



ИВЭЙ НИ (YIWEI NI)

OpenCV и Swift: шаг за шагом

МАЛЕНЬКИЙ IOS-ПРОЕКТ С ИСПОЛЬЗОВАНИЕМ
БИБЛИОТЕКИ МАШИННОГО ЗРЕНИЯ OPENCV И
ЯЗЫКА SWIFT



Перевод С. Кузнецов, 2018 г.

OpenCV with Swift - step by step

Get hands-on with practical Computer Vision using OpenCV 3

Yiwei Ni

Article, 2017:

<https://medium.com/@yiweini/opencv-with-swift-step-by-step-c3cc1d1ee5f1>

OpenCV и Swift: шаг за шагом

Маленький iOS-проект с использованием библиотеки машинного зрения OpenCV и языка Swift

Ивэй Ни

Статья, 2017 г.:

<https://medium.com/@yiweini/opencv-with-swift-step-by-step-c3cc1d1ee5f1>

Перевод: С. Кузнецов, 2018 г.





OpenCV с помощью Swift: шаг за шагом

Маленький iOS-проект с использованием библиотеки машинного зрения OpenCV и языка Swift.

Несколько недель назад во время моей летней интернатуры, мой супервизор попросил, чтобы я включил библиотеку [OpenCV](#) в существующий [iOS](#)-проект. В то время это задание прозвучало для меня настолько простым. Однако после чтения каждого ответа из результатов [Google](#)-поиска на шести страницах и попробовав каждый возможный пример, я все еще не смог написать что-либо:(

Каким образом другие люди могут сделать это, если я получаю эти странные ошибки и предупреждения на всем протяжении смены экранов?

Хорошо. После часов испытаний я наконец понял все, и поэтому для всех людей, которые хотят написать [OpenCV](#)-проекты в [iOS](#), я привожу описание того, как я сделал это. Я пытаюсь быть супер детализированным, поэтому не стесняйтесь пропускать знакомые вам шаги действий.

Установка

1. Загрузите [opencv2.framework](#). Я использую текущую новейшую версию [3.2.0](#) для [iOS](#).

2. Создайте новый **Xcode**-проект. Я использую интегрированную **IDE**-среду разработки **Xcode 8.3.3**.

×



Welcome to Xcode

Version 8.3.3 (8E3004b)



Get started with a playground

Explore new ideas quickly and easily.



Create a new Xcode project

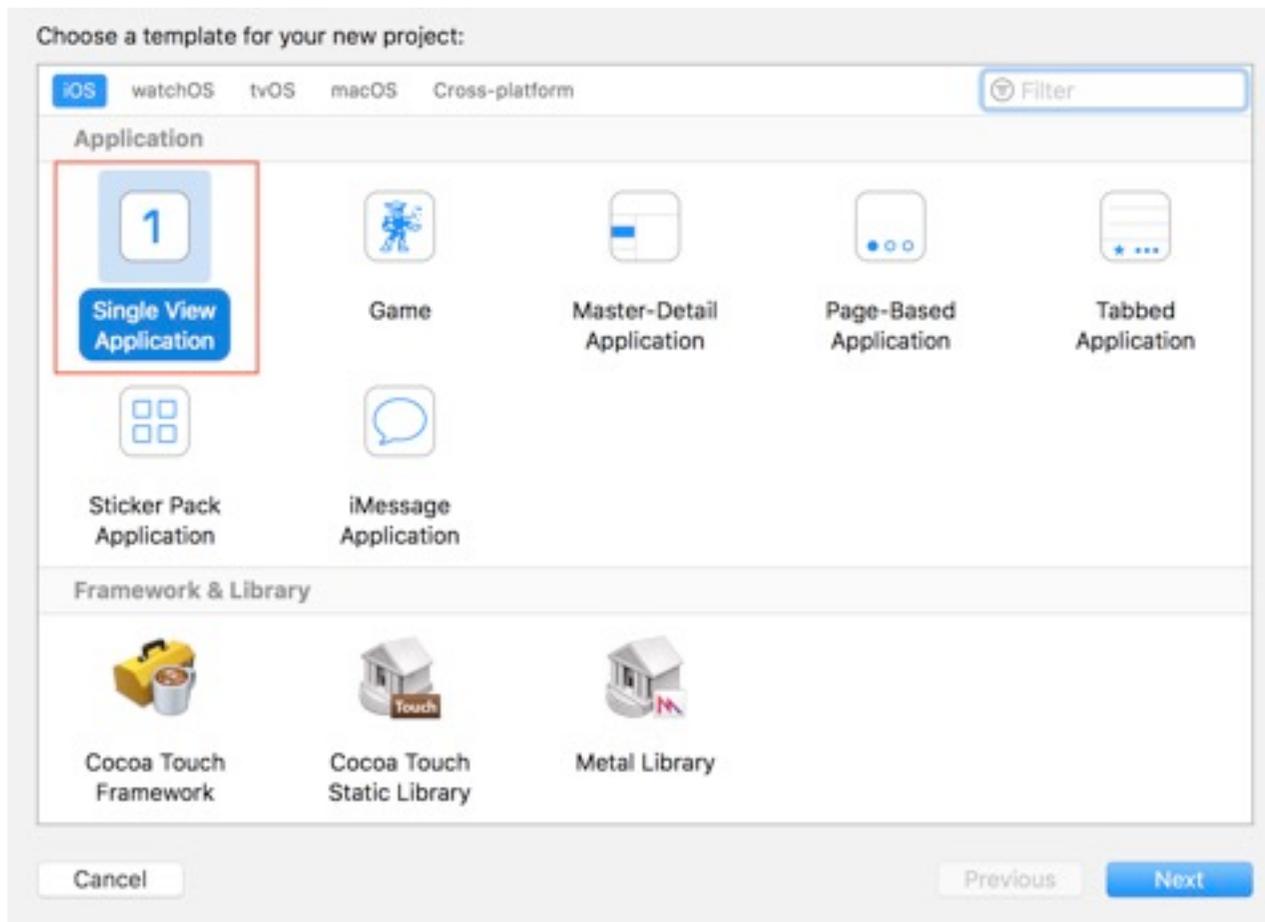
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.



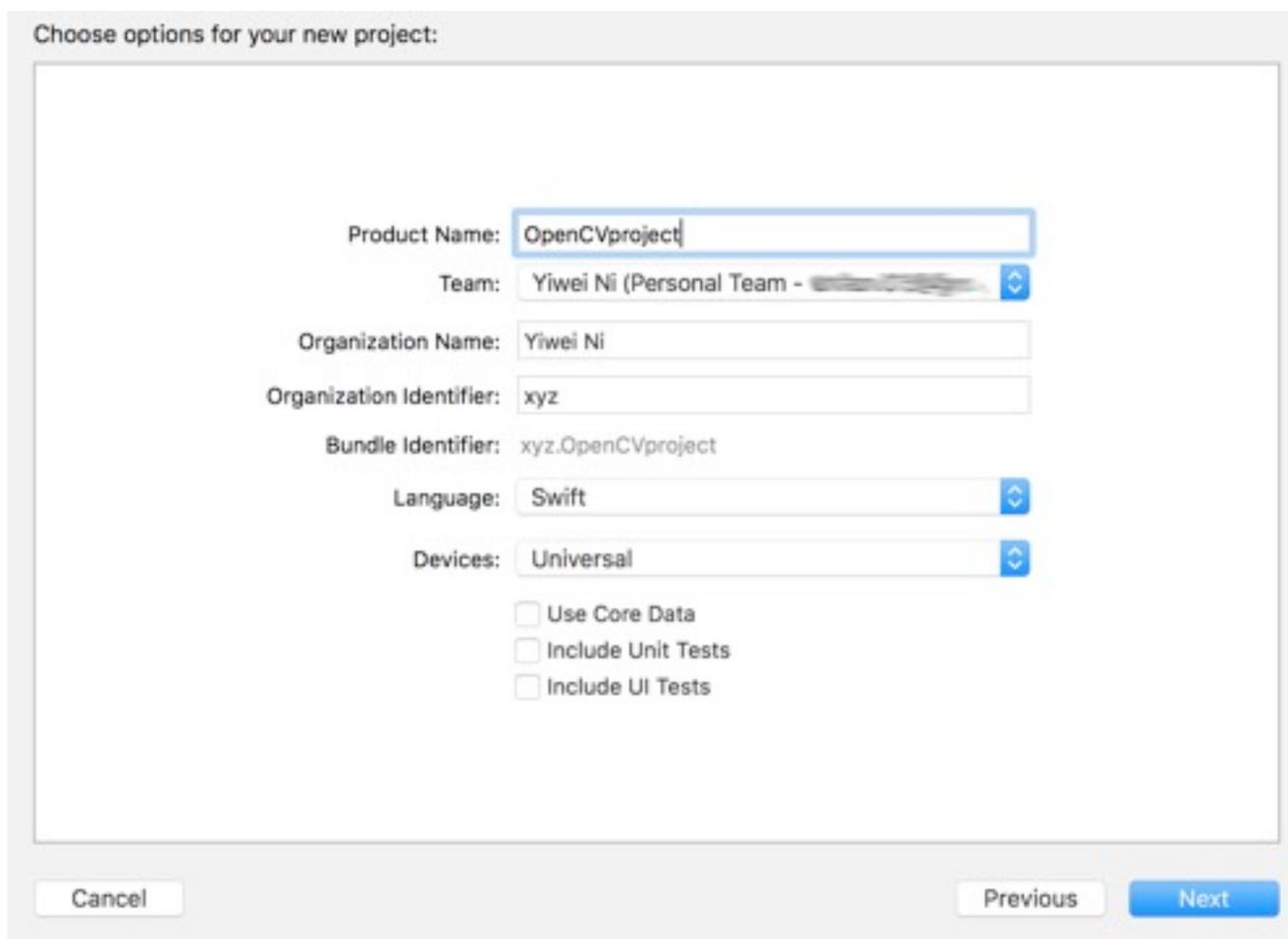
Check out an existing project

Start working on something from an SCM repository.

3. Выберите пункт "**Single View Application**".

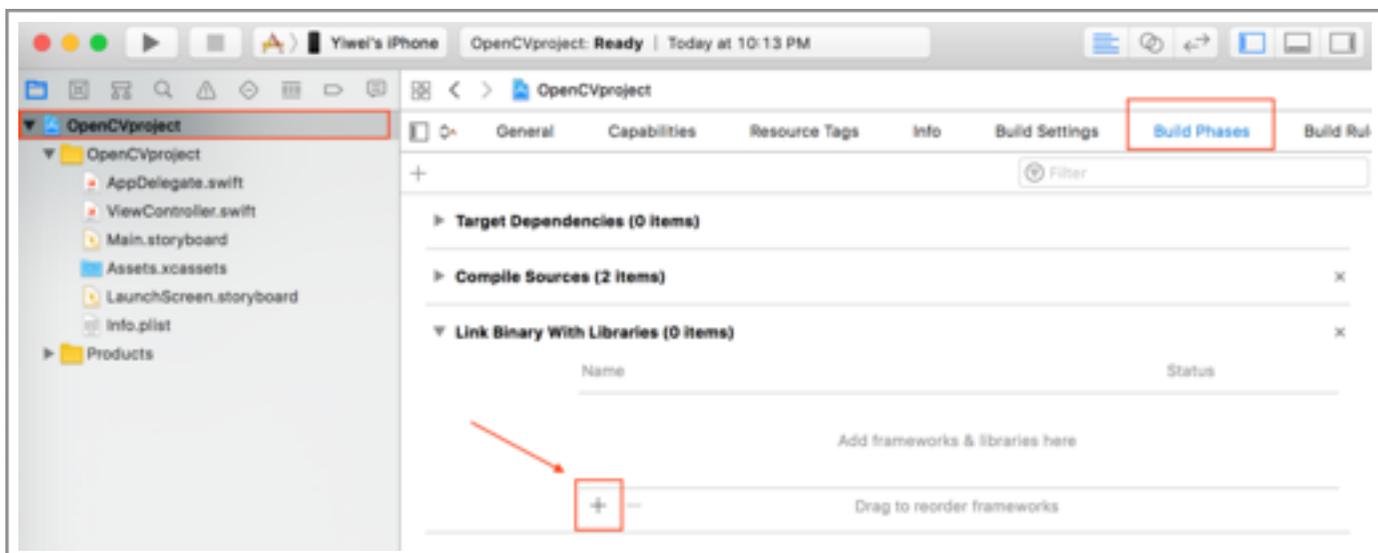


Я собираюсь назвать проект "OpenCVproject" и выбрать язык(language) **Swift**.

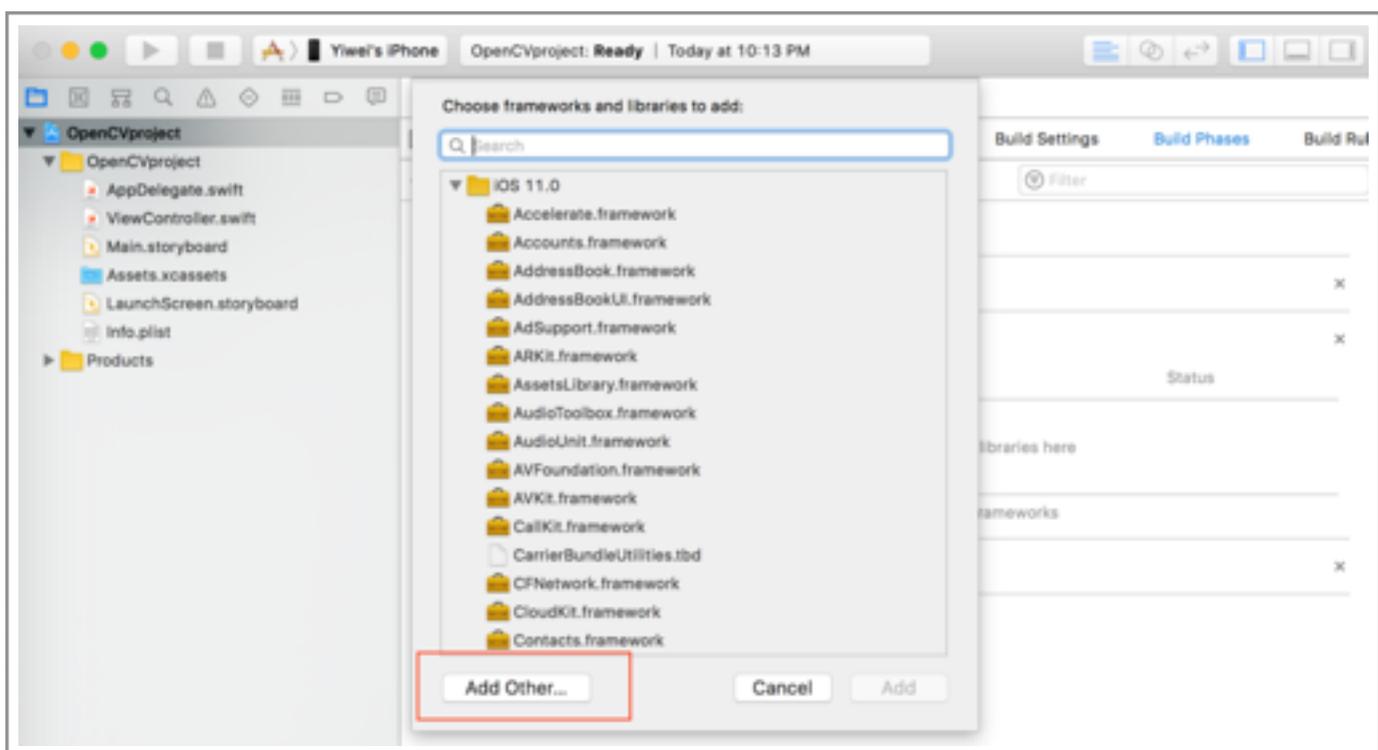


Давайте начнем кодировать

1. К **Xcode**-проекту добавьте фреймворк(библиотеку) **opencv2.framework**
2. Вы можете просто перетащить файл **opencv2.framework** на левую панель в навигаторе проекта в **IDE**-среде разработки **Xcode**. Однако я сделаю это более сложным способом, что убедиться, что он должным образом **присоединен (прилинкован; linked)**.
3. Щелкните по своему проекту в левой панели навигатора и пойдите по пунктам меню **Build Phases -> Link Binary With Libraries (Фазы построения -> Присоединить двоичный файл с библиотеками)**. Щелкните по кнопке "+".



4. Затем щелкните по кнопке **"Add Other ..."** ("Добавить другие ...").

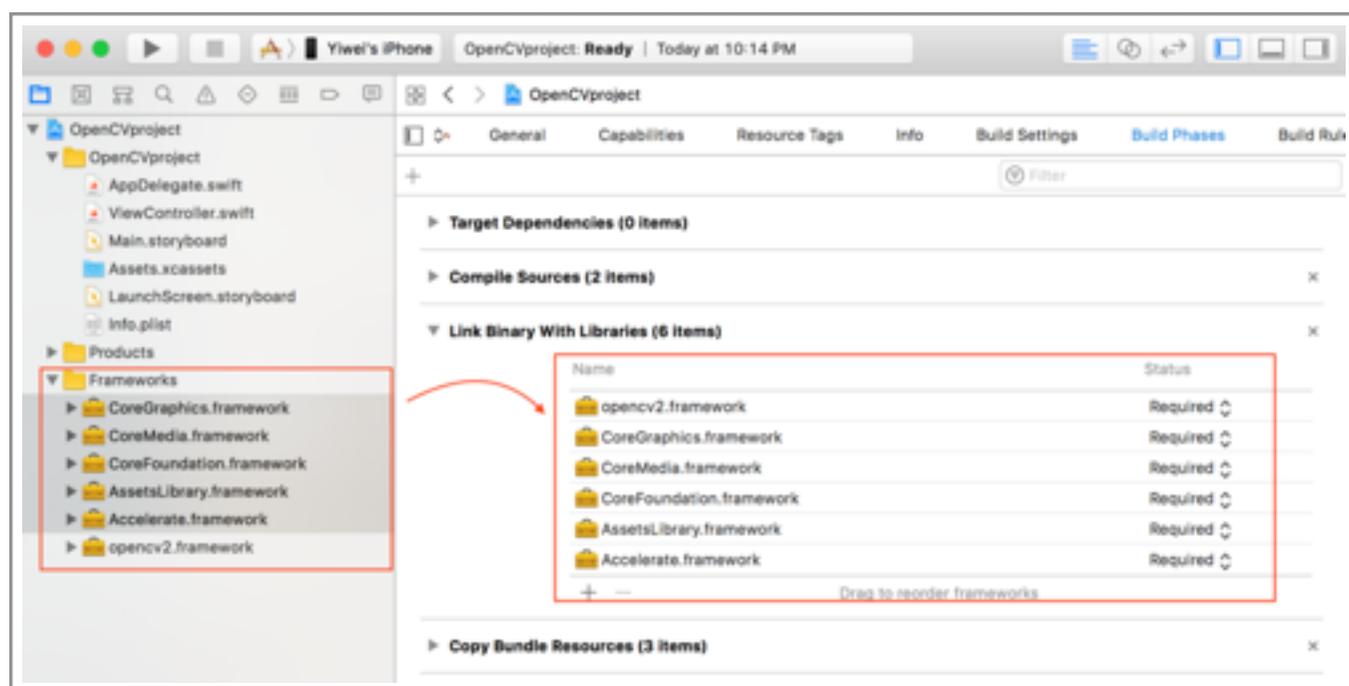


5. Перейдите прямо к файлу `opencv2.framework` и добавьте его. Это действие должно автоматически скопировать файл фреймворка (библиотеки) в ваш каталог проекта. Для должной работы фреймворка `OpenCV` вы, возможно, должны были бы в проект присоединить несколько других фреймворков(библиотек). Для моего проекта я добавил следующие:

- `AssetsLibrary`
- `CoreGraphics`
- `CoreMedia`
- `CoreFoundation`
- `Accelerate`

Если это не удастся, попробуйте фреймворки `UIKit`, `Foundation`, `CoreVideo`, `CoreImage` ...

Теперь в левой панели навигатора, должна появиться новая папка под названием "`Framework`" и содержать все фреймворки, которые вы только что добавили. Если какой-либо из этих фреймворков не перечислен на панели пункта меню `Build Phases -> Link Binary With Libraries` (Фазы построения -> Присоединить двоичный файл с библиотеками), то просто перетащите его из навигатора и добавьте его в список.



6. На панели пункта меню `Build Settings -> Framework Search Paths` (Параметры настройки построения -> Пути поиска фреймворков),

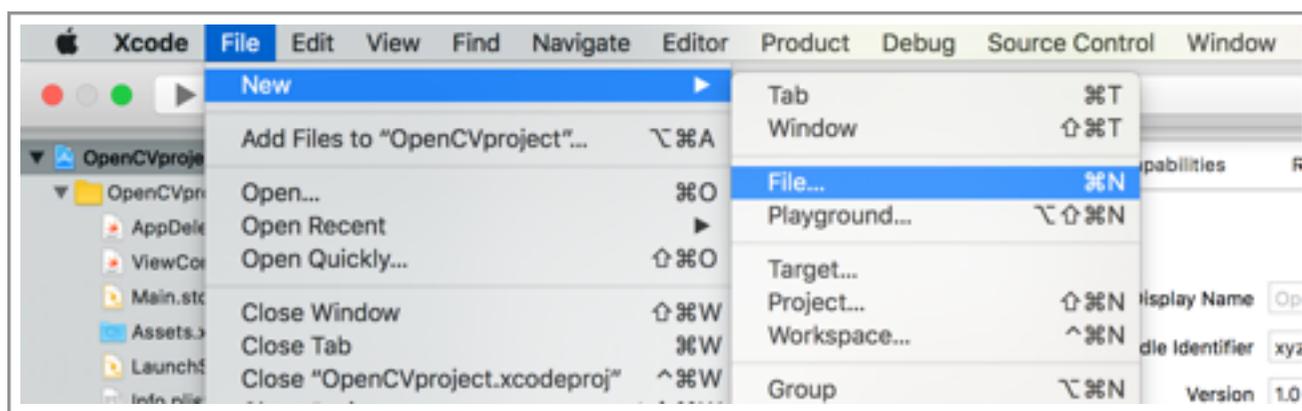
добавьте корректный путь для фреймворка `opencv2.framework`. Вы можете использовать выражение `$(PROJECT_DIR)/` для корневого каталога вашего проекта.



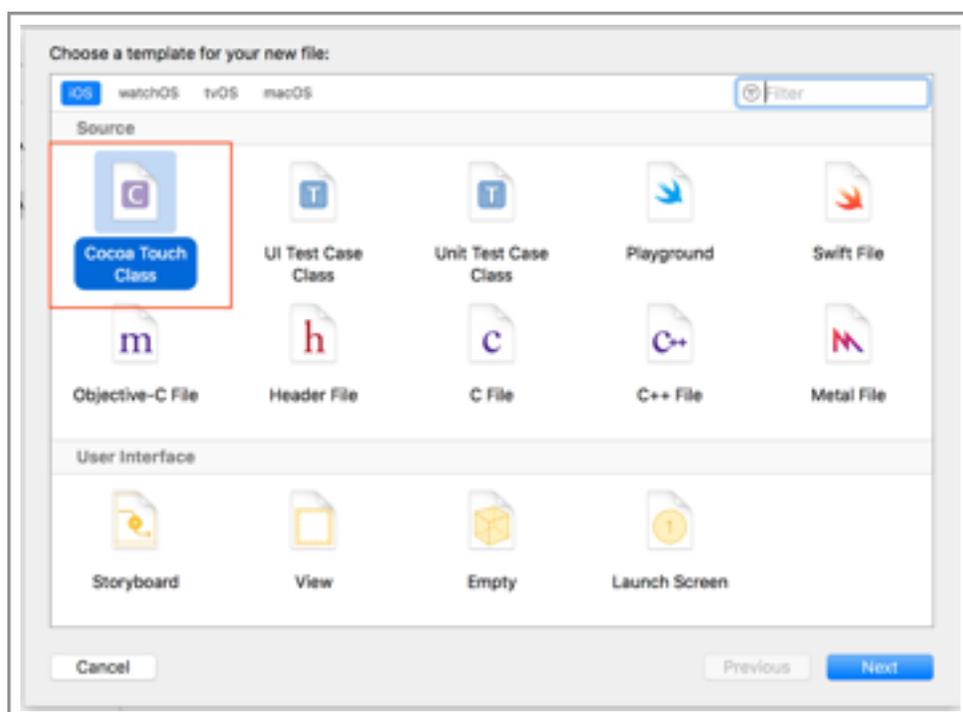
Это важно. Перейдите к фактическому каталогу и убедитесь в соответствии пути и расположения физического файла.

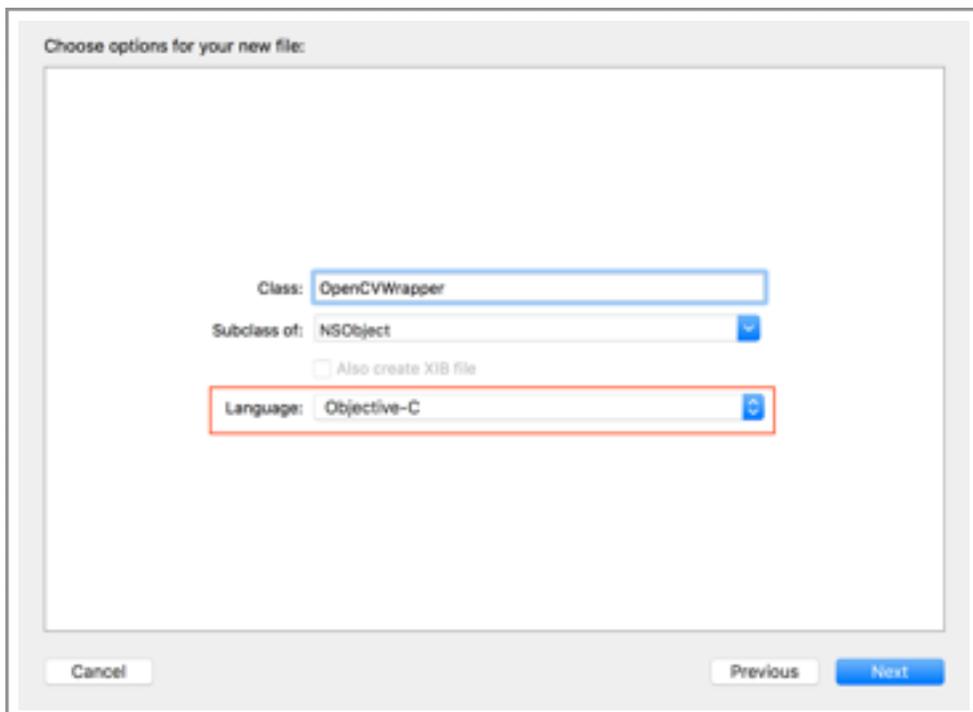
2. Создайте класс обертки и заголовок моста

1. Перейдите к пунктам меню **File -> New -> File...** (Файл -> Новый-> Файл...)

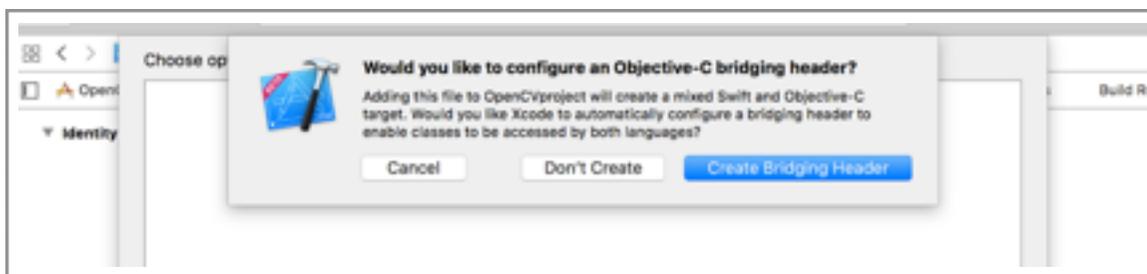


2. Создайте новый сенсорный класс **Cocoa Touch Class**. Давайте назовем его **OpenCVWrapper** и выберем язык **objective-C**.





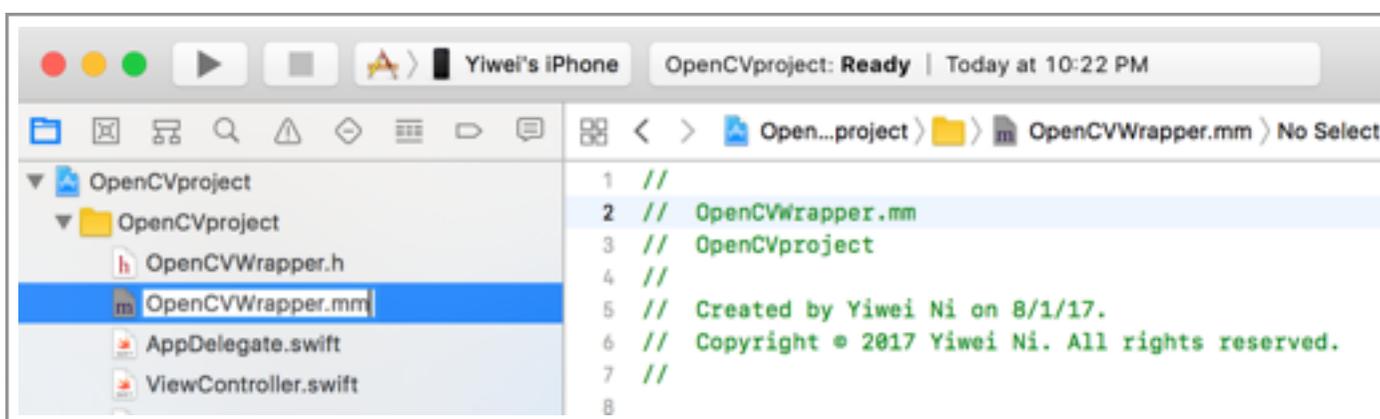
3. Сохраните его в каталоге своего проекта и нажмите кнопку **"Create Bridging Header"** ("Создать заголовок моста").



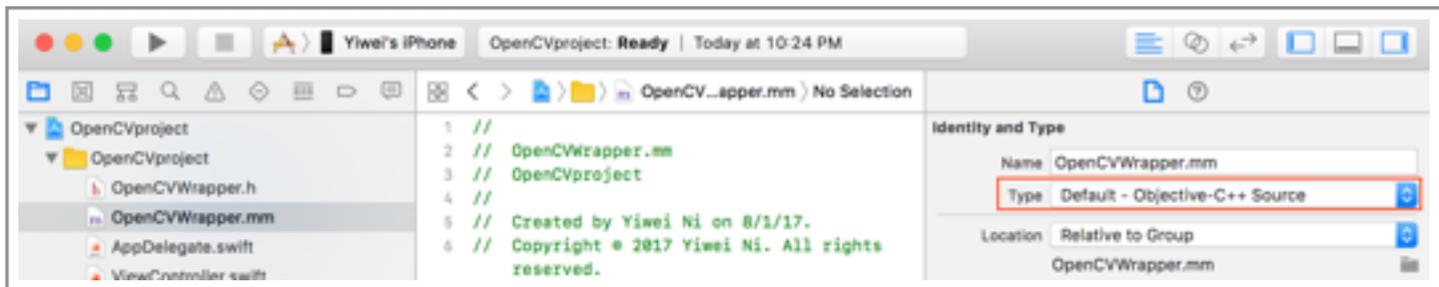
4. Вернитесь к панели слева с навигатором проекта. Здесь есть **3** новых файла, которые вы только что создали: **OpenCVWrapper.h**, **OpenCVWrapper.m** и **OpenCVproject-Bridging-Header.h**. Теперь, в файл **OpenCVproject-Bridging-Header.h** добавьте следующий оператор импорта.

```
#import "OpenCVWrapper.h"
```

5. Щелкните по файлу **OpenCVWrapper.m** и расширение файла переименуйте на **".mm"**. Если вы хотите, также в файле вручную измените **OpenCVWrapper.m** на **OpenCVWrapper.mm**.



Теперь, IDE-среда разработки **Xcode** должна распознать этот файл как **Objective C++**-файл.



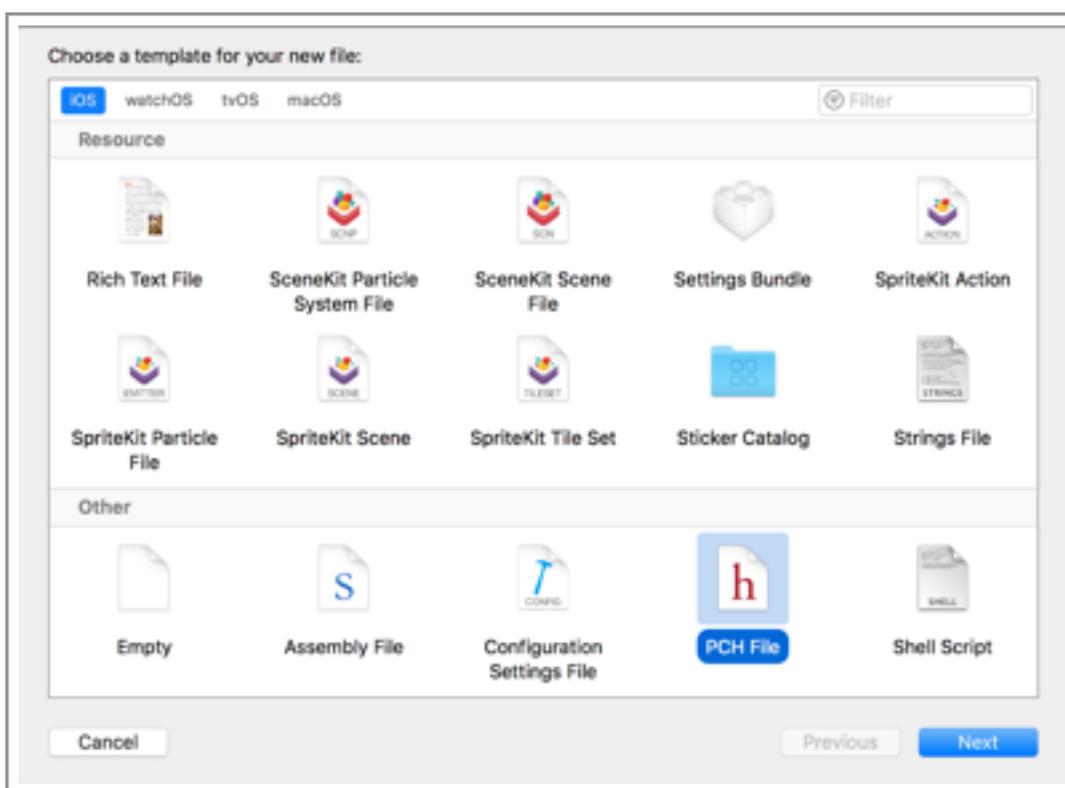
6. Перейдите к файлу **OpenCVWrapper.mm** и сверху файла добавьте следующий оператор импорта.

```
#import <opencv2/opencv.hpp>
```

IDE-среда разработки **Xcode** будет жаловаться, что она не может найти файл. Не волнуйтесь. Эта ошибка будет скоро устранена.

3. Добавьте заголовочный файл префикса

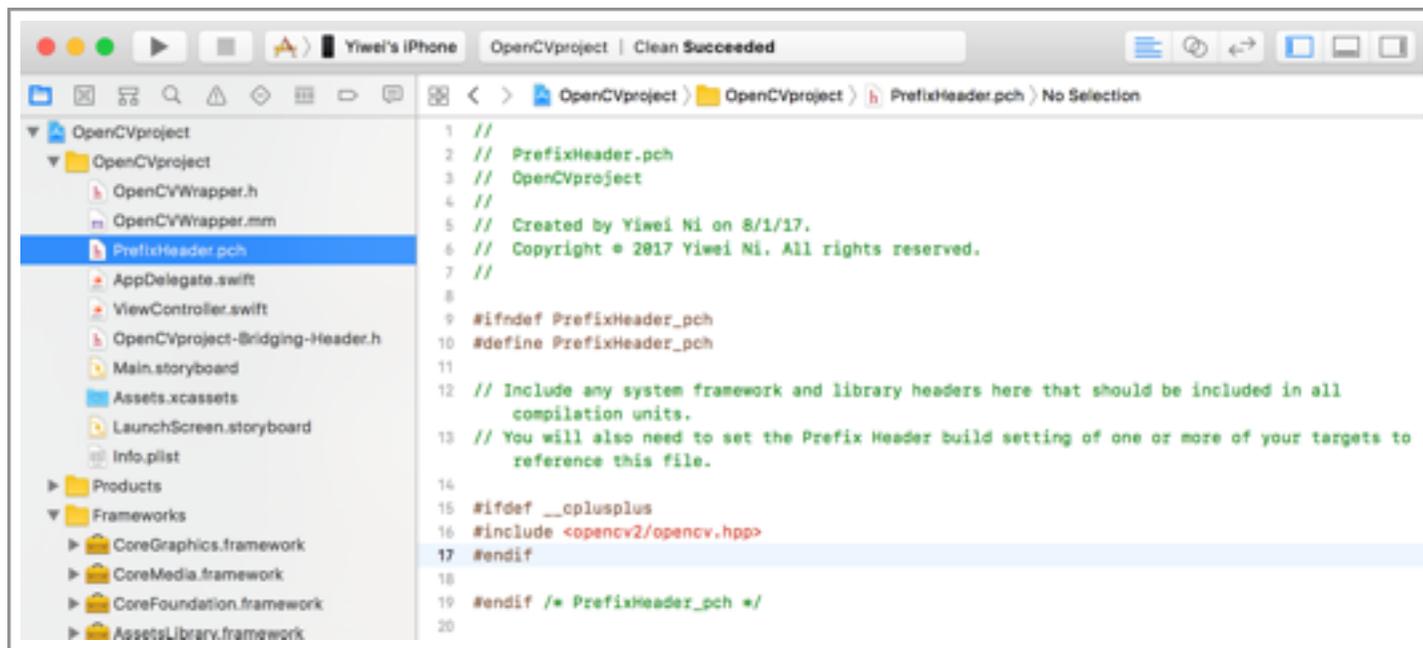
1. Перейдите к пункту меню **File -> New -> File...** (Файл -> Новый-> Файл ...) (снова). Прокрутите панель до низа, где вы найдете пункт **PCH File** (PCH-файл). Создайте файл и сохраните его в каталоге своего проекта.



2. Включите следующие строки в недавно созданный файл `PrefixHeader.pch`.

```
#ifndef __cplusplus
#include <opencv2/opencv.hpp>
#endif
```

Таким образом, теперь содержание файла похоже на это:

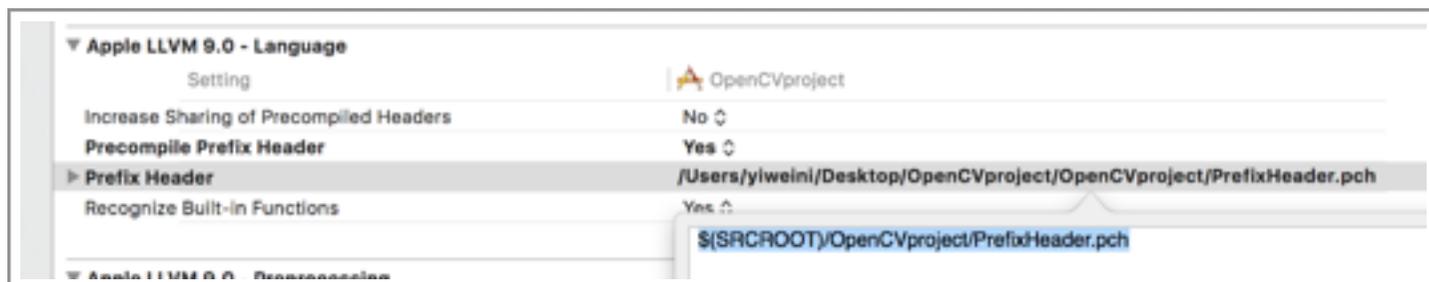


3. Щелкните по своему проекту в навигаторе на панели слева. На панели пункта меню `Build Settings -> Prefix Header (Параметры настройки построения -> Заголовок префикса)` добавьте корректный путь к вашему файлу `PrefixHeader.pch`. Это должна быть одна из следующих строк в зависимости от того, где вы создали файл.

```
$(SRCROOT)/OpenCVproject/PrefixHeader.pch
```

or

```
$(SRCROOT)/PrefixHeader.pch
```



Снова, это важно. Перейдите к фактическому каталогу и убедитесь в соответствии пути и расположения физического файла.

4. Распечатайте версию библиотеки OpenCV

Чтобы протестировать, правильно ли мы установили все, давайте в файле `ViewController.swift` напишем код и попытаемся распечатать версию библиотеки `OpenCV`.

1. Напишите простую функцию, возвращающую версию используемой вами библиотеки `OpenCV`. Внутри файла `OpenCVWrapper.mm` прямо ниже строки `@implementation OpenCVWrapper` добавьте следующую функцию.

```
+ (NSString *)openCVVersionString {  
    return [NSString stringWithFormat:@"OpenCV Version %s",  
        CV_VERSION];  
}
```

2. Не забудьте в файл `OpenCVWrapper.h` добавить заголовок функции. Ниже строки `@Interface OpenCVWrapper:NSObject` добавьте следующую строку.

```
+ (NSString *)openCVVersionString;
```

3. Теперь перейдите к файлу `ViewController.swift` и внутри функции `viewDidLoad()` после вызова функции `super.viewDidLoad()` запишите следующую строку:

```
print("\(OpenCVWrapper.openCVVersionString())")
```

4. Выполните проект и вы должны увидеть текст `Optional("OpenCV Version 3.2.0")`, распечатанный в консоли, если все корректно.

5. Процесс обработки изображения с помощью OpenCV

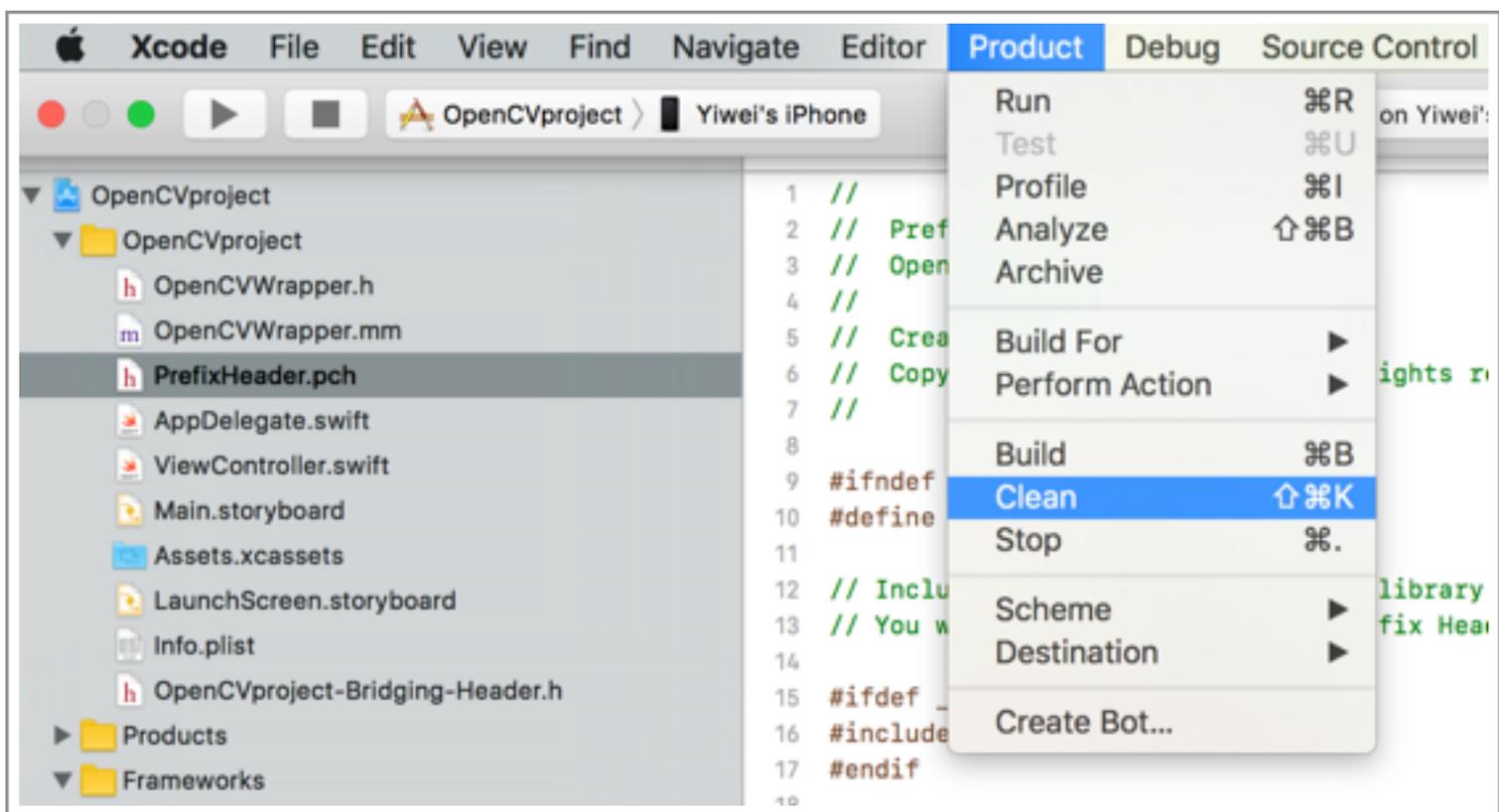
К счастью официальная учебная страница библиотеки `OpenCV` предоставила нам некоторые функции для преобразования между `cv:Mat` и `UIImage`.

(Примеры прибывают скоро)

Советы

Если вы получите сообщение `"opencv2/opencv.hpp not found"` ("`opencv2/opencv.hpp не найден`") или другие ошибки...:

- Поместите оператор `#import <opencv2/opencv.hpp>` ранее всех других операторов импорта.
- Убедитесь, что вы установили флажок в пункте `"Copy items if needed"` ("`Копировать элементы в случае необходимости`"), когда вы перетаскиваете `opencv2.framework` в свой Xcode-проект.
- Убедитесь, что в навигаторе слева ничто не выделено красным цветом. Если этот пункт будет выделено красным цветом, то вы должны будете удалить этот пункт и снова добавить файл назад, потому что IDE-среда разработки Xcode не может найти фактический файл по данному пути.
- Убедитесь, что фактический файл `opencv2.framework` существует в каталоге по пути, который вы поместили на панели пункта меню `Build Settings -> Framework Search Paths` (Параметры настройки построения -> Пути поиска фреймворков).
- Убедитесь, что фактический файл `PrefixHeader.pch` существует в каталоге по пути, который вы поместили на панели пункта меню `Build Settings -> Prefix Header` (Параметры настройки построения -> Заголовок префикса).
- Убедитесь, что вы корректно написали оператор импорта `import OpenCVWrapper.h` в файле `OpenCVproject-Bridging-Header.h`.
- Попробуйте выполнить очистку с помощью пунктов меню `Clean` (Очистить) и `Clean Build Path` (Очистить путь построения). Нажмите клавишу "Option" ("Режим"), в то время как выпадающее меню открыто, и пункт `Clean` (Очистить) превратится в пункт `Clean Build Path` (Очистить путь построения).



Кроме того...

Эта статья на [Medium](#)-сайте:

- [OpenCV и Swift: шаг за шагом \(OpenCV with Swift - step by step\)](#)

<https://medium.com/@yiweini/opencv-with-swift-step-by-step-c3cc1d1ee5f1>

Вот некоторые другие крутые [Medium](#)-истории о библиотеке [OpenCV](#) и ОС [iOS](#), которые считаю очень полезными. Надеюсь, парни вам понравится это чтение :)

- [Современное руководство по openCV для ОС iOS \(The modern guide to openCV on iOS\)](#)

<https://medium.com/ios-os-x-development/the-fd4fcb249358>

- [iOS: OpenCV и Swift \(iOS - OpenCV and Swift\(\)\)](#)

<https://medium.com/@borisohayon/ios-opencv-and-swift-1ee3e3a5735b>