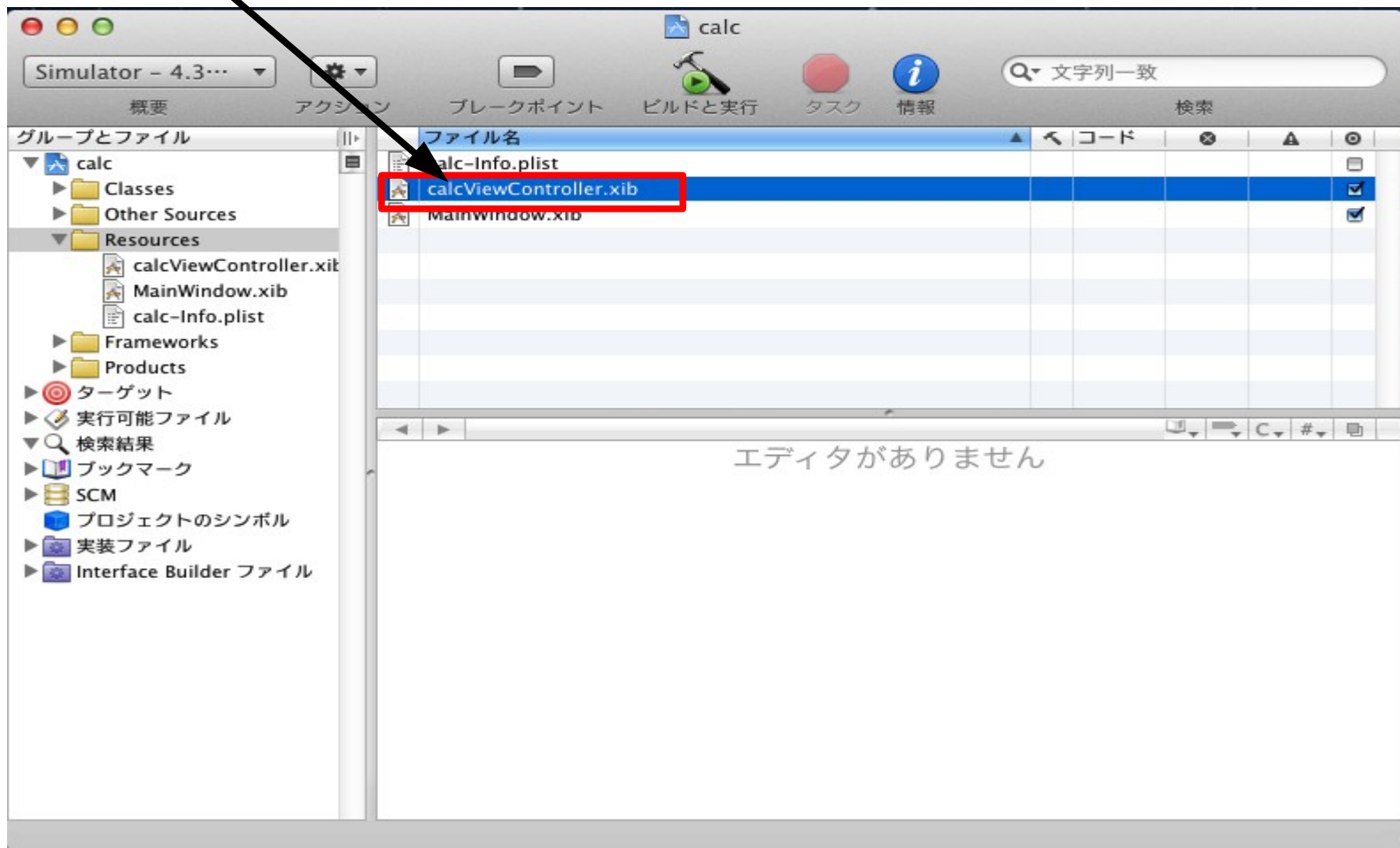
- ・ 「Simulator - 4.3」 のプルダウン選択し、「calc - iPhone Simulator 4.3」 を選択します。





## 電卓作成(8/24)

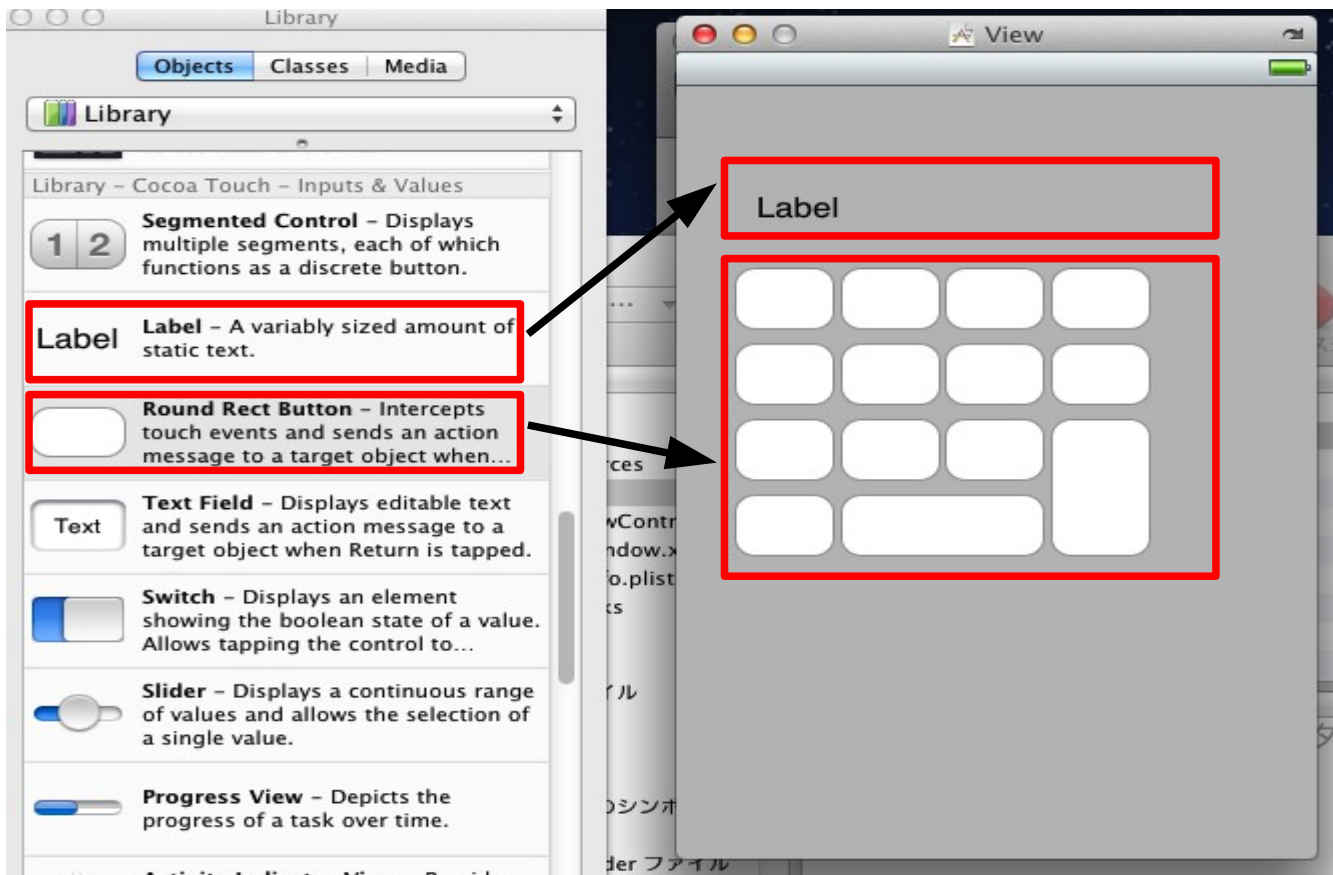
- ・「Resources」フォルダの「CalcViewController.xib」をダブルクリックします。





## 電卓作成(9/24)

- LabelとButtonを以下のように配置します。

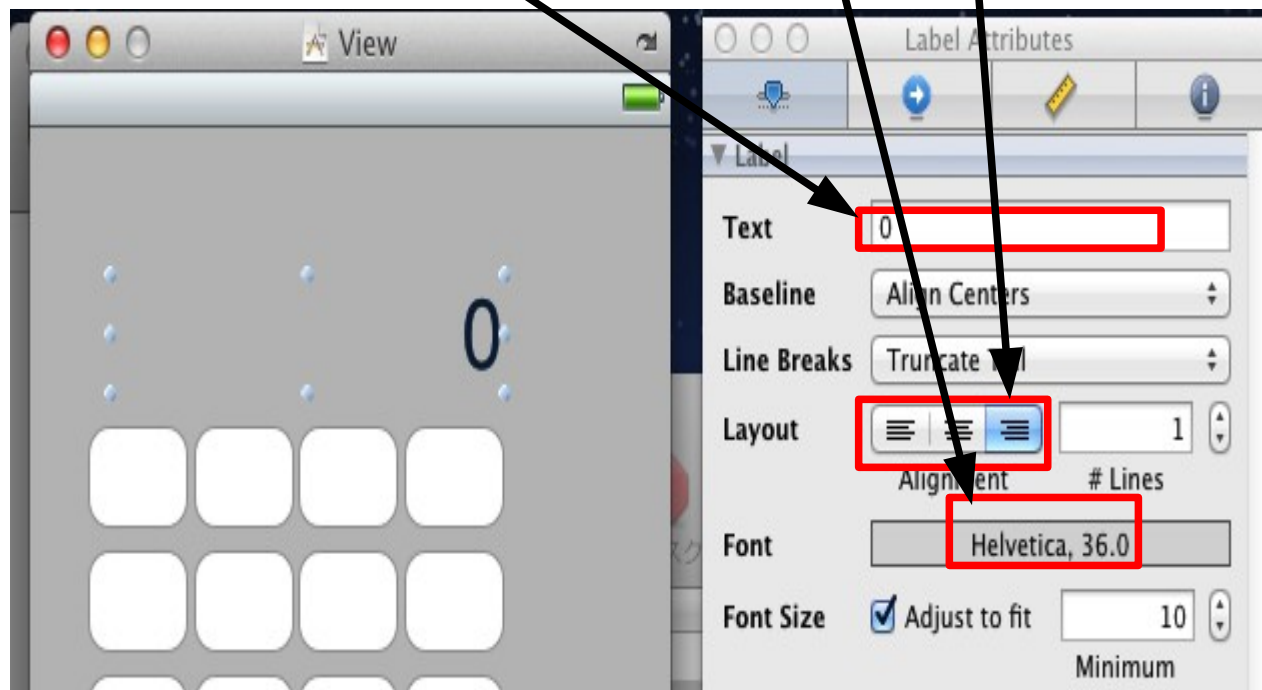






## 電卓作成(10/24)

- ・ LabelのTextプロパティを「0」、フォントサイズを「36」、Layoutを変更します。

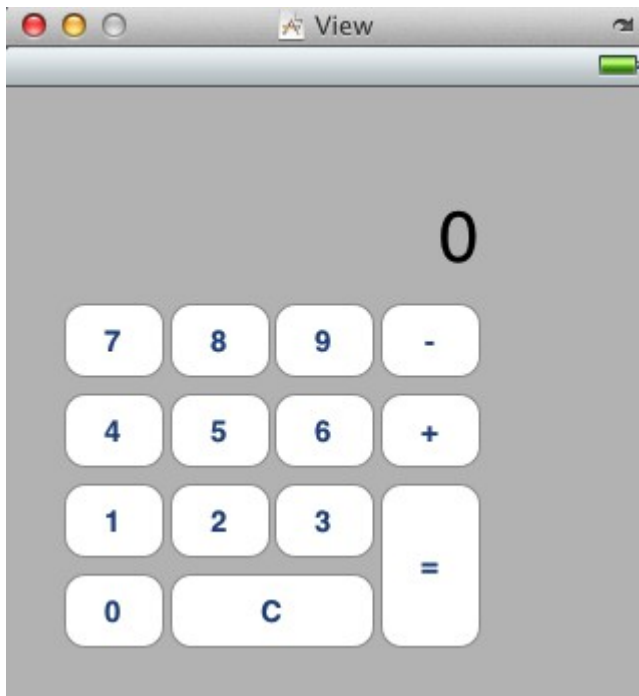




## 電卓作成(11/24)

・各ボタンの設定を以下の内容に変更します。

7 : Titleを「7」、Tagを「7」。8 : Titleを「8」、Tagを「8」。9 : Titleを「9」、Tagを「9」。4 : Titleを「4」、Tagを「4」。5 : Titleを「5」、Tagを「5」。6 : Titleを「6」、Tagを「6」。1 : Titleを「1」、Tagを「1」。2 : Titleを「2」、Tagを「2」。3 : Titleを「3」、Tagを「3」。0 : Titleを「0」、Tagを「0」。  
- : Titleを「-」、Tagを「1」。+ : Titleを「+」、Tagを「0」。= : Titleを「=」。C : Titleを「C」。





## 電卓作成(12/24)

- 「calcViewController.h」に以下のコードを追加します。保存も行ってください。

```
calcViewController.h — calc
概要 アクション ブレークポイント ビルドと実行 タスク 情報
文字列一致 検索
グループとファイル
calc
  Classes
    calcAppDelegate.h
    calcAppDelegate.m
    calcViewController.h
    calcViewController.m
  Other Sources
  Resources
    calcViewController.xib
    MainWindow.xib
    calc-Info.plist
  Frameworks
  Products
  ターゲット
  実行可能ファイル
  検索結果
  ブックマーク
  SCM
  プロジェクトのシンボル
  実装ファイル
  Interface Builder ファイル
ファイル名
calcViewController.h
code
calcViewController.h:21 @interface calcViewController : UIViewController {
    IBOutlet UILabel *label;
    BOOL startInput;
    int currentValue;
    int operation;
}
-(IBAction)numberButtonPressed:(id)sender;
-(IBAction)clearButtonPressed:(id)sender;
-(IBAction)equalButtonPressed:(id)sender;
-(IBAction)opButtonPressed:(id)sender;
@end
```



## 電卓作成(13/24)

- 「calcViewControler.rm」を以下のように変更します。保存も行ってください。

```
calcViewControler.m:87 > @implementation calcViewController
// Implement viewDidLoad to do additional setup after loading the view, typically
- (void)viewDidLoad {
    [super viewDidLoad];
    startInput = YES;
    currentValue = 0;
}

- (IBAction)numberButtonPressed:(id)sender
{
    UIButton *b = (UIButton *)sender;

    if( startInput ){
        // 最初の1桁目が0なら表示しない
        if( b.tag == 0 ) return;
        // 新しく表示する文字列を作成
        label.text = [NSString stringWithFormat:@"%d", b.tag];
        startInput = NO;
    } else {
        // すでに表示されている文字列に連結
        label.text = [NSString stringWithFormat:@"%s%d", label.text, b.tag];
    }
}

- (IBAction)equalButtonPressed:(id)sender
{
    // 直前に押された演算子で場合分け
    if( operation == 0 ){
        currentValue += [label.text intValue];
    } else if( operation == 1 ){
        currentValue -= [label.text intValue];
    }

    // 表示の更新
    label.text = [NSString stringWithFormat:@"%d", currentValue];
    startInput = YES;
}

- (IBAction)opButtonPressed:(id)sender
{
    UIButton *b = (UIButton *)sender;

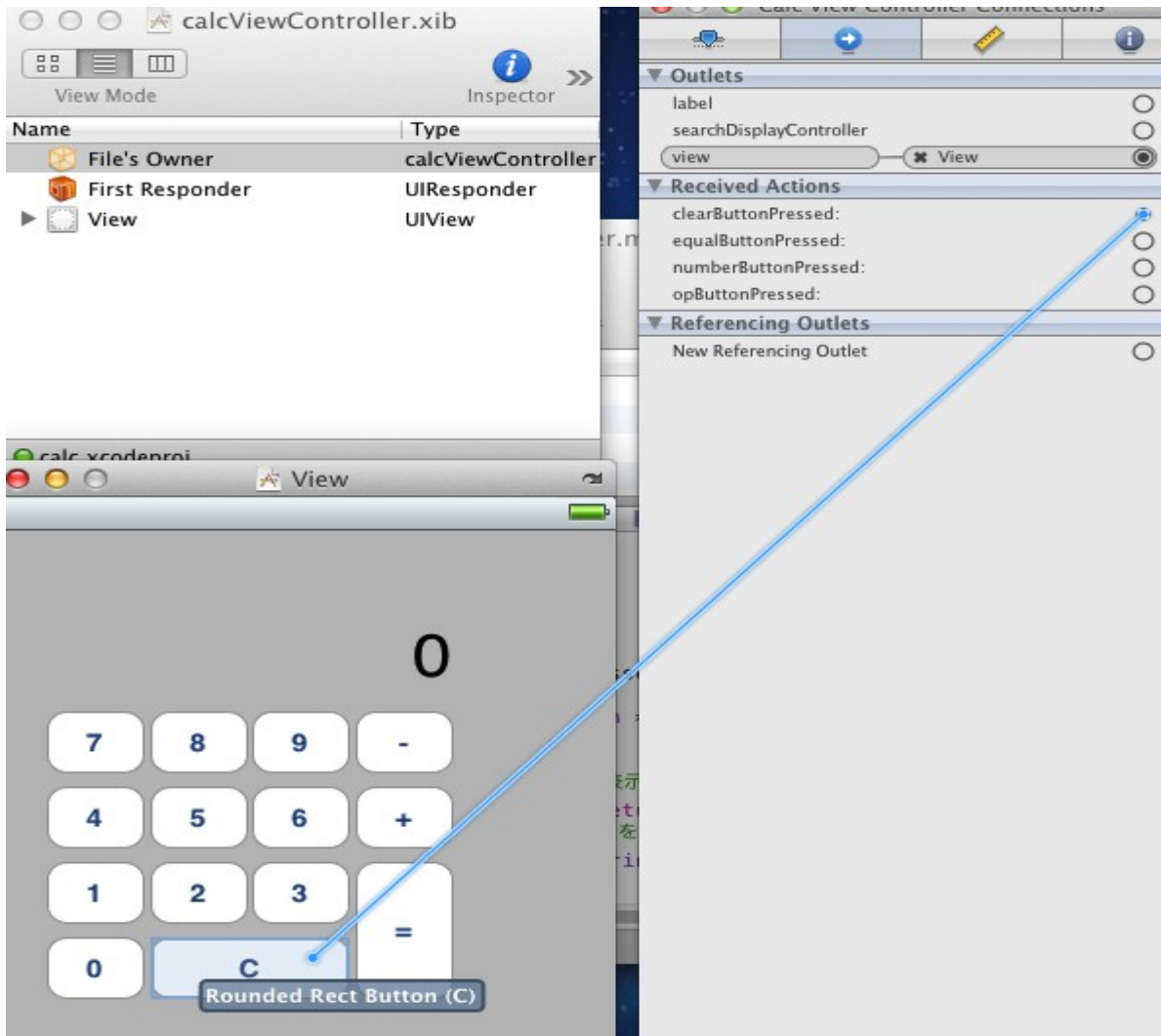
    // 現在値の保存
    currentValue = [label.text intValue];

    // 演算の保存
    operation = b.tag;
    startInput = YES;
}
```



## 電卓作成(14/24)

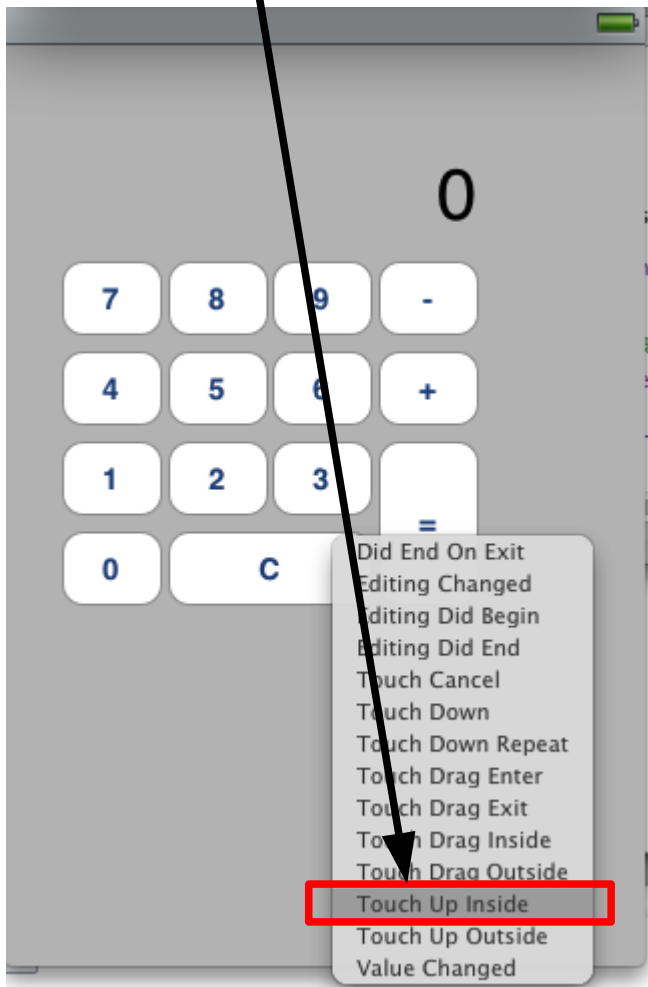
- 「File's Owner」を選択—「clearButtonPressed」の○をドラッグして「C」ボタンを上まで持っていく。





## 電卓作成(15/24)

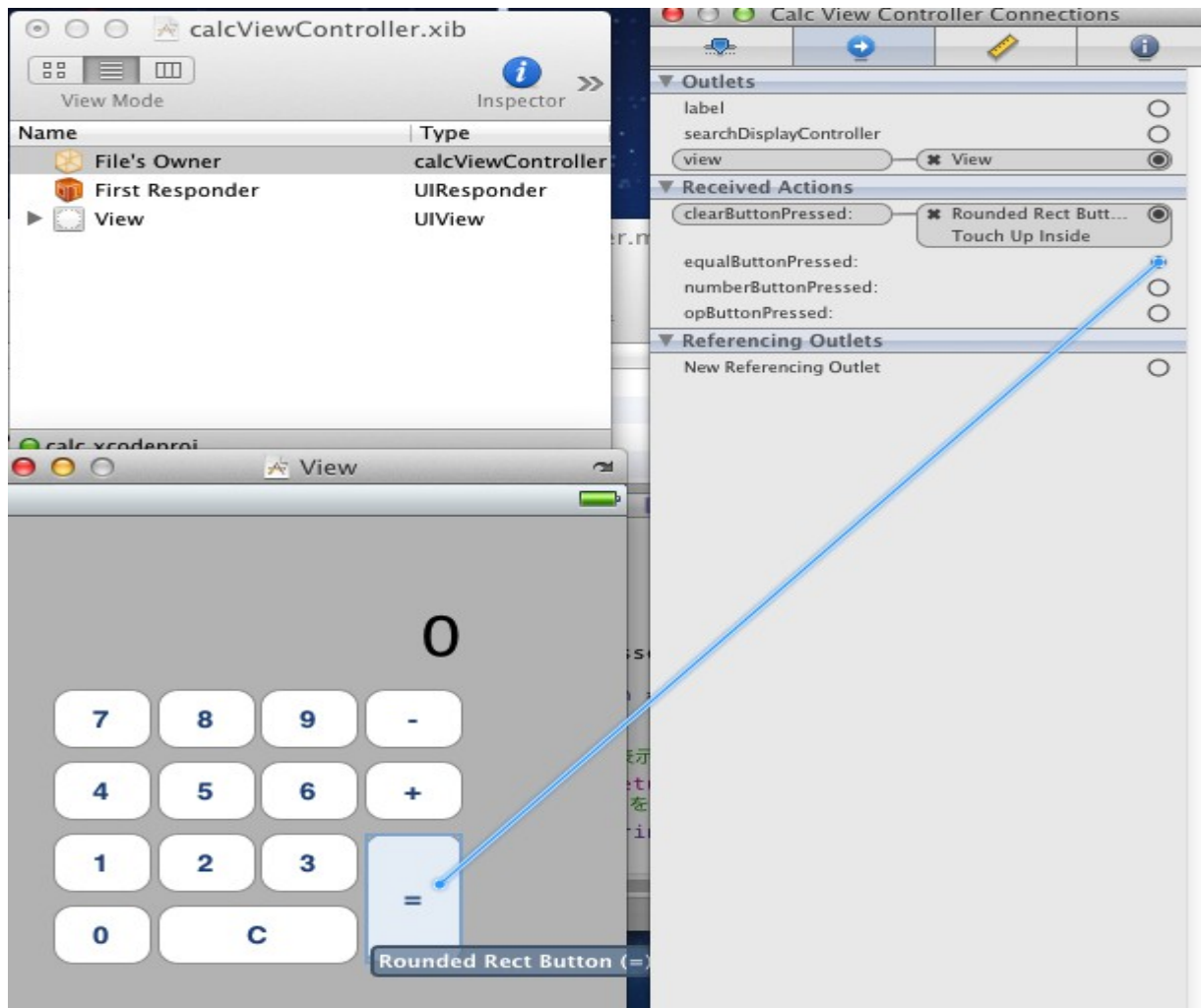
- ・「Touch Up Inside」を選択します。





## 電卓作成(16/24)

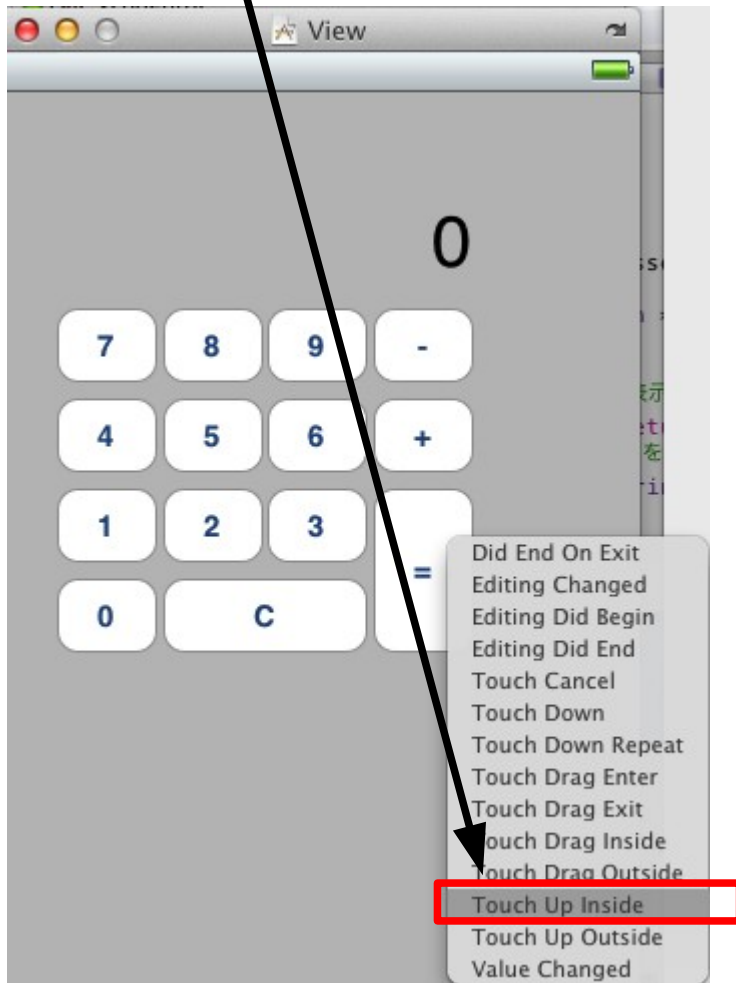
- 「File's Owner」を選択—「equalsButtonPressed」の○をドラッグして「=」ボタンを上まで持っていく。





## 電卓作成(17/24)

- ・「Touch Up Inside」を選択します。

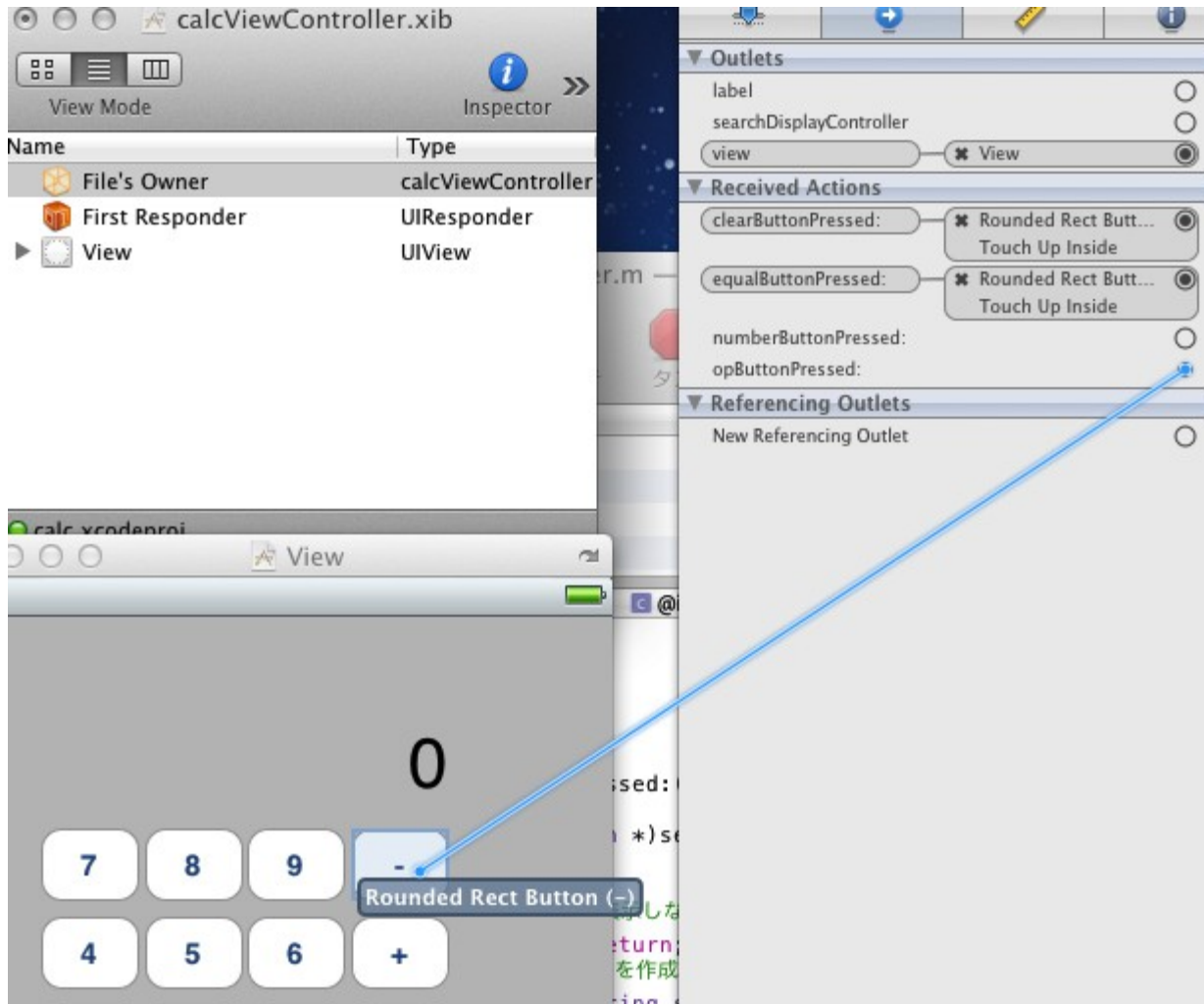






## 電卓作成(18/24)

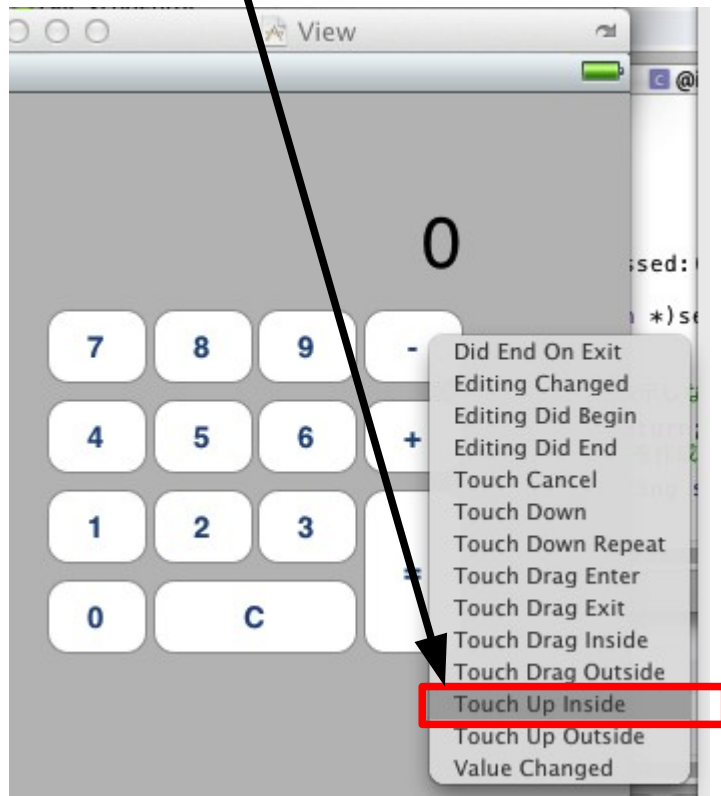
- 「File's Owner」を選択し「opButtonPressed」の○をドラッグして「-」ボタンを上まで持っていく。





## 電卓作成(19/24)

- ・「Touch Up Inside」を選択します。

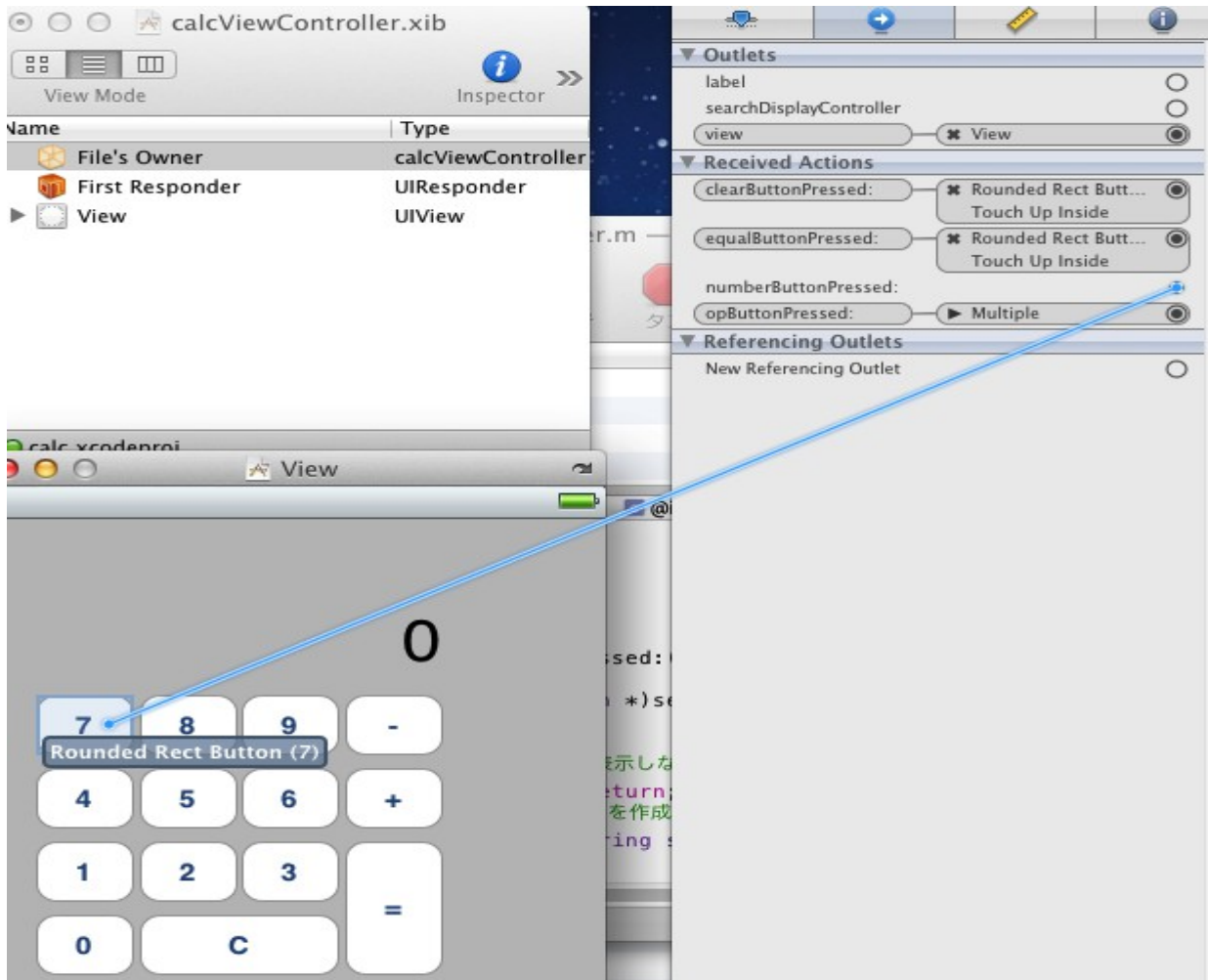


- ・「+」ボタンも同様の操作を実施する必要あり。画面操作の説明は省略します。



## 電卓作成(20/24)

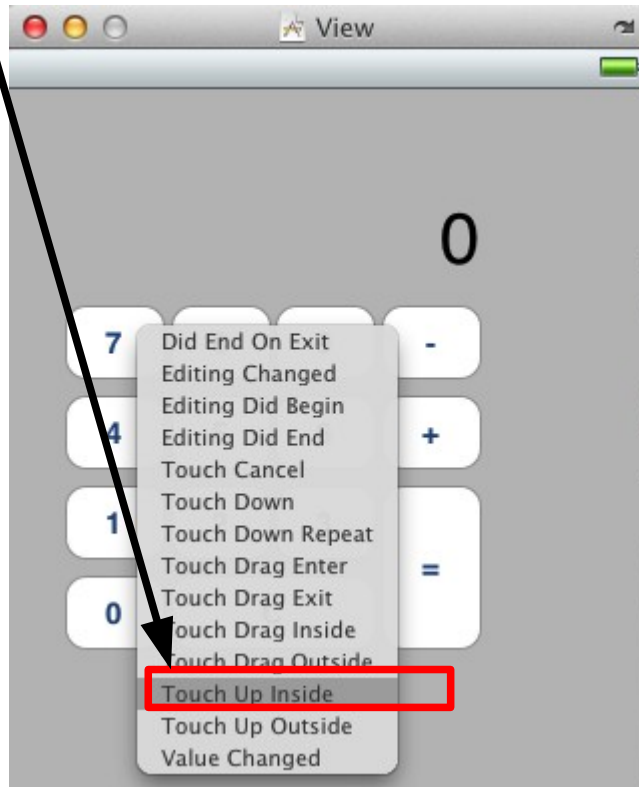
- 「File's Owner」を選択—「numberButtonPressed」の○をドラッグして「7」ボタンを上まで持っていく。





## 電卓作成(21/24)

- ・ 「Touch Up Inside」 を選択します。

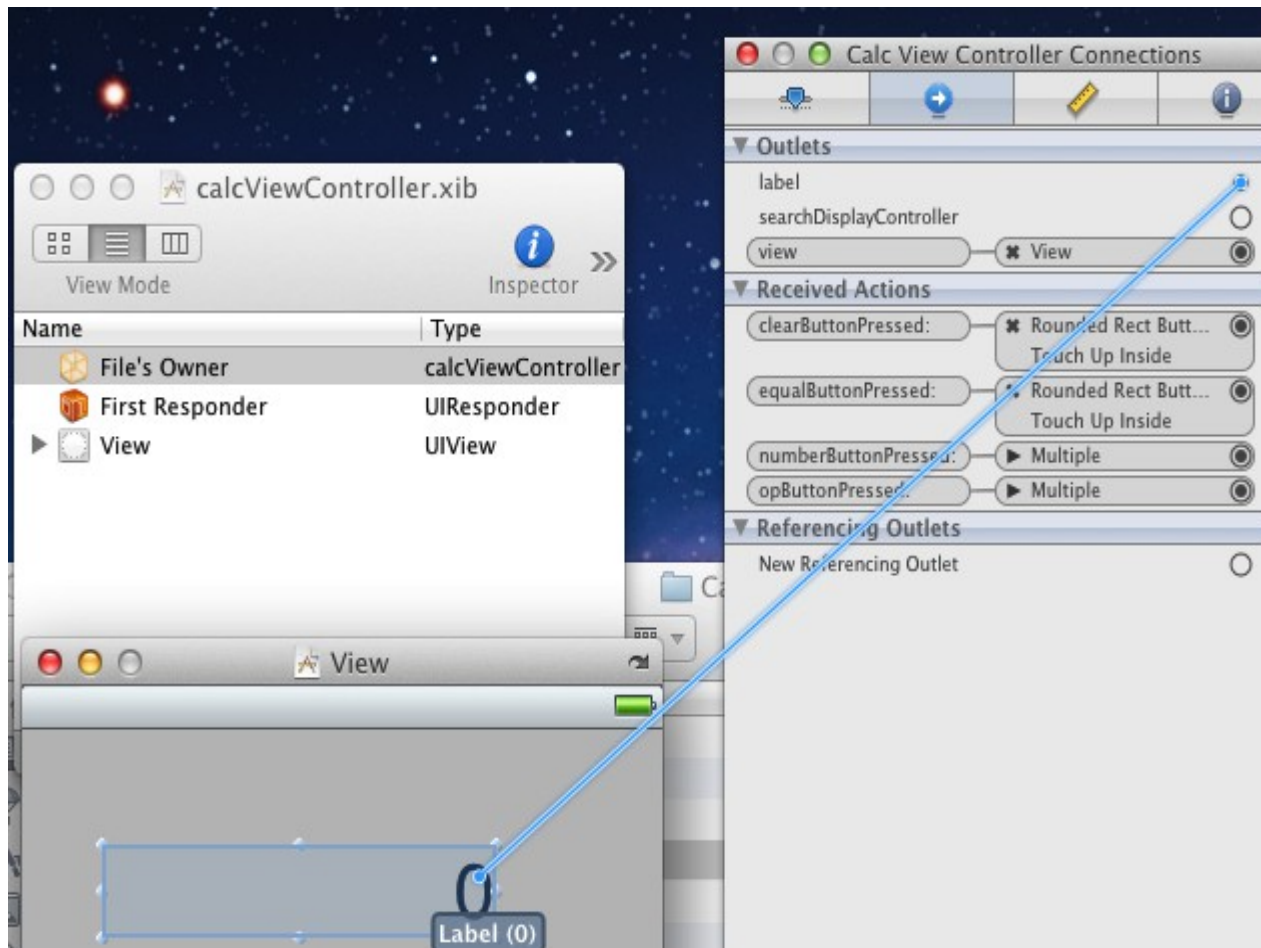


- ・ 8, 9, 4, 5, 6, 1, 2, 3, 0ボタンも同様の操作を実施する必要あり。画面操作の説明は省略します。



## 電卓作成(22/24)

- 「File's Owner」を選択ー「Label」の○をドラッグしてLabelを上まで持っていく。





## 電卓作成(23/24)

- ・「ビルドと実行」をクリックします。





## 電卓作成(24/24)

- ・以下の画面が出れば成功です。

