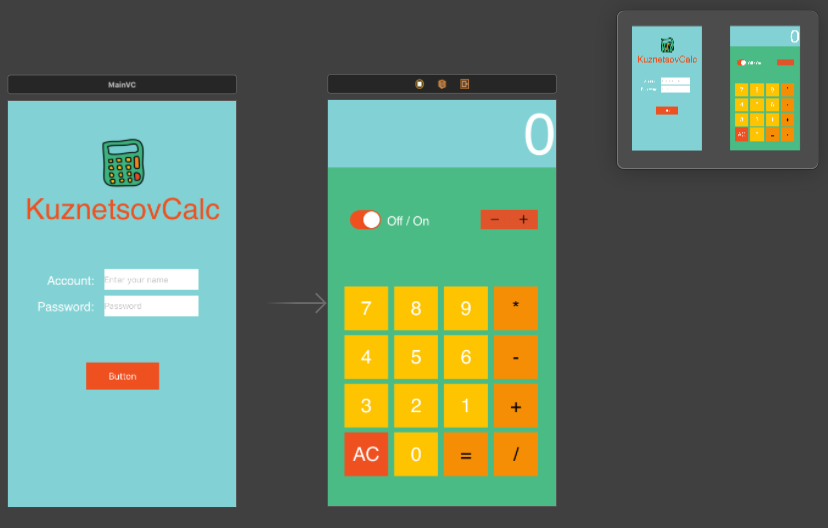
Калькулятор на Swift

Кузнецов К.А.



# Исходные данные

В данной работе использовалось следующее программное обеспечение:

* MacOS Catalina
* XCode 12
* Adobe XD
* Макет приложения для Adobe XD - <https://drive.google.com/file/d/1kxKxndEK9VRBjy1E9RCkrvv_8v4UU-Ks/view?usp=sharing>
* Макет приложения для Figma - <https://www.figma.com/file/swnqYwJ8RYoGFxWKu3lt5C/Calculator>

Средства разработки:

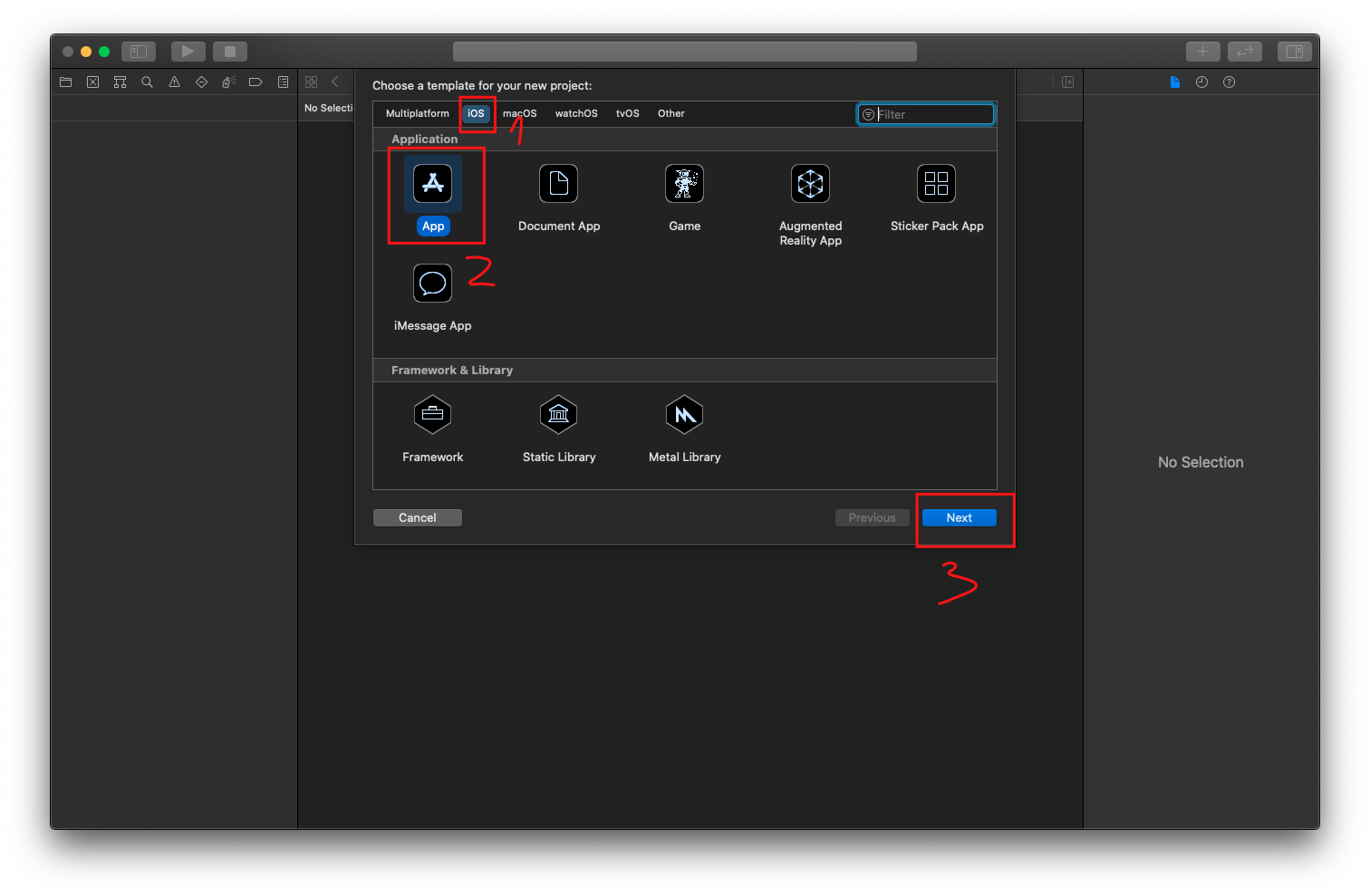
* UIKit framework
* App Interface Storyboard
* Language Swift

# Создание нового проекта

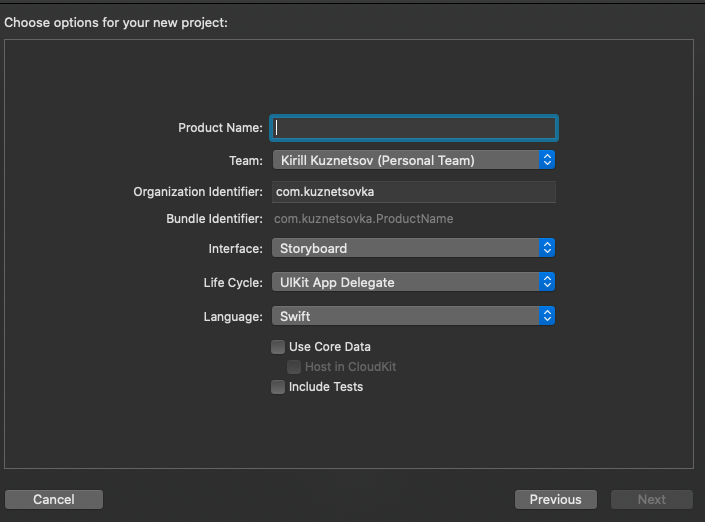
* Запустите X Code из launchpad или dashboard
* В открывшемся окне выберите пункт: «Create a new Xcode project»



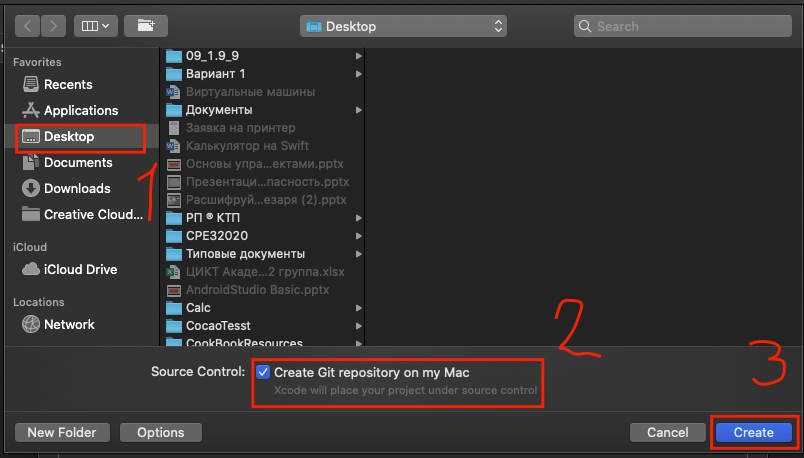
* В открывшемся окне выберите платформу iOS и шаблон приложения App, нажмите кнопку Next



* В следующем шаге внесите информацию о своем проекте где:
  + Product name – имя вашего приложения
  + Team – аккаунт разработчика (если такой есть, параметр необязательный)
  + Organization Identifier – идентификатор организации (например: ru.kuznetsovka)
  + Interface – storyboard
  + Life Cycle – UIKit App Delegate
  + Language – Swift
  + Для данного проекта нам не нужны галочки напротив Use Core Data и Include Tests, снимаем все checkbox



* После внесения информации о проекте нажмите кнопку Next
* Выберите место где будет хранится ваш проект, при создании нового проекта в X Code можно сразу создать локальный git репозиторий в папке поставив соответствующий checkbox. После нажмите завершите создание проекта нажав на кнопку Create.

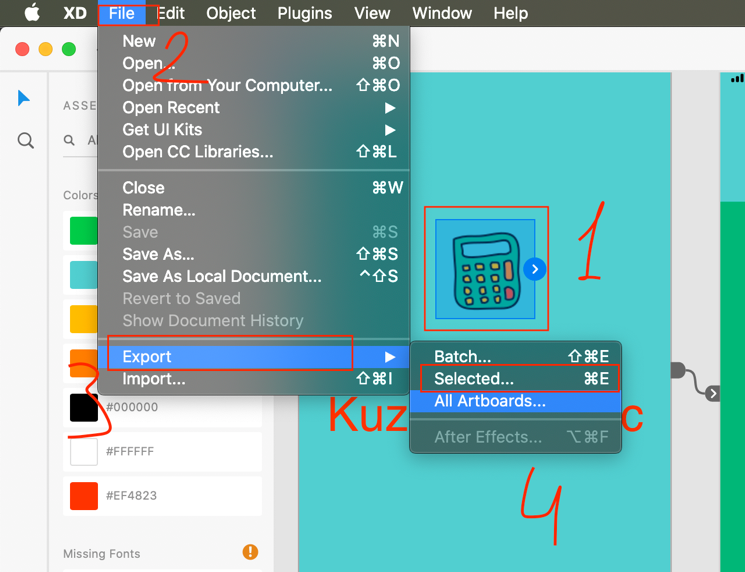


# Добавление ресурсов в проект из макета

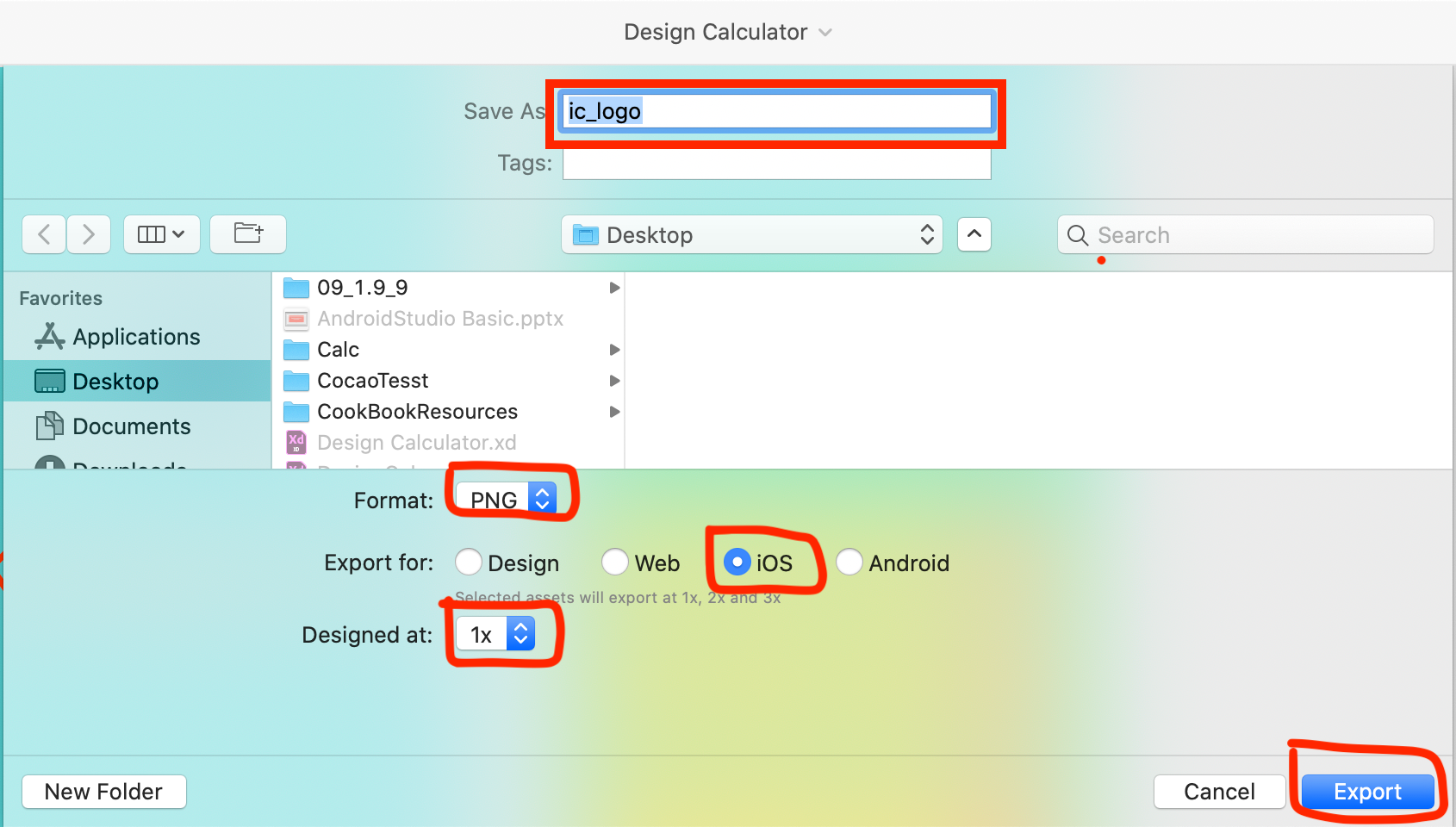
* Запустите скачанный макет приложения



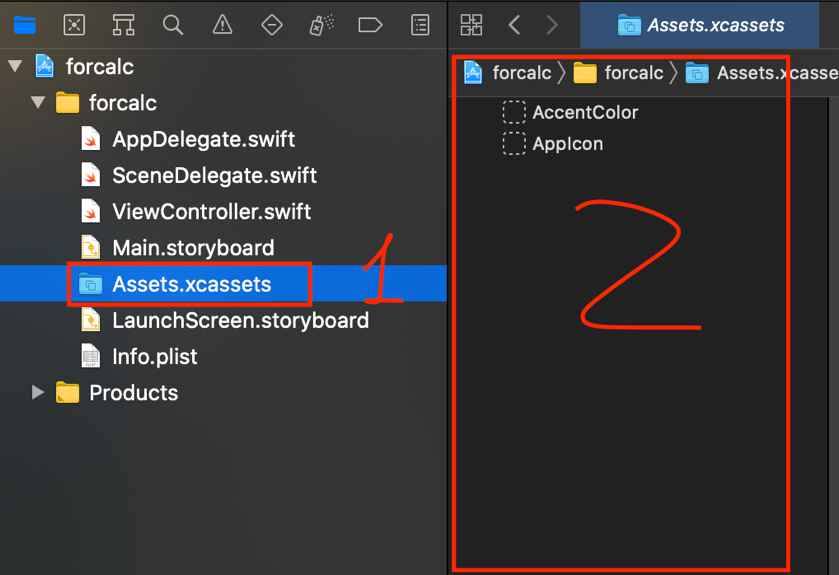
* В данном макете единственным элементом который необходимо экспортировать является логотип на экране загрузки. Для его экспорта графического ресурса в проект Xcode сделайте левый клик на изображения на панели инструментов выберите File, в выпадающем списке выберите пункт Export и Selected. (Пункт selected - экспортирует только те изображения которые выбраны)



* В открывшемся окне выберите место хранения для экспортируемых файлов, при необходимости информацию о файле, где:
  + Format: PNG (Формат экспортируемого изображения)
  + Export for: iOS (Выбор платформы)
  + Designed at: 1x (В каком формате нарисован)



* Для завершения экспорта нажмите на кнопку Export
* Для импорта графических ресурсов, откройте проект и перейдите в папку Assets.xcassets. Перетащите графические ресурсы в область 2



* В папке Assets.xcassets по умолчанию создается категория для иконок приложения которая называется AppIcon. Она хранит иконки для рабочего стола, меню настроек и уведомлений. Для добавления новой иконки, кликните по надписи AppIcon перетащите графический файл в нужную ячейку. В AppIcon все ячейки имеют определенное разрешения, например для иконки рабочего стола используются изображения размерами 120x120px для @2x и 180x180px для @3x.

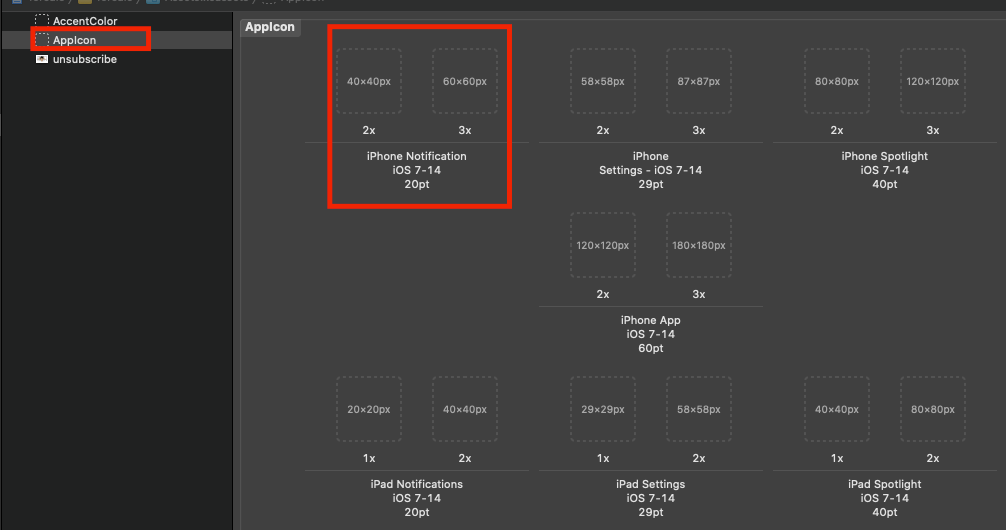
iPhone Notification - иконки для уведомлений

iPhone Settings - иконки для меню настроек

iPhone Spotlight - иконки для окна поиска

iPhone App - иконки для рабочего стола

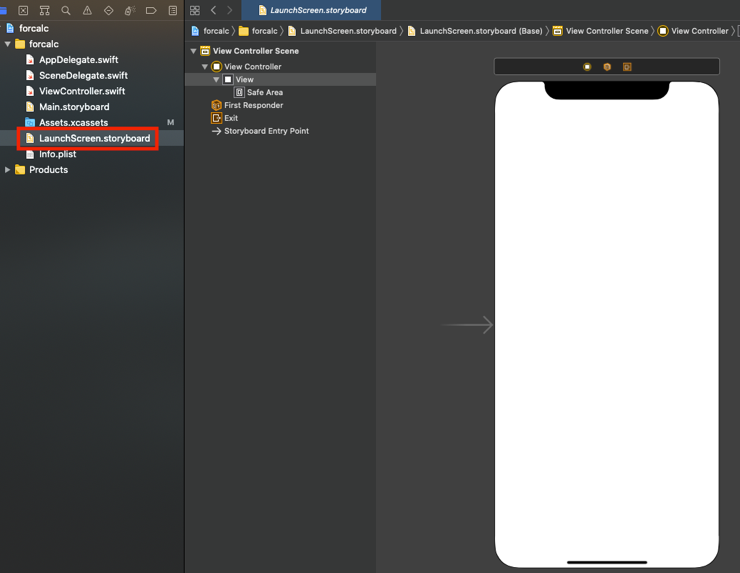
AppStore - иконка для магазина AppStore



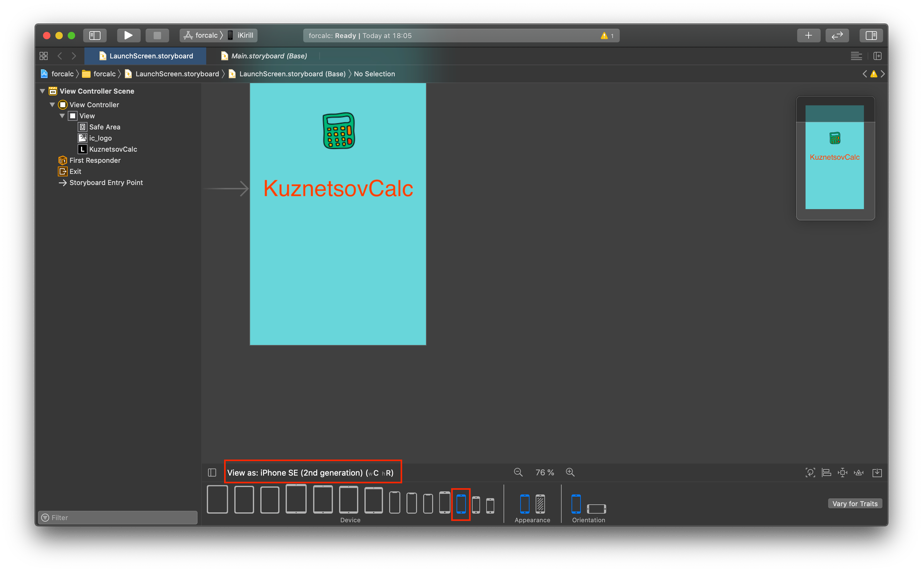
* В данном проекте необходимо добавить иконки приложения для рабочего стола. Для автоматического создания иконок используйте публичные сервисы, графический редактор или плагин в Adobe XD и Figma – App Icon Generator

# Экран загрузки (LaunchScreen)

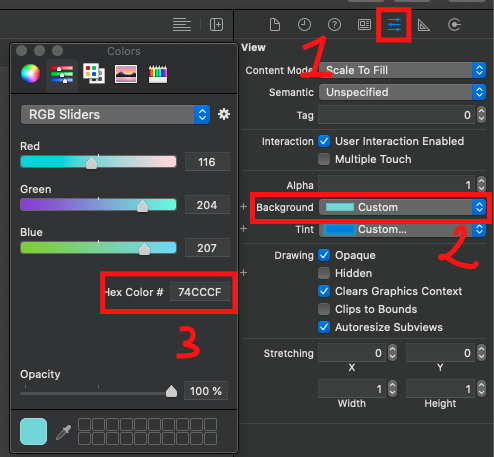
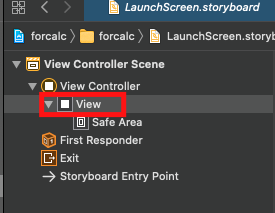
* При первом запуске приложения у пользователя в iPhone отображается LaunchScreen (Экран загрузки). Данный экран всегда отображается когда приложение загружается первый раз и не находилось открытым в фоновом режиме. Для реализации экрана загрузки в проекте XCode есть файл LaunchScreen.storyboard



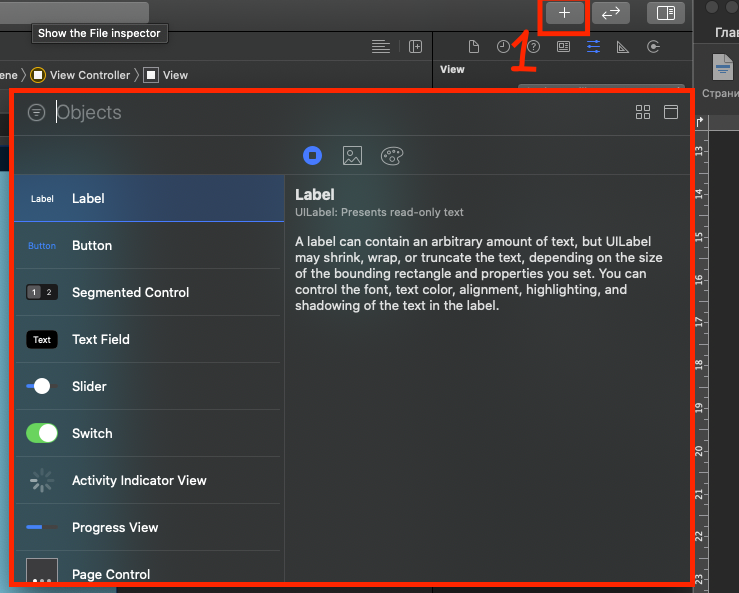
* Данный проект разрабатывается под смартфон iPhone 7, 8, SE 2GE. Чтобы в storyboard отобразить модель именно этого телефона нажмите в нижней части экрана на надпись View as и выберите необходимое устройство



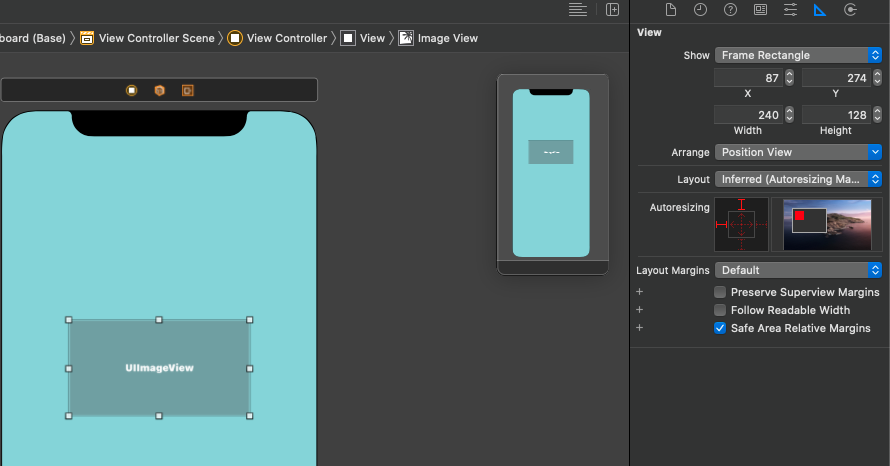
* Согласно макету, приложение должно содержать иконку с выравниванием по центру, голубой цвет фона окна и текст с названием приложения. В качестве элементов будут использоваться Label и ImageView.
* Для изменения фона окна в структуре нажмите на элемент View и в Attributes Inspector измените свойство background выбрав цвет Custom и введя код цвета с макета #74CCCF.



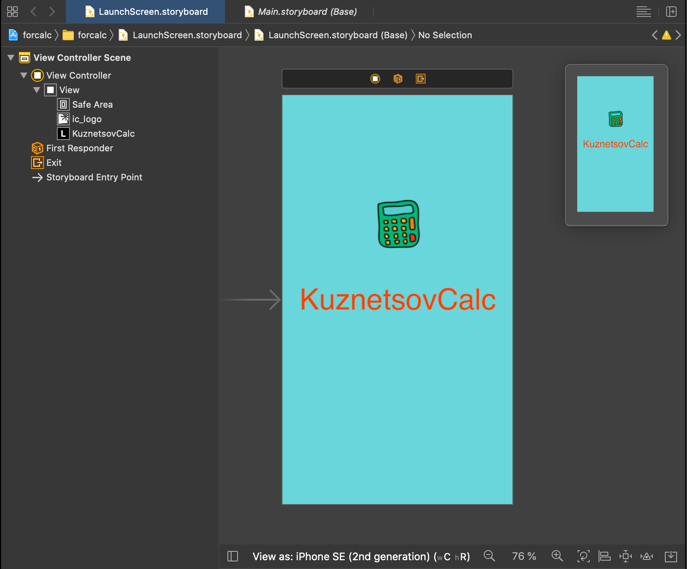
* Для добавления иконки на экран в правой части экрана нажмите на плюс для отображения библиотеки элементов.



* В открывшемся списке выберите элемент ImageView и перетащите его в центр экрана. Далее выберите элемент и перейдите в Inspector Area на вкладку Size Inspector. Измените свойства расположения объекта в группе View как в макете, где x = 138, y = 158, width = 100, height = 100

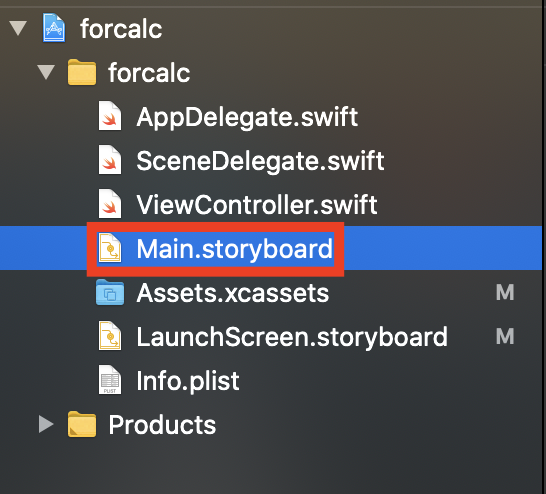


* В Inspector area откройте вкладку Attribute inspector и в свойстве Image выберите загруженный ранее логотип приложения в Assets
* Далее необходимо добавить на экран текст с названием приложения, для этого из библиотеки переместите элемент Label. Аналогично ImageView поменяйте свойства в Size Inspector для расположения элемента на экране, где x = 28, y= 334, weight = 320, height = 48. В Attribute Inspector измените свойства для текста введя свою фамилию и постфикс Calc (Пример: KuznetsovCalc), так же поменяйте настройки шрифта в свойстве Font выбрав:
  + Font – Custom
  + Family – Helvetica
  + Style – Regular
  + Size – 48.0

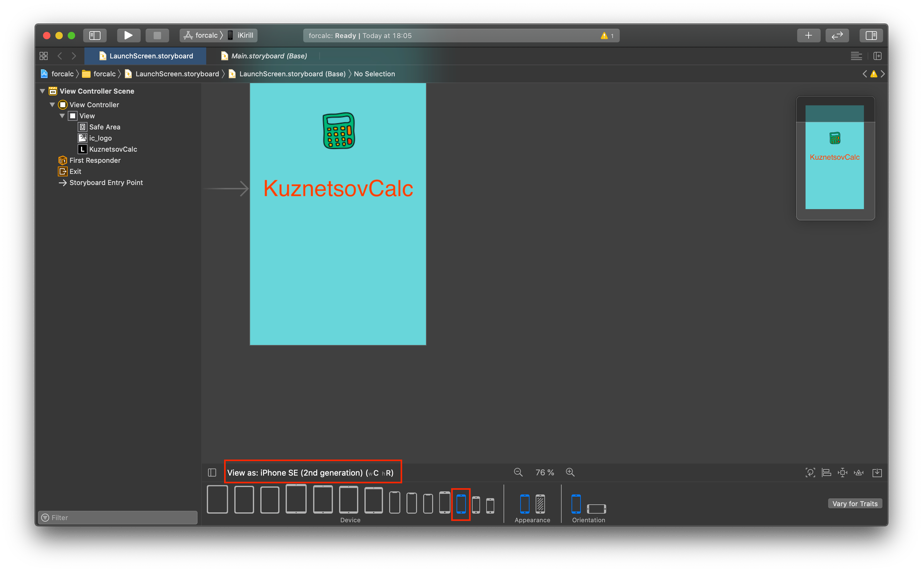


# Проектирование приложения

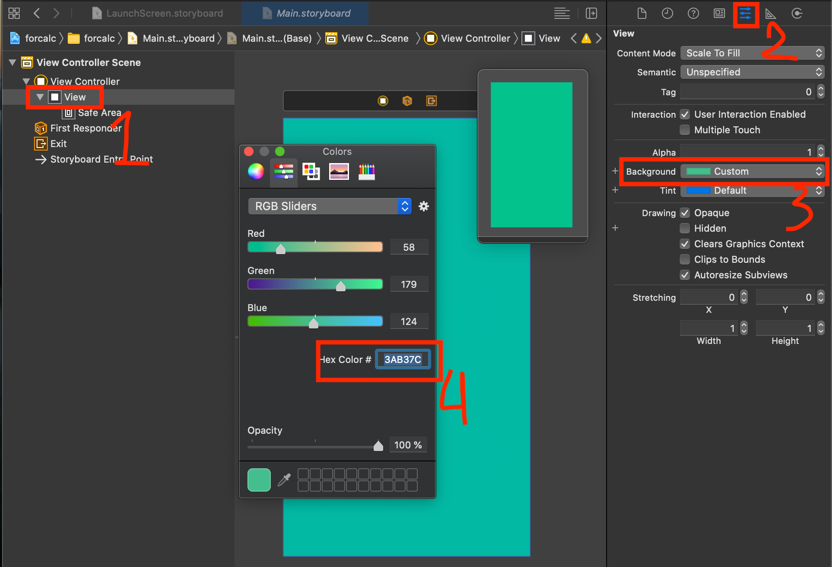
* Основным файлом в приложении является Main.storyboard, в данном файле будет производится проектирование экранов с помощью элементов.



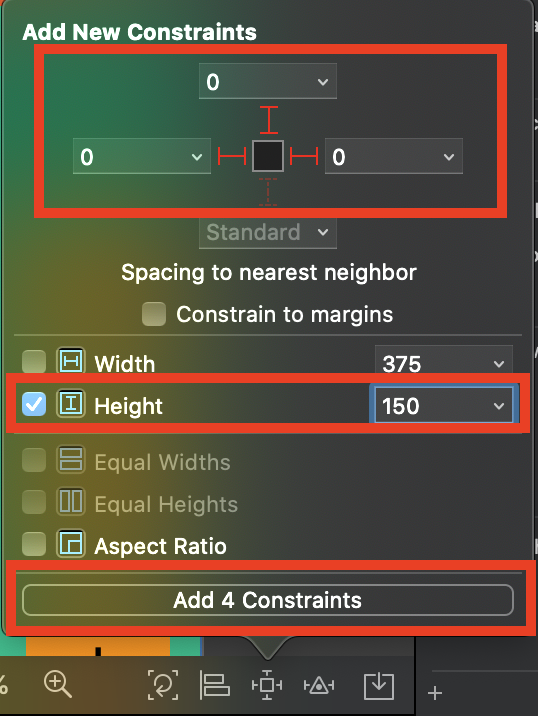
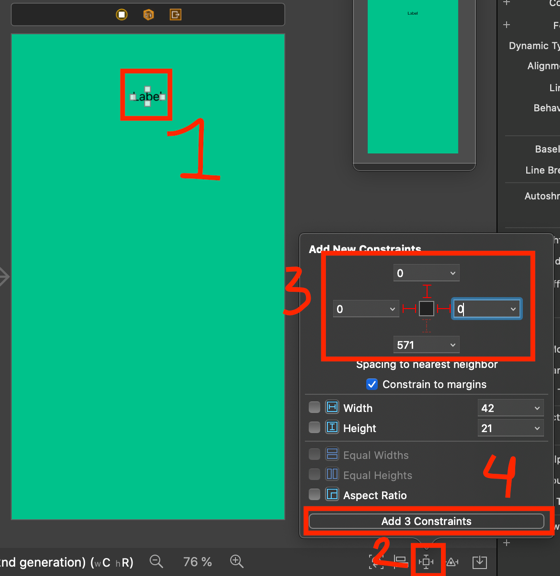
* Данный проект разрабатывается под смартфон iPhone 7, 8, SE 2GE. Чтобы в storyboard отобразить модель именно этого телефона нажмите в нижней части экрана на надпись View as и выберите необходимое устройство.



* Выберите в структуре окна элемент View и в свойствах Attribute Inspector измените значение у свойства background на custom на цвет #3AB37C.



* Добавьте на экран из библиотеки элементов Label, в нижний части экрана нажмите на кнопку добавления constraints (Необходим для привязки выбранного элемента к границам экрана), поставьте 0-ое значение для привязки левой, верхней и правой границы, для height укажите значение в 150 и нажмите на кнопку Add 4 Constraints для добавления

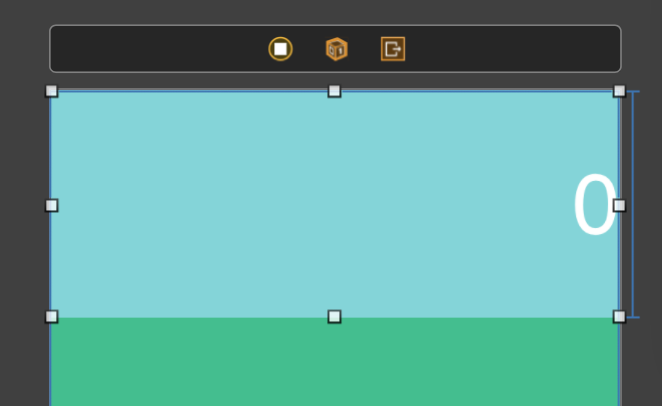


* Далее в Attribute Inspector выставите выравнивание Alignment для Label по правой стороне. В качестве отображаемого текста выставите значение 0 (ноль). Цвет текста Color поменяйте на белое (white). Для свойства Font выберите:
  + Font – Custom
  + Family – Helvetica
  + Style – Regular
  + Size – 48

Измените фон Label на цвет #74CCCF поменяв значение в свойстве background

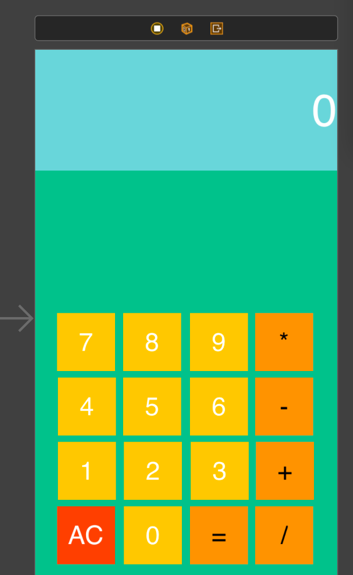


* Для элемента Label в Size Inspector выставите значение для высоты (height) равное 150



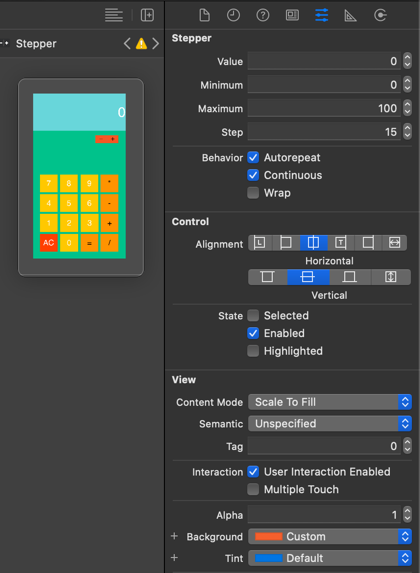
* Основными элементами приложения будут кнопки. Добавьте из библиотеки элементов на экран кнопку (button). В Size Inspector выставите следующие значения: x = 27, y = 480, weight = 72, height = 72. В Attribute Inspector выставите значение у свойства Font:
  + Font – Custom
  + Family – Helvetica
  + Style – Regular
  + Size – 32
* Для текста кнопки выставите у свойства Color значение на white. Согласно макету добавьте остальные кнопки на экран и выставите следующие значения для background:
  + У кнопки AC цвет #EF4823
  + У кнопок 0 – 9 цвет #FEBF10
  + У кнопок +, -, \*, /, = цвет #F6851F

В X Code для копирования элементов с имеющимися свойствами выберите элемент, нажмите на клавишу option и перетяните элемент на свободное пространство

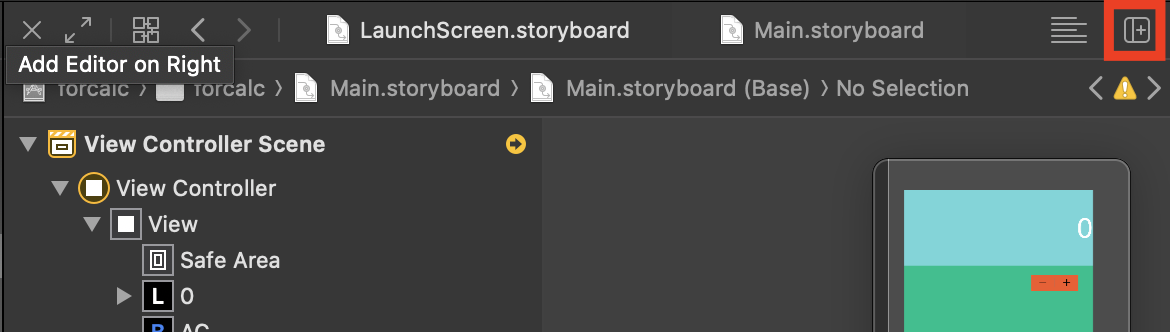


* У кнопок с арифметическими операндами выставите цвет текста кнопок на черный
* Последним элементов на экране остается stepper его функционал будет позволять изменять размер текста на экране. Выставите в Attribute Inspector:
  + Value – 0
  + Minimum – 0
  + Maximum – 100
  + Step – 15

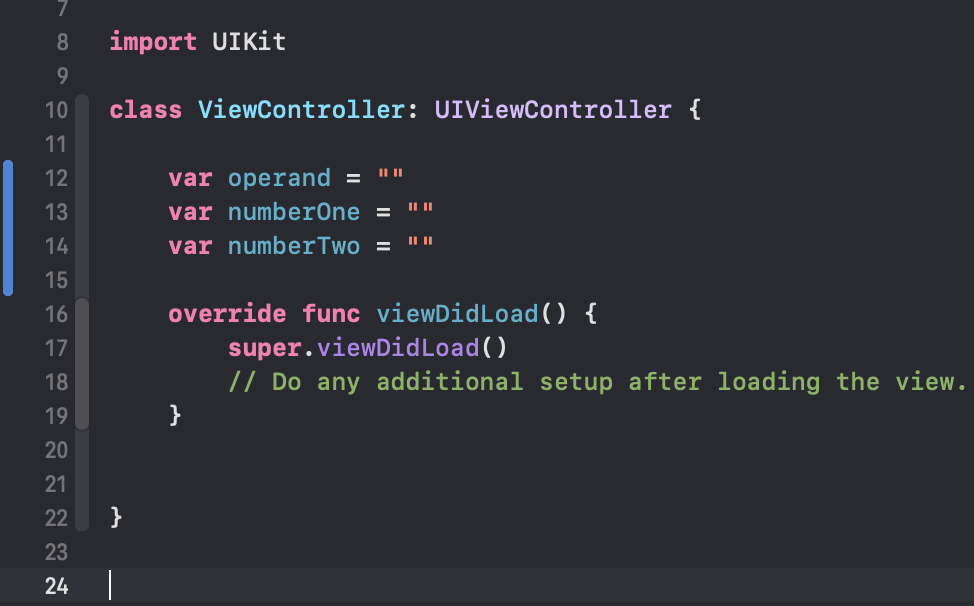
У свойства Background выставите цвет #EF4823



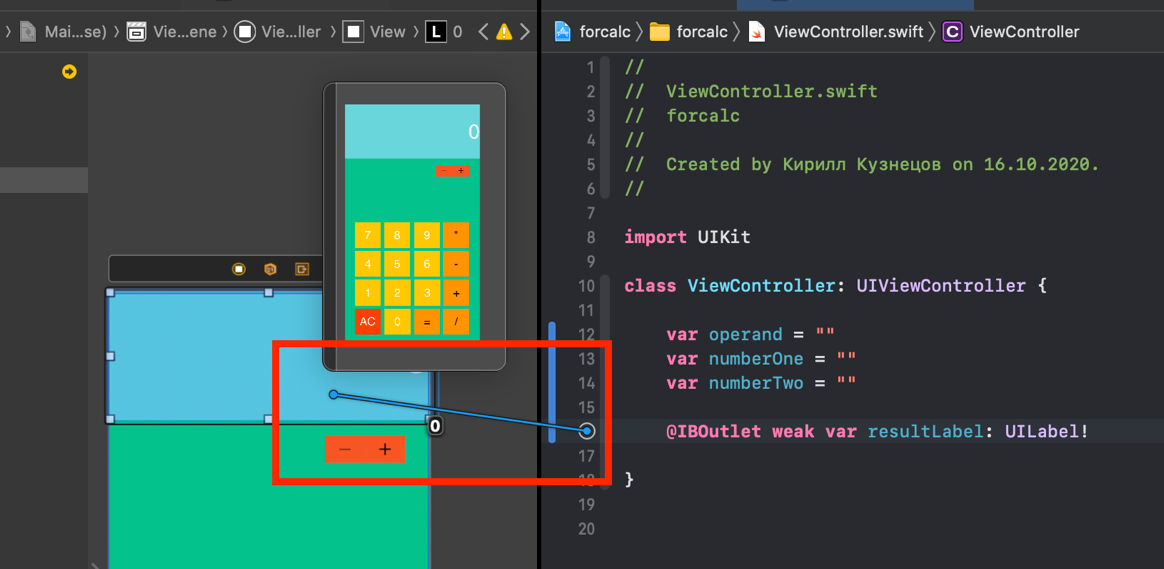
* На этом настройка интерфейса приложения закончена. По умолчанию вместе с проектом создается файл ViewController, чтобы настроить функционал нажмите на нем с зажатой клавишей option что бы разделить экран на два окна в одном, из которых будет Storyboard в другом файл ViewController. Либо воспользуйтесь кнопкой разделения экрана.



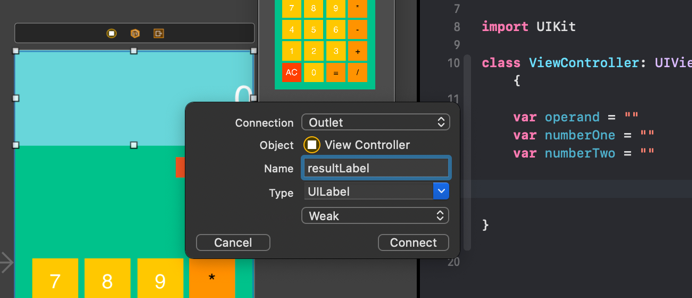
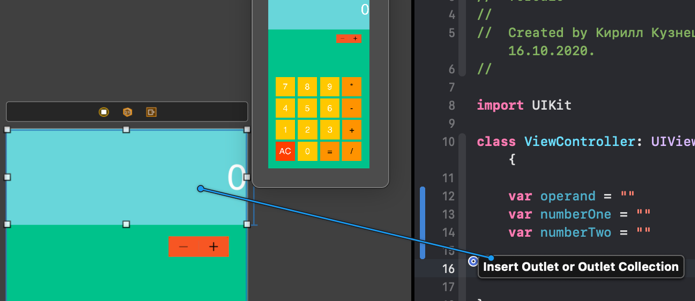
* Добавьте в проект 3 переменных с типом данных String: operand, numberOne, numberTwo



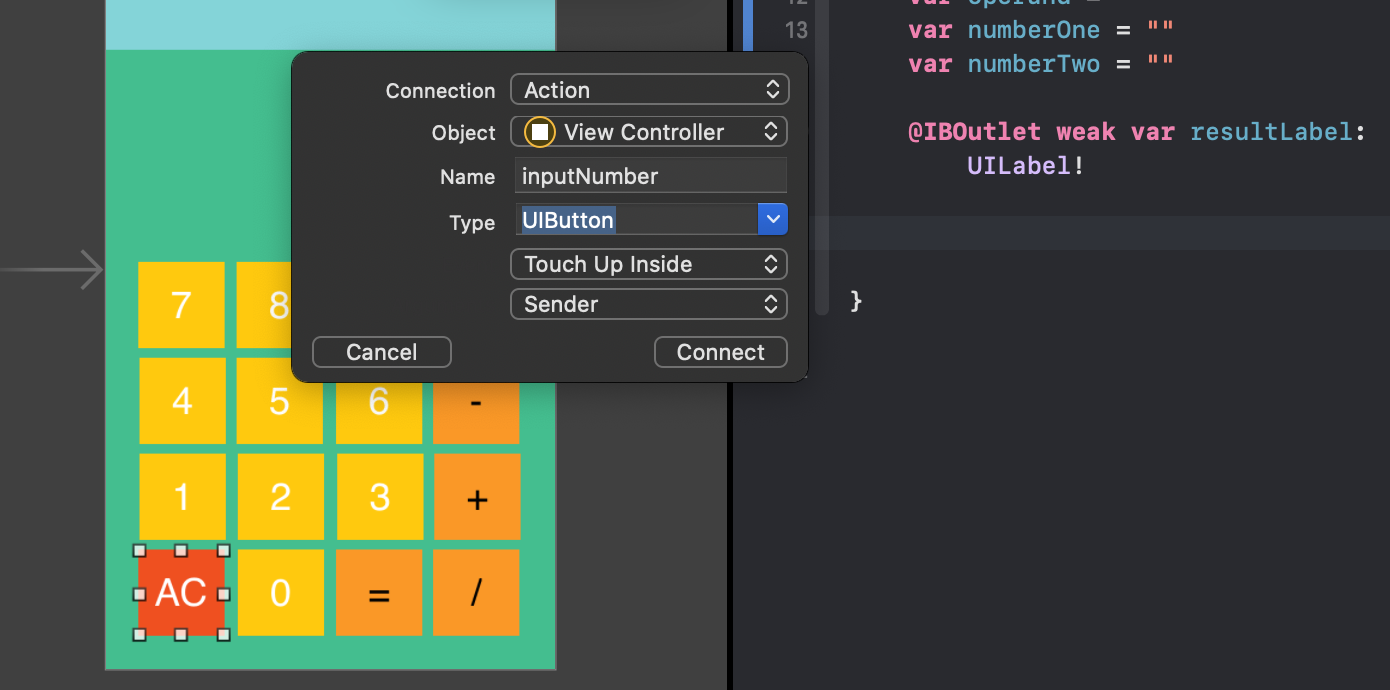
* В данном приложение функция viewDidLoad() нам не нужна поэтому можете удалить ее.
* В верхней части экрана Label будет отображать результат выполнения действий, создайте для него Outlet с именем resultLabel. И от точки сделайте связь кода с элементом.



Альтернативным вариантом можно сделать Outlet выделив элемент и зажав правую клавишу мыши перевести его в код.



* Создайте функцию для обработки нажатия кнопок от 0 до 9, для этого выберите любую кнопку и с помощью зажатой правой клавиши мыши переместите курсор в код. В открывшемся окне выберите:
  + Connection – Action
  + Name – inputNumber
  + Type - UIButton



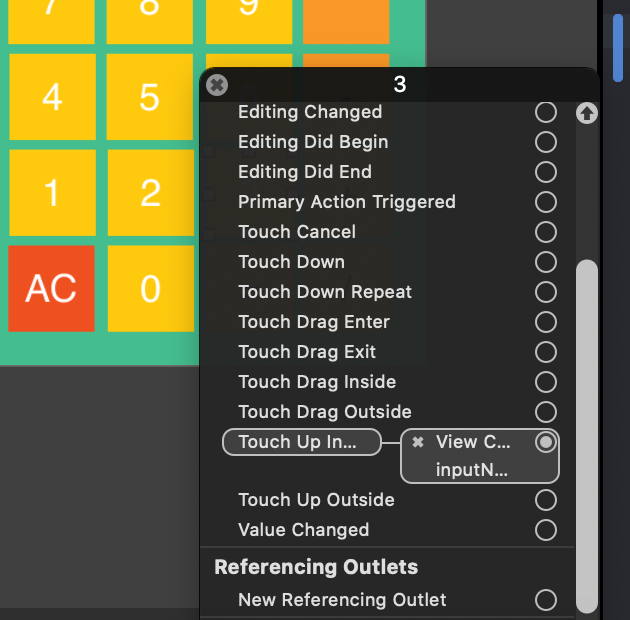
* Нажмите на кнопку Connect, результатом будет создана функция. Далее от точки в коде переместите связь для всех кнопок 0…9. Таким образом при нажатии на данные кнопки будет отрабатываться код находящийся в нашей функции



* При наведении курсора на точку рядом с функцией inputNumber на storyboard будут подсвечиваться кнопки, привязанные к данной функции



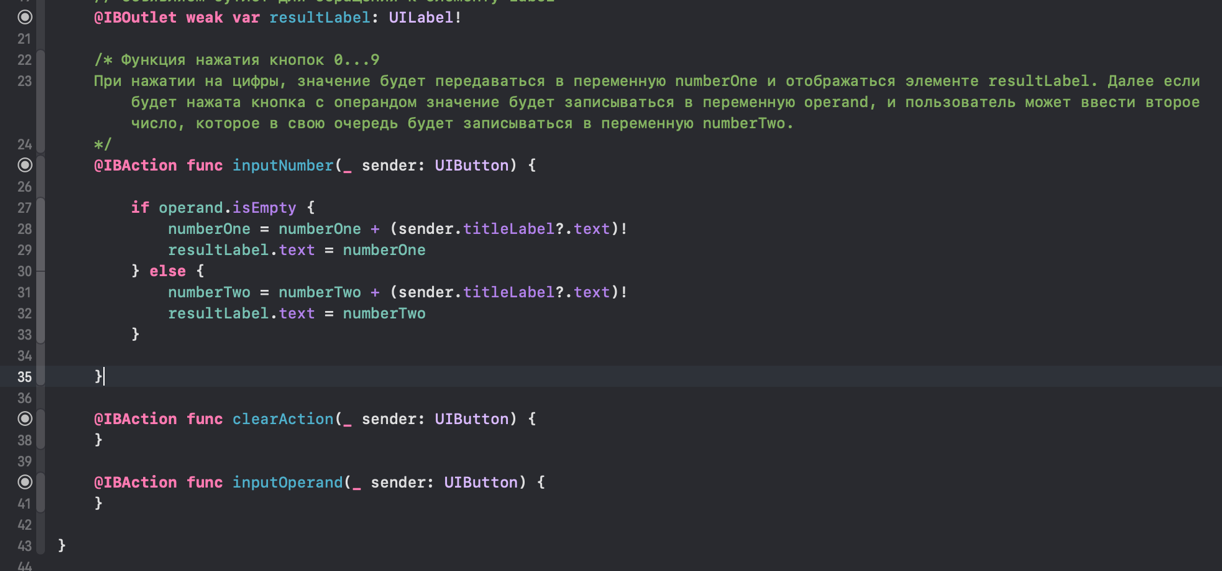
* В X Code если имя функции сменилось все связи будут потеряны и их необходимо пересоздать, если у какой то кнопки необходимо удалить связь то нажмите на кнопку с зажатой клавишей control и в открывшемся списке удалите нужную связь



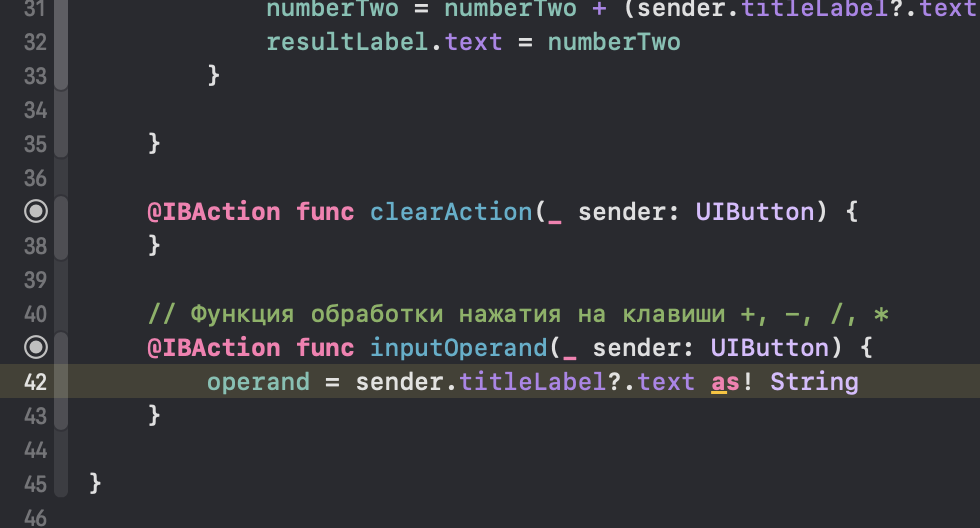
* Создайте функцию для кнопки AC с именем clearAction и функцию для операндов с именем inputOperand



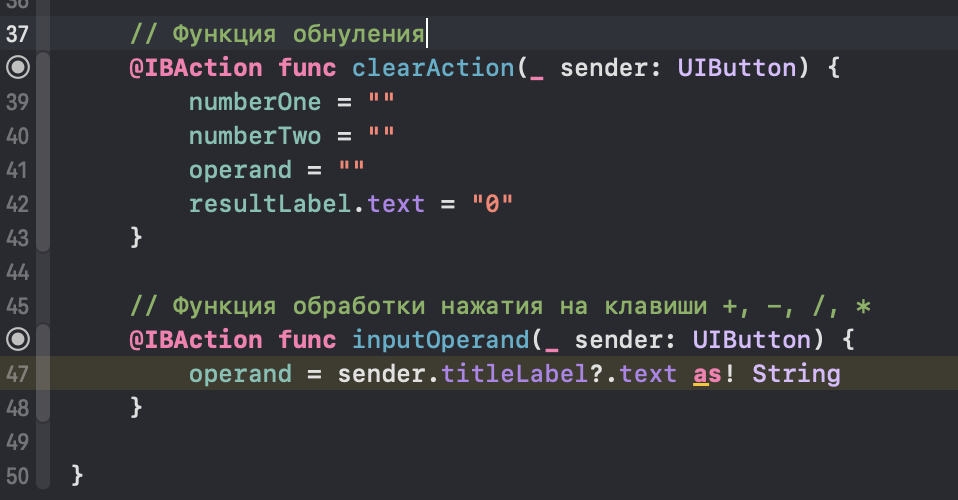
* Логика приложения будет следующая при нажатии на цифры, значение будет передаваться в переменную numberOne и отображаться элементе resultLabel. Далее если будет нажата кнопка с операндом значение будет записываться в переменную operand, и пользователь может ввести второе число, которое в свою очередь будет записываться в переменную numberTwo. После нажатия на кнопку равно будет выводит итог в resultLabel
* Приступим к реализации проекта. В функции inputNumber будет использоваться условие если переменная operand пусто, то введеное значение записывается в переменную numberOne



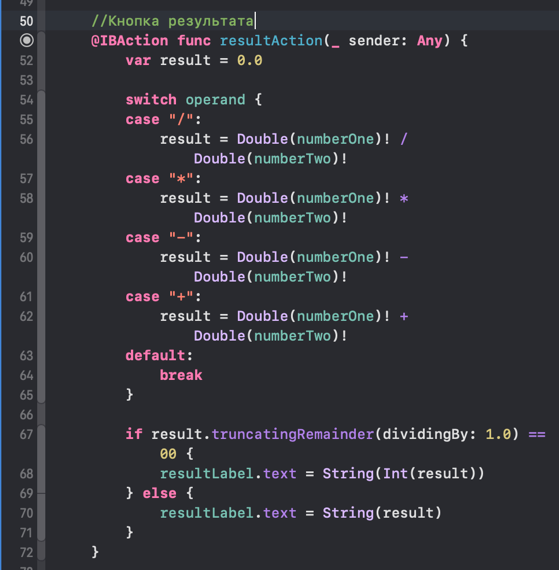
* Далее сделаем обработку нажатия на клавиши операндов



* Создайте функцию для обнуления переменных и Label

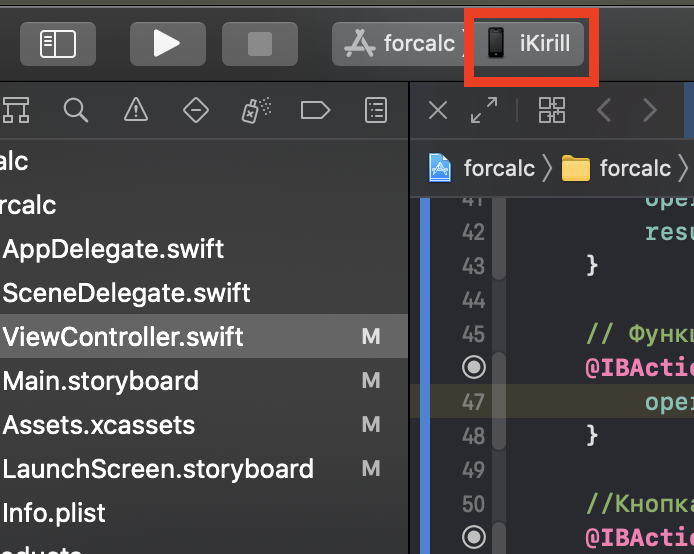


* Создайте новую функцию для отображения результата с именем resultAction. В ней будет создана переменная result с типом данных Double которая будет хранить результат. С помощью конструкции switch проверяем переменную operand и в соответствии с хранимым значением будет выполняться один из кейсов. Кейсы созданы для каждого операнда и выполняют соответствующее арифметическое действие. Далее результат выводится в resultLabel.



# Запуск проекта

* В X Code для проверки проекта и тестирования реализован Simulator в нем можно просмотреть свой проект на виртуальном устройстве. Для этого в левой части экрана выберите устройство которое будет запущено.



* После выбора устройства нажмите на кнопку Play. Если в проекте не было запущено ошибок, приложение будет запущено на Simulator.

