**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

**ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ**

**«КОЛЛЕДЖ ПРЕДПРИНИМАТЕЛЬСТВА № 11»**

|  |  |
| --- | --- |
| УТВЕРЖДАЮ  Директор ГАПОУ КП №11  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шептуха  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |  |

**ПРОГРАММА**

профессиональной пробы

в рамках проекта по ранней профессиональной ориентации

учащихся 6-11 классов общеобразовательных организаций

«Билет в будущее»

по профессиональному направлению

«Разработчик мобильных приложений»

2021 год

**Разработчик мобильных приложений**

Автор программы: Диянов Владислав Васильевич, Преподаватель, эксперт WorldSkills Russia по компетенции «Разработка мобильных приложений».

Контакты автора: Москва, vladdiya01@gmail.com, +79775204660

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Уровень сложности | Формат проведения | Время проведения | Возрастная категория | Доступность для участников с ОВЗ |
| базовый | очная | 90 | 10-11  класс | - допустимые нозологические группы:   1. Общие заболевания (нарушение дыхательной системы, пищеварительной, эндокринной систем, сердечно-сосудистой системы и т.д.) 2. Тяжёлые нарушение речи   - необходимые специальные условия: присутствие на занятии сопровождающего, знающего особенности участника профессиональной пробы и умеющего с ними работать.  - возможность проведения пробы в смешанных группах «участники без ОВЗ + участники с ОВЗ» |

1. **Содержание программы**

**Введение (10 мин)**

1. Краткое описание профессионального направления

Разработчик мобильных приложений занимается проектированием и разработкой приложений для мобильных устройств: смартфонов, планшетов, умных часов, а также систем Smart TV. Для разработки используется различное программное обеспечение, выбор которого зависит от целевой операционной системы. В настоящий момент наиболее распространенными являются операционные системы iOS и Android. Для iOS наиболее распространенная среда разработки является XCode, а для Android – Android Studio

1. Место и перспективы профессионального направления в современной экономике региона, страны, мира

Информационные технологии в целом, и программирование, в частности, развиваются очень активно. Разработка мобильных приложений является одной из наиболее быстроразвивающихся отраслей, поскольку популярность портативных устройств только растет. Смартфоны и планшеты используются не только для игр и досуга, но и для занятий спортом, приобретения товаров, ведения бизнеса, общения, образования, удовлетворения многих других потребностей современного человека.

Повсеместное внедрение мобильных технологий делает данную профессию очень перспективной.

В текущий момент на рынке труда существует нехватка специалистов данного направления. Уровень заработной платы немного выше в сравнении с разработчиками других направлений (backend, frontend). Средняя заработная плата разработчика мобильных приложений составляет 100 тысяч рублей.

1. Необходимые навыки и знания для овладения профессией

Ключевыми навыками разработчика мобильных приложений являются умение программировать, знание принципов построения удобного пользовательского интерфейса, внимательность. Владение английским языком также является важным навыком любого разработчика, поскольку вся документация публикуется на английском языке.

1. 1-2 интересных факта о профессиональном направлении

Еще несколько лет назад использовались простые мобильные приложения. Раньше, чтобы провести любую операцию с деньгами, приходилось идти в банк, но сейчас мы можем сделать практически любую операцию, даже смену Пин-кода карты через мобильное приложение.

Первые соревнования по компетенции “Разработка мобильных приложений” в рамках чемпионата WorldSkills были проведены в России в 2017 году. В 2019 году компетенция была представлена на чемпионате мира в Казани. Участники чемпионата должны были разработать приложения для участия в квестах для планшетов, часов и TV.

1. Связь профессиональной пробы с реальной деятельностью

В рамках данной пробы участнику необходимо разработать небольшое мобильное приложение для смартфона под управлением операционной системы iOS. Из аналогичных задач и состоит деятельность разработчика мобильных приложений.

**Постановка задачи (5 мин)**

1. Постановка цели и задачи в рамках пробы

Участникам необходимо разработать свое мобильное приложение.

1. Демонстрация итогового результата, продукта

Участники демонстрируют рабочее мобильное приложение.

**Выполнение задания (55 мин)**

Для выполнения задания необходимо использовать среду разработки XCode версии 12.0 или новее. Язык программирования – Swift.

Настройка рабочего пространства:

1. Запустите XCode.

Создание приложения:

1. Создайте iOS-проект с iOS типа App.

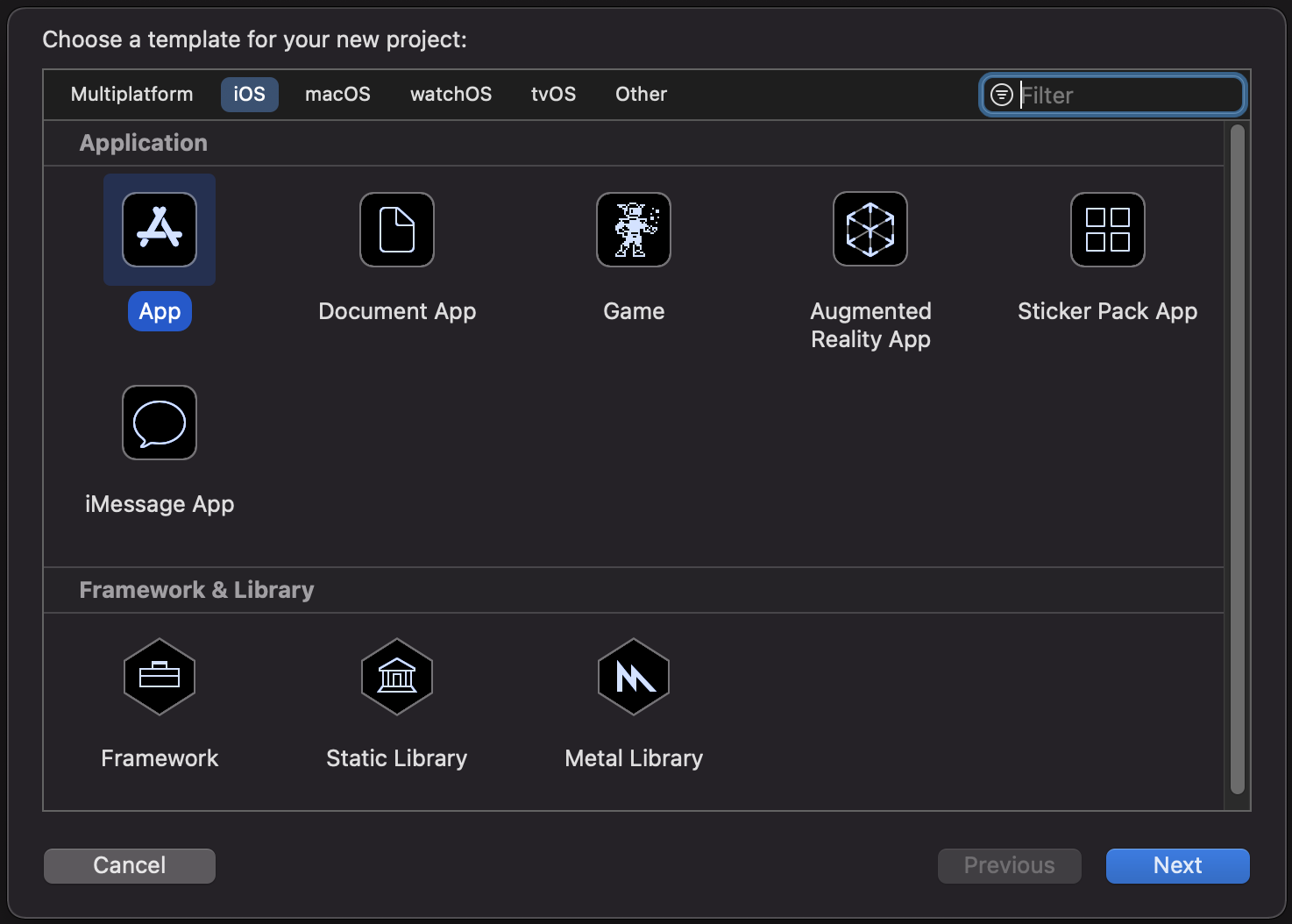
Для этого в левом верхнем углу экрана выберите File -> New -> Project

Рисунок 1 – Создание проекта. Шаг 1.

Выберите App и нажмите кнопку Next.

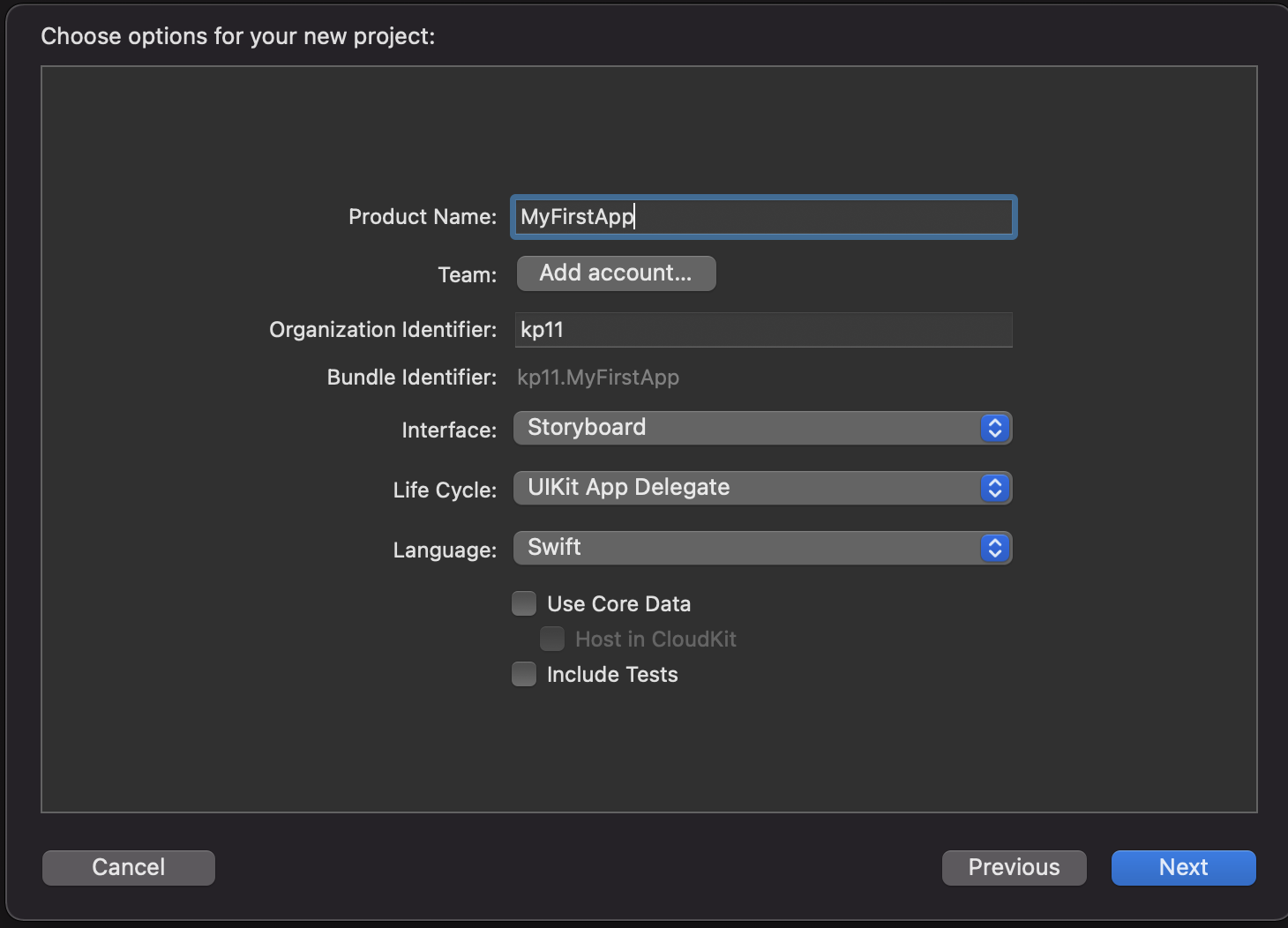
****

Рисунок 2 – Создание проекта. Шаг 2.

Заполните форму по образцу. После заполнения нажмите кнопку Next. В следующем окне нажмите кнопку Create.

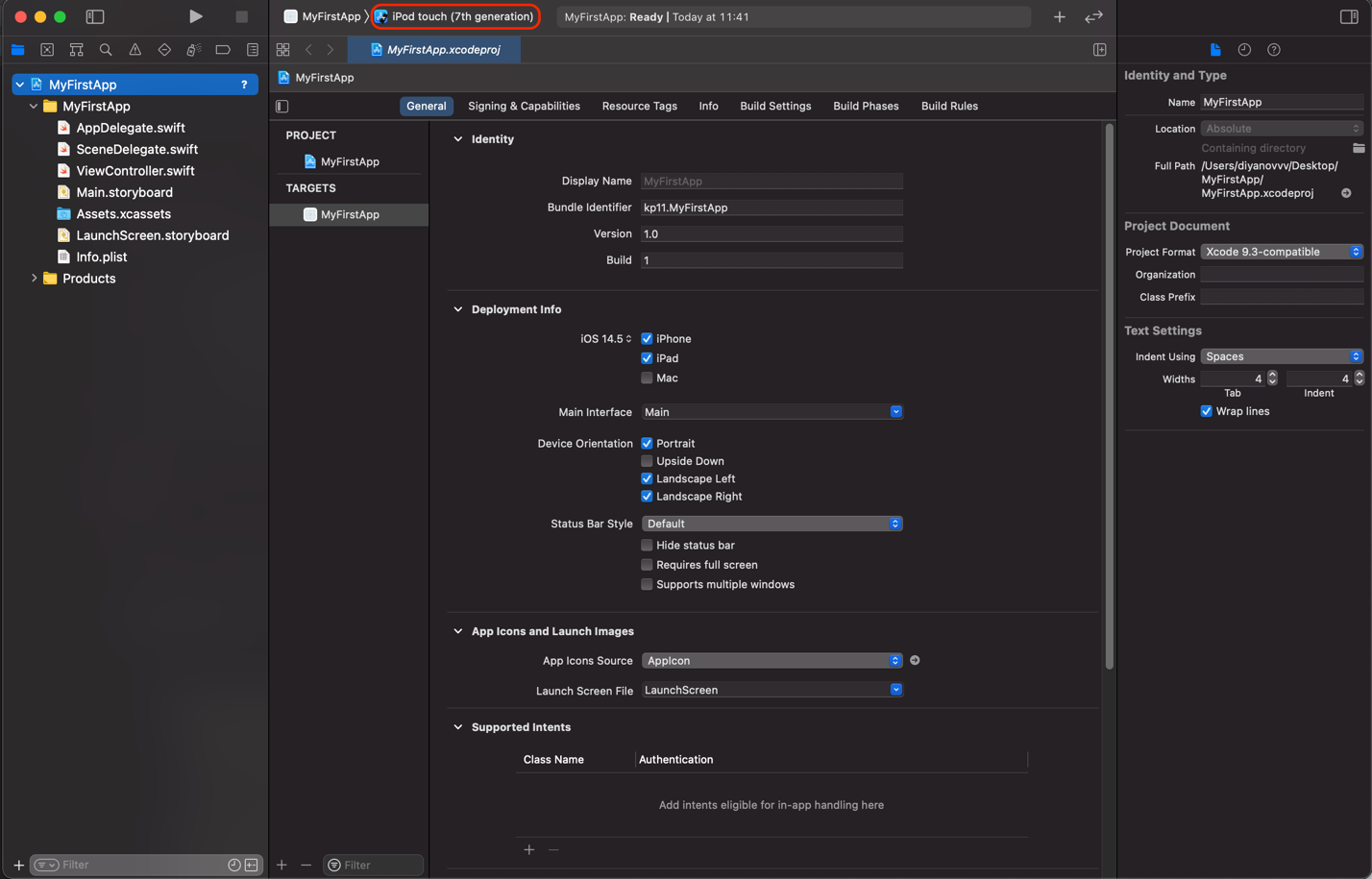


Рисунок 3 – Выбор эмулятора.

В верхней части программы, нажмите на выделенную область, и выберите симулятор iPhone 11.

1. В левой части экрана расположена структура проекта. Откройте файл Main.storyboard расположенные в корне проекта.

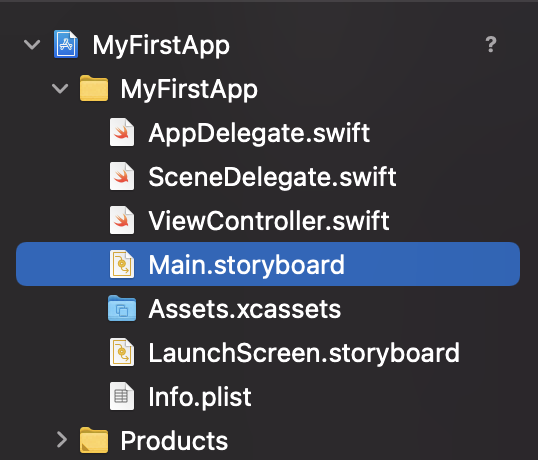
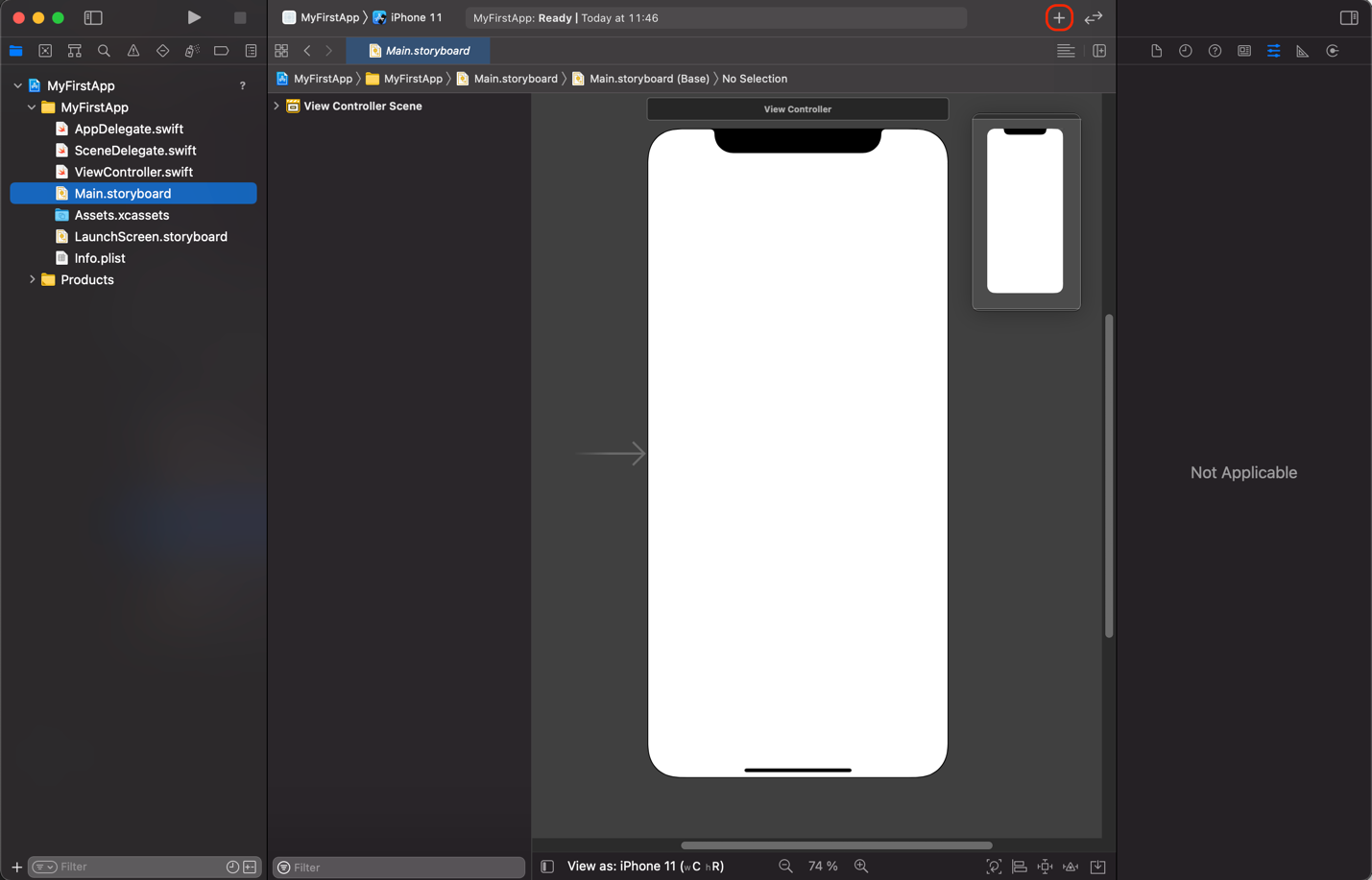


Рисунок 4 – Структура проекта.

В верхней правой части программы найдите значок «+», который отобразит все элементы для создания мобильного приложения. И нажмите на него:



Выберите в открывшемся списке элемент «Label» и перетащите его на экран.

Выберите элемент Label, который вы перетащили на экран, и в правой части измените свойство «Alignment», чтобы текст был по центру.

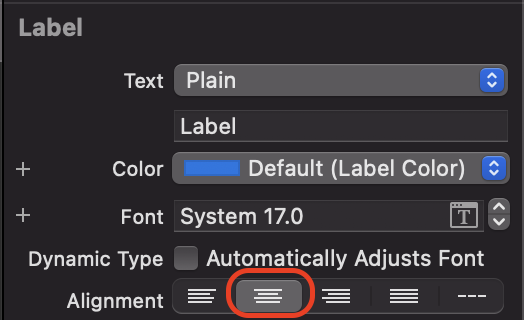


Рисунок 5 – Свойство Label.

Теперь необходимо переключиться на режим «Size Inspector». В данном режиме мы можем изменить размеры элемента и выставить его так, как нам нужно.



Рисунок 6 – Size inspector.

В правой части экрана замените значения на те, которые приведены на Рисунке 7.

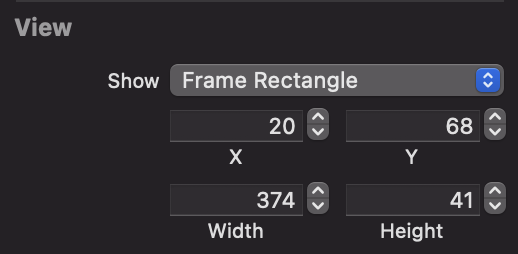
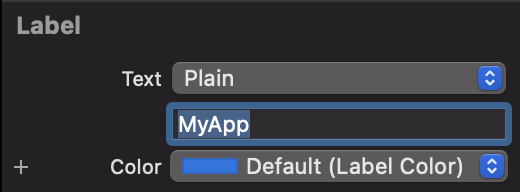


Рисунок 7 – Местоположение Label

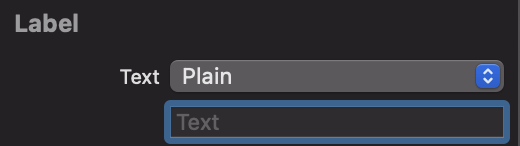
Возвращаемся обратно в режим  «Attributes Inspector», в этом режиме мы поставили текст по центру.

Теперь заменим текст «Label» на «MyApp». Для этого нам необходимо заменить текст в данном окне и нажать Enter.



Добавим на наш экран еще один элемент Label. Ставим тексту отображение по центру, как на рисунке 5.

Изменим его название на пустое значение и нажимаем Enter.



Переходим в режим Size Inspector, как на Рисунке 6. И меняем данные на те, что на рисунке ниже.

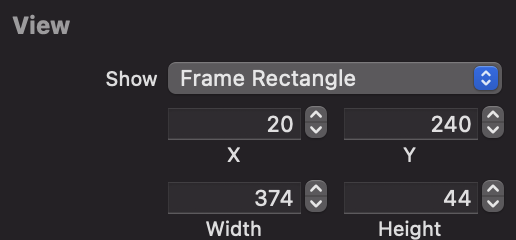
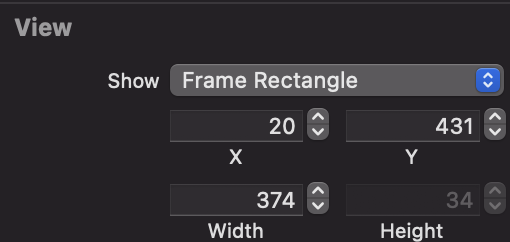


Рисунок 8 – Размеры 2го Label.

И добавим еще один элемент textField. Найдем его среди элементов, и перетянем на наш экран



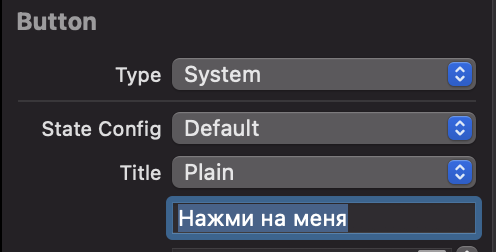
Теперь изменим свойства данного элемента, перейдем в режим Size Inspector. И выставляем такие же значения, что на на рисунке ниже



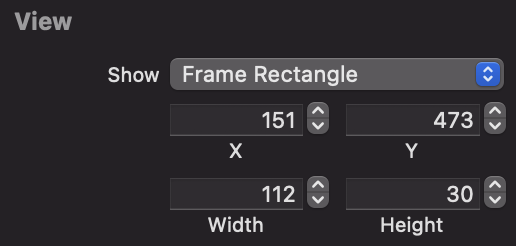
Также добавим еще кнопку, на наш экран. Выбираем в списке элементов, элемент «Button».



В свойствах кнопки, в режиме Attribute Inspector изменяем текст кнопки.



Переходим в свойства Size Inspector и меняем местоположение кнопки, на данные ниже.



1. Теперь после того как мы добавили элементы на экран, мы можем запустить проект и посмотреть, что у нас получилось. Нажимаем в левой верхней части программы кнопку «Build and Run».

Теперь нам необходимо привязать наши элементы к коду. Для этого мы нажимаем на наш экран, как на рисунке ниже.

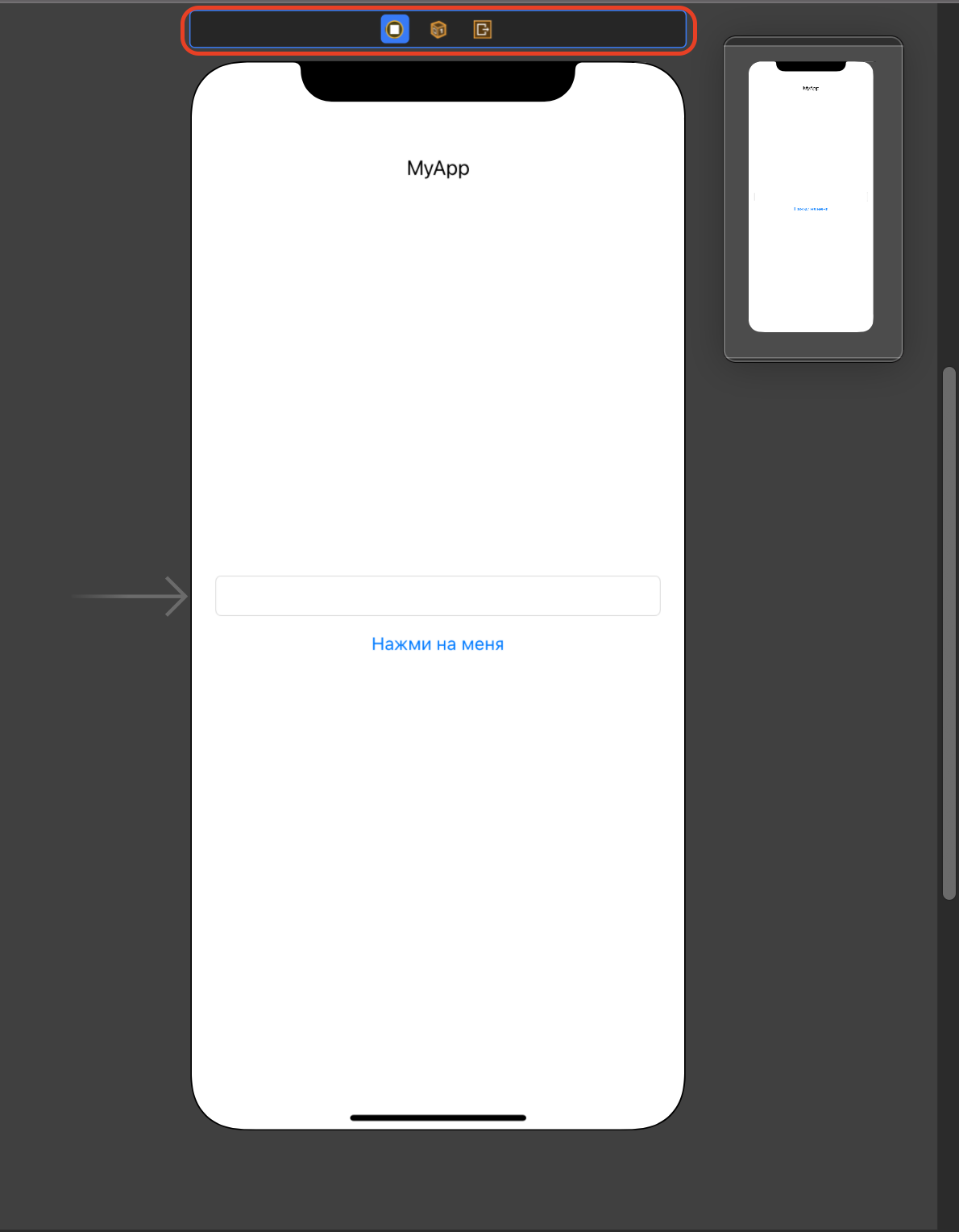


Рисунок 9 – Выделение экрана.

И нажимаем на кнопку  сверху программы.

И выбираем свойство Assistant

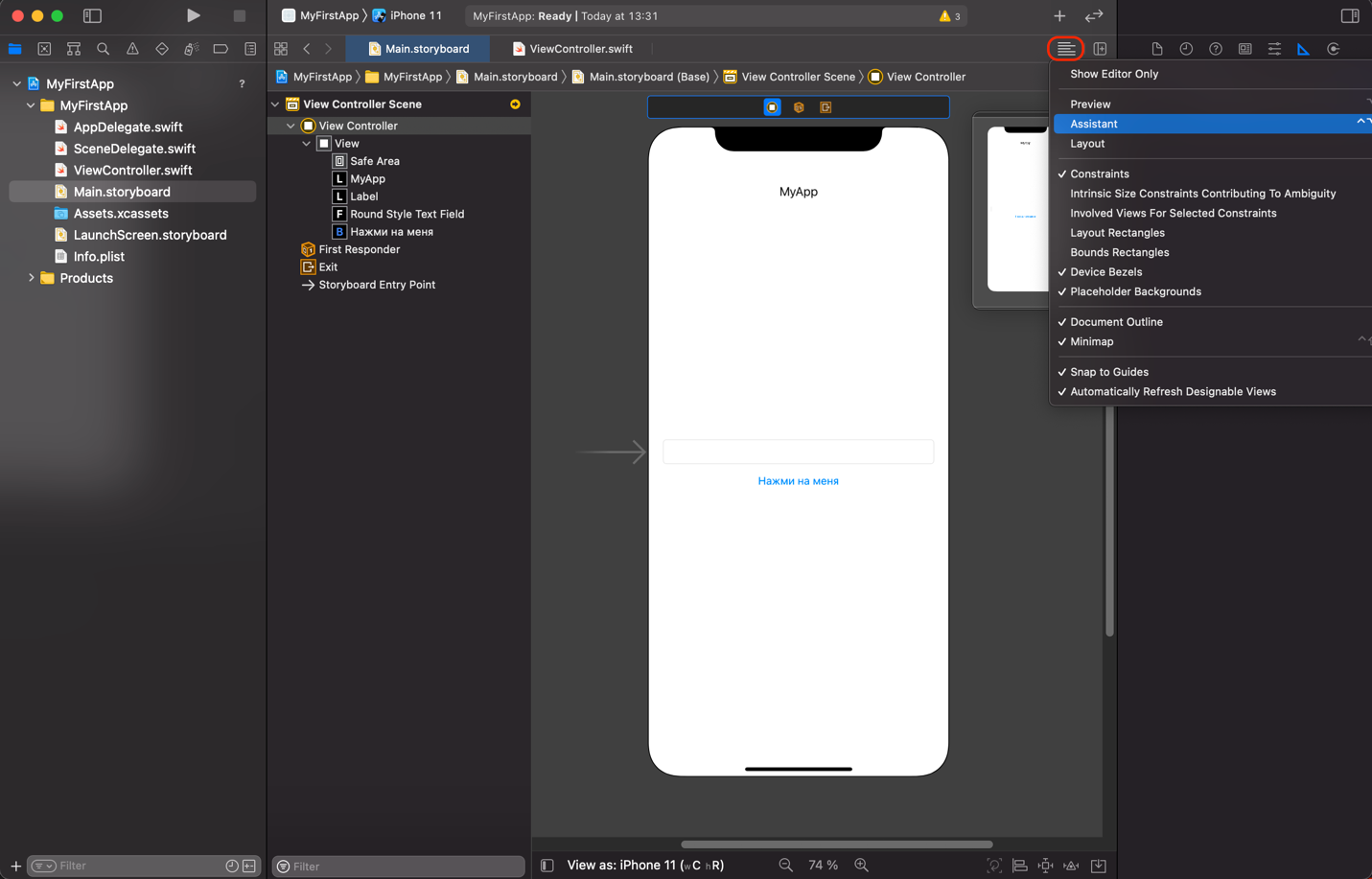


Рисунок 10 – Свойство Assistant.

Таким образом мы видим экран и код, который относится к данному экрану.

1. Теперь привязываем элементы. Для этого нам нужно выбрать нужный нам элемент в левой части экрана и перетянуть этот элемент в код правой кнопкой мыши.

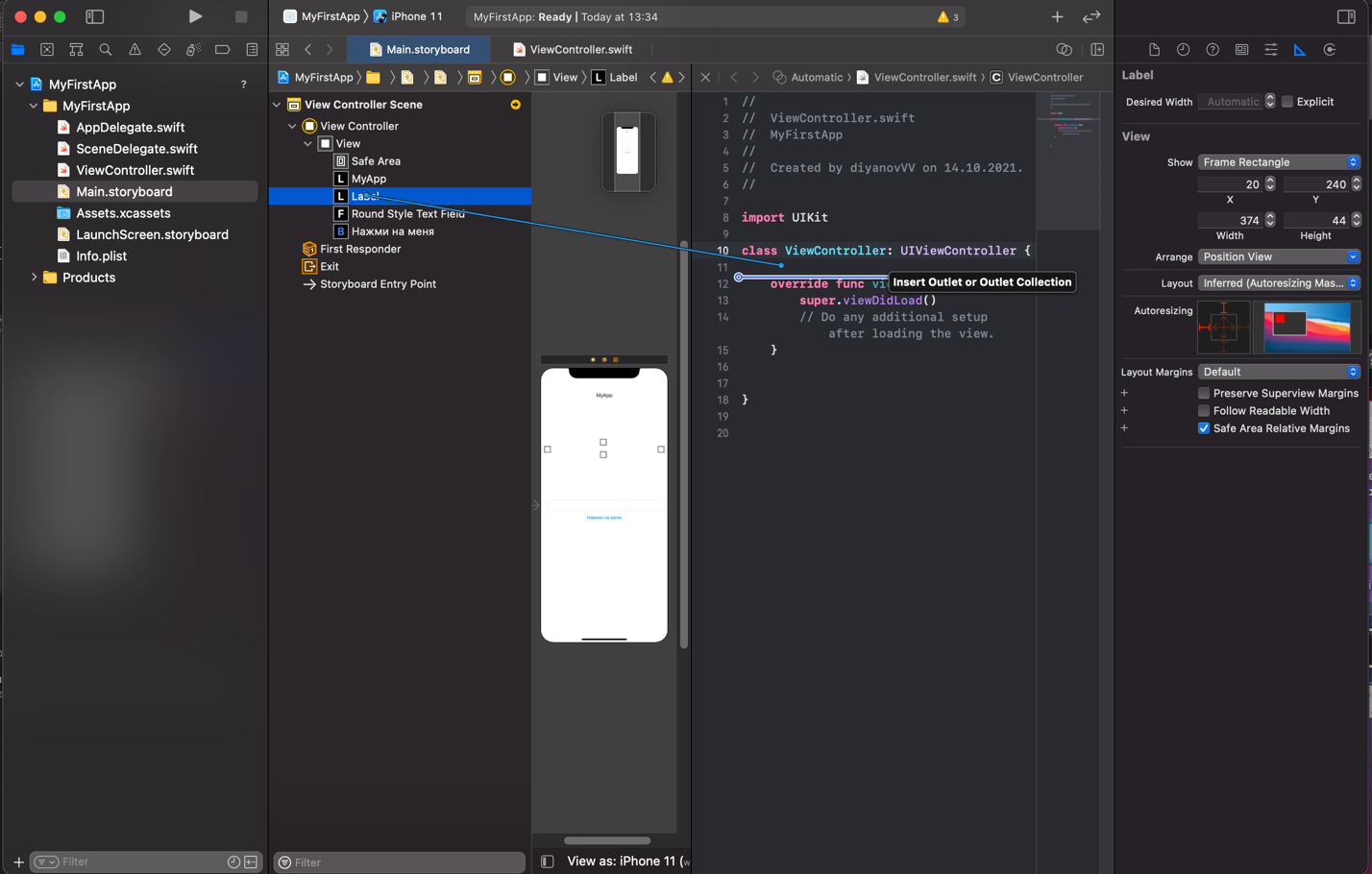
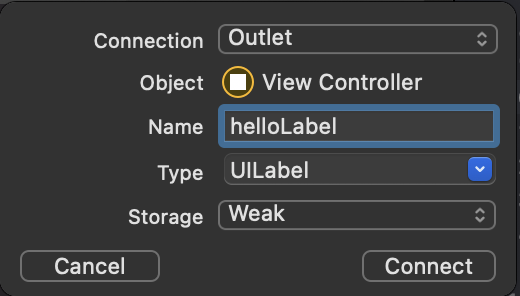


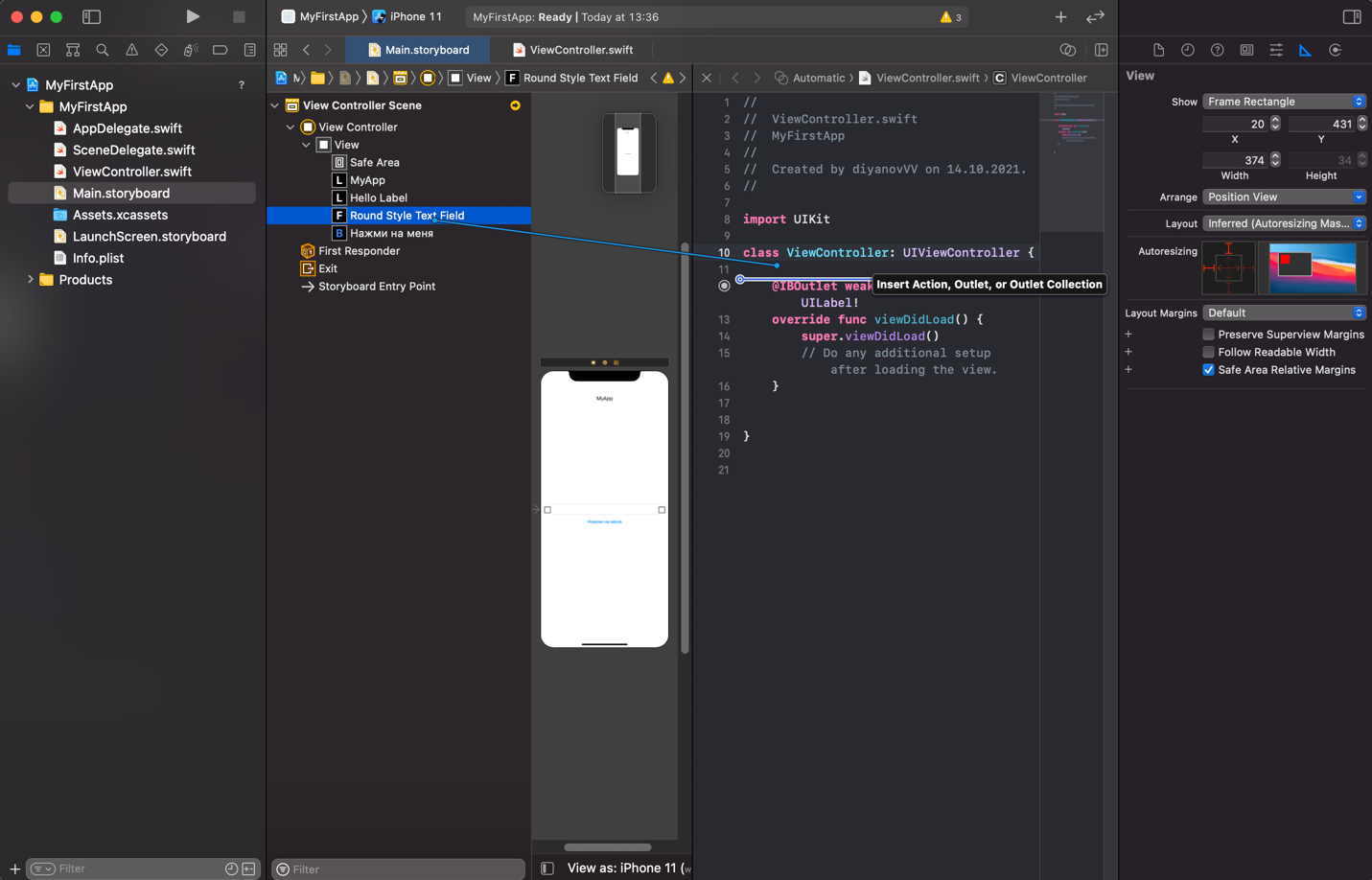
Рисунок 11 – Подключение элемента к коду.

В открывшемся окне, вписываем следующий текст



Поле Name отвечает за название данного элемента в коде, чтобы мы могли к нему обращаться и менять его свойства. После нажимаем кнопку «Connect».

Теперь перетягиваем таким же способом элемент Round Style Text Field.



И даем ему название nameTextField. После нажимаем кнопку «Connect».

А кнопку «Нажми на меня» нам необходимо подключить таким же способом, только изменить его свойство, и место подключения. Точно таким способом перетягиваем кнопку, только теперь перед последней фигурной скобкой }.

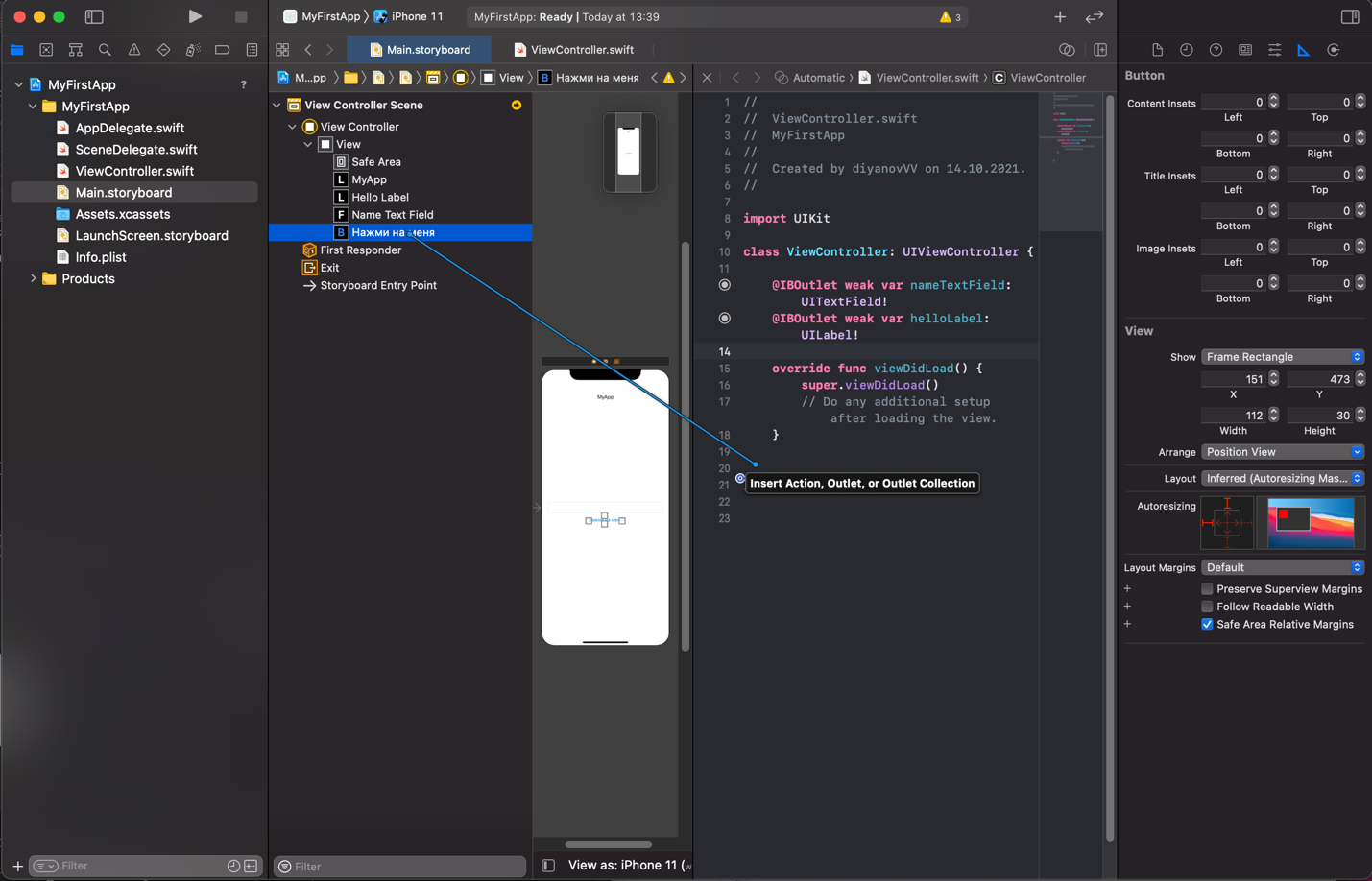


Рисунок 12 – Подключение кнопки.

Таким способом мы подключаем действие для кнопки, чтобы после нажатия на нее что-то происходило. И ставим такие же свойства и название как на рисунке ниже. После изменения свойств, нажимаем кнопку «Connect».

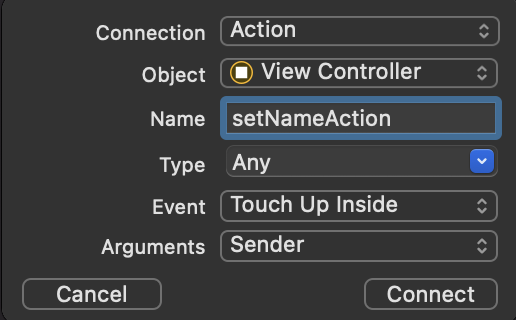
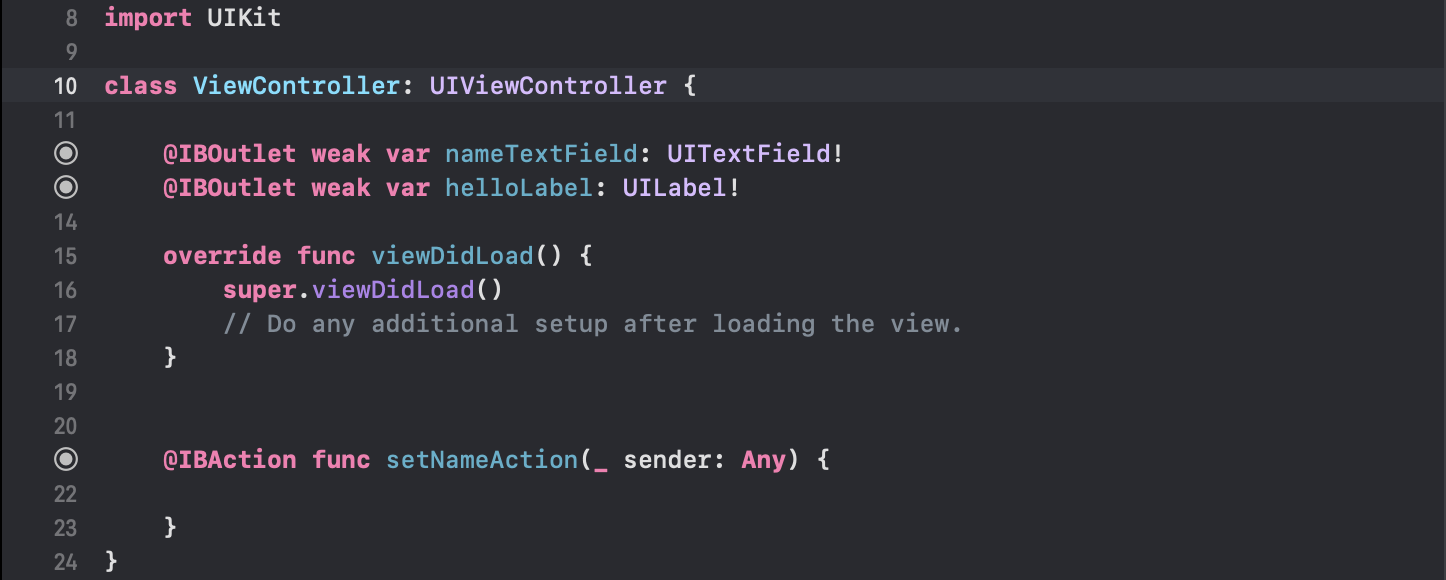
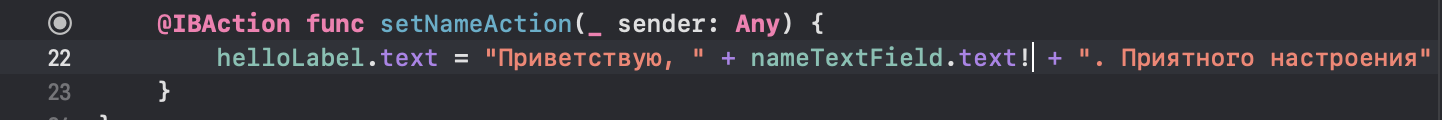


Рисунок 13 – Свойства подключения кнопки.



В результате мы получаем следующее. 2 переменных наших элементов TextField и Label, а также действие на кнопку.

1. Теперь нам нужно прописать действие, которое будет срабатывать по нажатию на кнопку. Для этого, внутри нашей функции, мы прописываем следующее:



Свойство у переменной «helloLabel» .text, отвечает за изменение текста в данном элементе.

1. Теперь можем запустить наш проект, и посмотреть, что у нас получилось.
2. Открываем файл Main.storyboard

Добавляем на наш экран новый элемент, imageView. Данный элемент отображает картинки, добавленные в наш проект.

Чтобы отобразить картинку, нам необходимо в корне проекта открыть папку Assets.xcassets

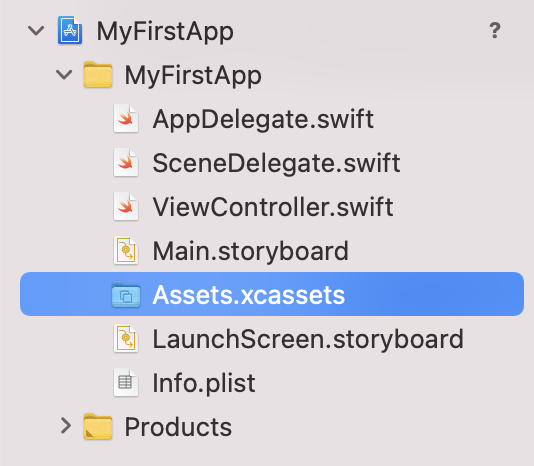


Рисунок 14 – папка Assets.xcassets

Теперь нам необходимо добавить сюда 2 наших картинки, которые вы можете найти в папке с заданием. Выделяем картинки и перетаскиваем их в окно, выделенное ниже на рисунке.

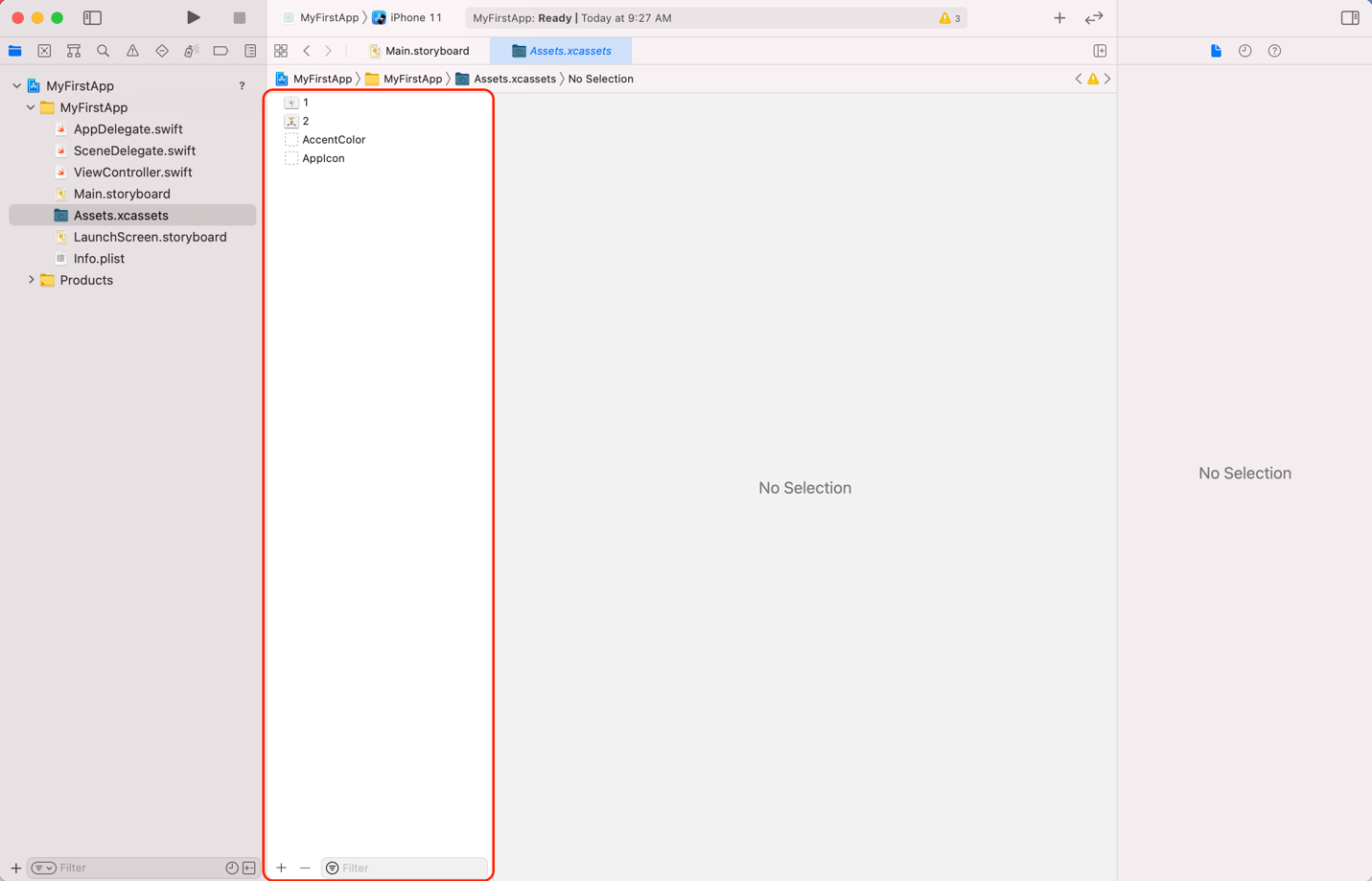
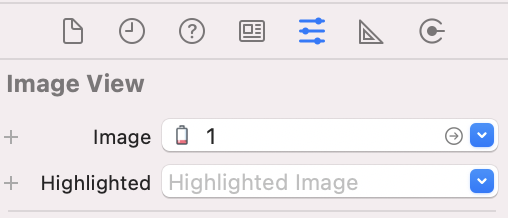


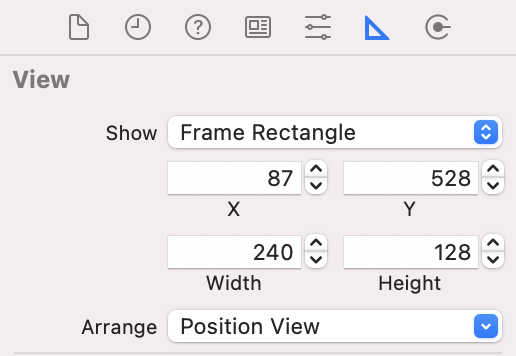
Рисунок 15 – Добавление изображений в проект.

Теперь возвращаемся в файл Main.storyboard и нажимаем на элемент ImageView.

В режиме Attributes Inspector меняем изображение, выбрав нужную нам картинку в списке.



Переходим в режим Size Inspector и выставляем свойства, как на рисунке ниже.



Теперь привяжем элемент ImageView, таким же способом, как мы привязывали элементы Label и textField. С названием «myImageView».

После переходим в файл ViewController.swift

В функции setNameAction мы сделаем проверку, пустое ли значение в поле для ввода. Если да, то менять мы ничего не будет, а если не пустое, тогда отображаем и меняем картинку.

Для этого нам нужно написать следующее

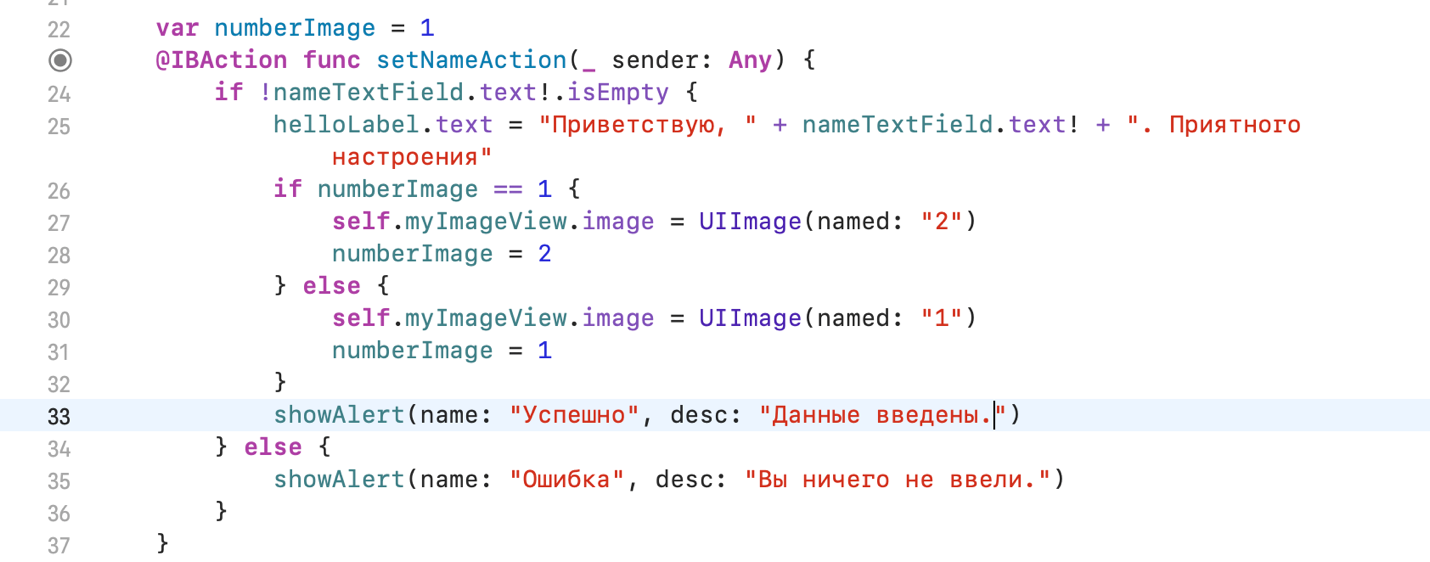


Теперь можем запускать наш проект, и смотреть что у нас получилось.

Внесем небольшие поправки в наше приложение. Когда в поле для ввода ничего не ввели, нам нужно выводить пользователю всплывающее окно, которое оповестит его об ошибке. Для этого нужно прописать следующее. Мы создаем отдельную функцию, которая будет отвечать за всплывающее окно.

****

Функция showAlert не будет работать, пока мы ее не вызовем. Для того, чтобы ее вызвать, нам необходимо в функции setNameAction прописать следующее.



Теперь можем запускать наш проект, и смотреть, что у нас получилось

**Контроль, оценка и рефлексия (20 мин)**

1. Критерии успешного выполнения задания

Мобильное приложение создано полностью – 100%

Мобильное приложение создано частично (% выполнения задания равен проценту выполненной работы)

1. Рекомендации для наставника по контролю результата, процедуре оценки

Во время выполнения участниками профессиональной пробы стоит контролировать создание мобильного приложения, и подсказывать в случае технических ошибок.

1. Вопросы для рефлексии учащихся

Какие среды разработки используются для создания мобильных приложений под iOS и Android?

В каких сферах жизни используются мобильные приложения?

Какие навыки необходимы разработчику мобильных приложений?

1. **Инфраструктурный лист**

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Рекомендуемые технические характеристики с необходимыми примечаниями** | **Количество** | **На**  **на 1 чел.** |
| ПК | iMac  процессор: intel core i5 и 2.4ghz или выше  оперативная память: 8 gb или выше  ос: macOS  свободное место на диске: 10 gb  выделенная видео память: 2 gb | 1 | 1 |
| Компьютерная мышь | Проводная / беспроводная | 1 | 1 |
| Компьютерная клавиатура | Проводная / беспроводная | 1 | 1 |
| Программное обеспечение | Программа: XCode 12.0 или выше | 1 | 1 |

1. **Приложение и дополнения**

|  |  |
| --- | --- |
| Ссылка | Комментарий |
| https://apps.apple.com/ru/app/xcode/id497799835?mt=12 | Ссылка для загрузки актуальной версии приложения XCode. Оплата и регистрация не требуется. |
| https://swiftbook.ru/contents/doc/ | Учебник по языку Swift для начинающих и продвинутых. |