

Лабораторная работа № 7. Разработка простых мобильных приложений на языке Swift и Objective-C

Содержание

Лабораторная работа № 7. Разработка простых мобильных приложений на языке Swift и Objective-C.....	1
ВВЕДЕНИЕ.....	3
СТРУКТУРА ЛАБОРАТОРНОЙ РАБОТЫ.....	3
ПОРЯДОК СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ.....	3
ТРЕБОВАНИЯ К ОТЧЁТУ И ОФОРМЛЕНИЮ КОДА.....	5
КРИТЕРИИ ОЦЕНИВАНИЯ.....	6
РЕГИСТРАЦИЯ РАЗРАБОТЧИКА.....	6
ЗАДАНИЕ 1. ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ФОНОМ НА ЯЗЫКЕ SWIFT	7
ЗАДАНИЕ 2. ПОКАЗ ПОГОДЫ И ДРУГИХ СВЕДЕНИЙ О ГОРОДЕ НА ЯЗЫКЕ OBJECTIVE-C.....	12
упражнение 2.1.....	12
упражнение 2.2.....	12
ЗАДАНИЕ 3. ПРИЛОЖЕНИЕ ДЛЯ РАСЧЕТА КОЛИЧЕСТВА КАЛОРИЙ И ИНДЕКСА МАССЫ ТЕЛА НА ЯЗЫКЕ SWIFT.....	18
ЗАДАНИЕ 4. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ С ИСПОЛЬЗОВАНИЕМ SWIFTUI НА ЯЗЫКЕ SWIFT.....	31
ЗАДАНИЕ 5. ПРОСТОЕ МОБИЛЬНОЕ ПРИЛОЖЕНИЕ НА ЯЗЫКЕ SWIFT....	32
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	32
Варианты.....	33
Вариант 1.....	33
Вариант 2.....	33
Вариант 3.....	33
Вариант 4.....	33
Вариант 5.....	34

Вариант 6.....	34
Вариант 7.....	34
Вариант 8.....	34
Вариант 9.....	35
Вариант 10.....	35
Вариант 11.....	35
Вариант 12.....	35
Вариант 13.....	36
Вариант 14.....	36
Вариант 15.....	36
Вариант 16.....	36
Вариант 17.....	37
Вариант 18.....	37
Вариант 19.....	37
Вариант 20.....	37
Вариант 21.....	37
Вариант 22.....	38
Вариант 23.....	38
Вариант 24.....	38
Вариант 25.....	38
Вариант 26.....	39
Вариант 27.....	39
Вариант 28.....	39
Вариант 29.....	39
Вариант 30.....	40
Вариант 31.....	40
Вариант 32.....	40
Вариант 33.....	40
Вариант 34.....	41
Вариант 35.....	41

Вариант 36.....	41
Вариант 37.....	41
Вариант 38.....	41
Вариант 39.....	42
Вариант 40.....	42

ВВЕДЕНИЕ

Данные методические указания предназначены для выполнения лабораторных работ по дисциплине «Технологии программирования» студентами 2 курса специальности «Прикладная информатика».

СТРУКТУРА ЛАБОРАТОРНОЙ РАБОТЫ

Каждая лабораторная работа содержит тексты задач и контрольные вопросы, ответы на которые проверяются преподавателем при приеме работы у студента. Для некоторых задач приводятся указания по их решению.

Каждая задача имеет уникальный в пределах данной лабораторной работы номер. В ссылках на задачу указывается номер задачи и в скобках номер страницы. Например, «задача 5 (см. с. 33)» означает пятую задачу, описание которой приведено на странице 33.

Список литературы [1–6] содержит библиографические ссылки на источники, рекомендуемые к прочтению при решении задач.

В приложение вынесены требования и рекомендации по оформлению исходных текстов программ.

ПОРЯДОК СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

Выполнение студентом лабораторной работы и сдача ее результатов преподавателю происходит следующим образом.

1. Студент выполняет разработку программ.

В ходе разработки студент обязан следовать указаниям к данной задаче (в случае их наличия). Исходные тексты программ следует разрабатывать в соответствии с требованиями к оформлению, приведёнными в приложении.

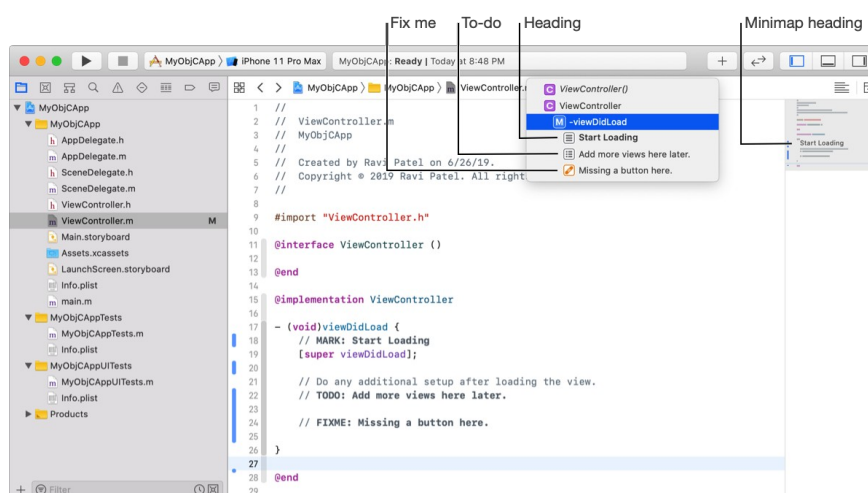
В частности, с помощью аннотаций (комментариев) добавить пояснения к коду или другую полезную информацию, такую как задачи, напоминания об исправлении ошибок и заголовки разделов, которые доступны во всплывающем меню при просмотре структуры проекта в меню навигации (см. рис. ниже).

Для добавления задачи вставьте комментарий с префиксом TODO: — например (см. рис. ниже): // TODO: [your to-do item].

Чтобы добавить напоминание об исправлении ошибки: вставьте комментарий с префиксом FIXME: — например (см. рис. ниже): `// FIXME: [your bug fix reminder]`.

Чтобы добавить заголовок для блока кода, вставьте комментарий с префиксом MARK: — например (см. рис. ниже): `// MARK: [your section heading]`. На миникарте блоков кода также появится заголовок.

В код можно добавлять строку-разделитель выше или ниже комментария. Чтобы добавить разделитель выше, добавьте дефис (-) перед частью комментария аннотации, например: `// MARK: - [your content]`. Чтобы добавить разделитель ниже, добавьте дефис (-) после комментария к аннотации, например: `// MARK: [your content] -`.



2. Студент выполняет самостоятельную проверку исходного текста каждой разработанной программы и правильности её работы, а также свои знания по теме лабораторной работы.

Исходные тексты программ должны соответствовать требованиям к оформлению, приведенным в приложении. Недопустимо отсутствие в тексте программы следующих важных элементов оформления: спецификации программного файла и подпрограмм, а также отступов, показывающих структурную вложенность языковых конструкций.

Для проверки правильности работы программы студенту необходимо разработать набор тестов и на их основе провести тестирование программы. Тестовый набор должен включать примеры входных и выходных данных, приведенные в тексте задачи, а также тесты, разработанные студентом самостоятельно.

Самостоятельная проверка знаний по теме лабораторной работы выполняется с помощью контрольных вопросов и заданий, приведенных в конце текста лабораторной работы.

3. Студент защищает разработанные программы. Защита заключается в том, что студент должен ответить на вопросы преподавателя, касающиеся разработанной программы, и контрольные вопросы.

К защите необходимо представить исходные тексты программ, оформленных в соответствии с требованиями, и **протоколы тестирования** каждой программы, подтверждающие правильность ее работы.

Протокол тестирования включает в себя тест (описание входных данных и соответствующих им выходных данных), описание выходных данных, полученных при запуске программы на данном тесте, и отметку о прохождении теста. Тест считается пройденным, если действительные результаты работы программы совпали с ожидаемыми.

ТРЕБОВАНИЯ К ОТЧЁТУ И ОФОРМЛЕНИЮ КОДА

В файле Readme проекта на github должна быть ссылка на отчёт. Отчет опубликовать в репозитории в папке docs или во внешнем хранилище, добавив ссылку на него в файл Readme репозитория.

Репозиторий должен содержать следующие ветки:

1. **main** — файл Readme со ссылкой на отчет и описание веток и выполненных заданий.
2. **feature-task1** — каталог проекта task2, содержащий проект для задачи 2.
3. **feature-task2** — каталог проекта task2, содержащий проект для задачи 2.
4. **feature-task3** — каталог проекта task3, содержащий проект для задачи 3.
5. **feature-task4** — каталог проекта task4, содержащий проект для задачи 4.
6. **feature-task5** — каталог проекта task5, содержащий проект для задачи 5.

Отчет по лабораторной работе содержит тексты задач, протоколы тестирования для задач 1-5.

Исходный код приложений должен быть оформлен согласно руководству стиля в зависимости от языка. В частности, следовать требованиям:

1. Каждый файл проекта должен содержать имя и фамилию автора, группу.
2. Важные фрагмента кода обязательно должны быть помечены аннотациями «// MARK: [your content] -» и другими. См. как оформлять комментарии на стр. 3 в ПОРЯДОК СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ .

1. Требования для оформления кода на языке Swift:

- a) <http://ilya2606.ru/?p=1846>
- b) <https://github.com/RedMadRobot/RMR-swift-style-guide>
- c) <https://github.com/raywenderlich/swift-style-guide>

2. Требования для оформления кода для языка Objective-C:

- a) <https://github.com/DigDes/objective-c-style-guide>
- b) <https://github.com/raywenderlich/objective-c-style-guide>

КРИТЕРИИ ОЦЕНИВАНИЯ

Оценка 4

Выполнены **Задание 1** и **Задание 2 (упражнение 2.1)**. Отчёт содержит описание задачи, протоколы тестирования для каждой задачи и комментарии к основным блокам кода согласно требованиям. Отчёт должен быть опубликован в репозитории или должна быть ссылка на него в файле Readme. Отчёт и исходный код проекта может содержать ошибки. Максимальная задержка на предоставление отчёта, исходного кода проекта, включая публикацию в репозитории и задании курса, а также его защиту — 2 недели.

Оценка 5-6

Выполнены **Задание 1-3**. Отчёт содержит описание задачи, протоколы тестирования для каждой задачи и комментарии к основным блокам кода согласно требованиям. Отчёт должен быть опубликован в репозитории или должна быть ссылка на него в файле Readme. Отчёт и исходный код проекта может содержать ошибки. Максимальная задержка на предоставление отчёта, исходного кода проекта, включая публикацию в репозитории и задании курса, а также его защиту — 1 неделя.

Оценка 7-8

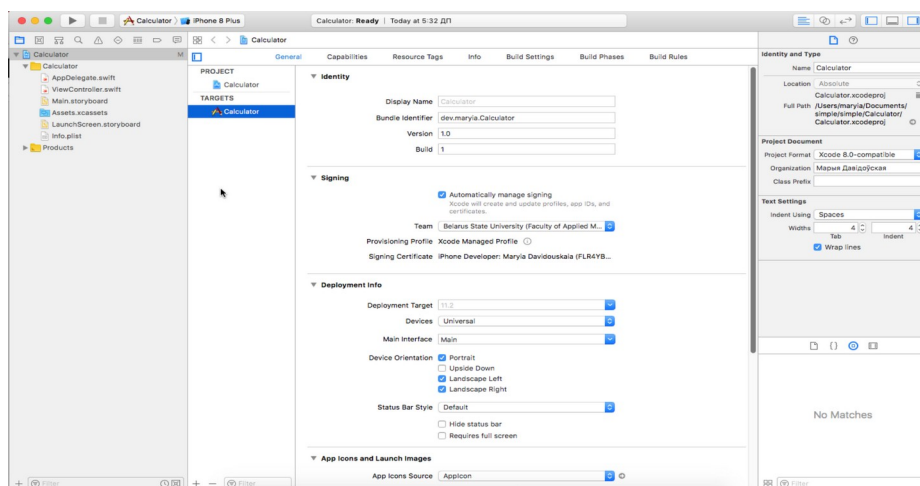
Выполнены **Задание 1-5**. Отчёт содержит описание задачи, протоколы тестирования для каждой задачи, ответы на контрольные вопросы и комментарии к основным блокам кода согласно требованиям. Отчёт должен быть опубликован в репозитории или должна быть ссылка на него в файле Readme. Отчёт и исходный код могут содержать незначительные ошибки. Максимальная задержка на предоставление отчёта, исходного кода проекта, включая публикацию в репозитории и задании курса, а также его защиту — 1 неделя.

Оценка 9

Выполнены **Задание 1-5**. Отчёт содержит описание задачи, протоколы тестирования для каждой задачи, ответы на контрольные вопросы и комментарии к основным блокам кода согласно требованиям. Отчёт должен быть опубликован в репозитории или должна быть ссылка на него в файле Readme. Лабораторная работа опубликована в репозитории и в задании курса в срок, а так же защищена в срок. Не содержит ошибок.

РЕГИСТРАЦИЯ РАЗРАБОТЧИКА

После создания проекта и его сохранения перед вами появится рабочее пространство Xcode.



Если возникает ошибка с регистрацией как разработчика, то авторизуйтесь на сайте <https://developer.apple.com> и проделайте следующие действия:

1. Сгенерируйте сертификат в разделе Certificates.
2. Добавьте идентификатор организации, см. шаг 3, под которым будете регистрировать свои приложения, в разделе Identifier. Например, на скриншоте используется dev.marya.*.
3. Создайте provisioning profile в разделе Provisioning Profile.
4. Добавьте устройство под управлением iOS, задав имя и введя одну латинскую букву, например A или B в начале строки и случайное значение цифр. Сохраните.

ЗАДАНИЕ 1. ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ФОНОМ НА ЯЗЫКЕ SWIFT

Цель:

Создать iOS приложение с использованием объекта “Switch” для изменения фона контроллера. В положении “Вкл” – изображение “bg1.jpg”, в положении “Выкл” – изображение “bg2.jpg”. Изображения “bg1.jpg” и “bg2.jpg” ниже, слева направо соответственно (рис.8). Научиться работать с графическими ресурсами приложения.

В руководстве приведён пример исходного кода для Swift 3. Адаптировать для Swift 5 и, при необходимости, исправить ошибки, связанные с версией языка. В проект можно добавить любые изображения, но перед загрузкой в проект переименовать в bg1.jpg и bg2.jpg.



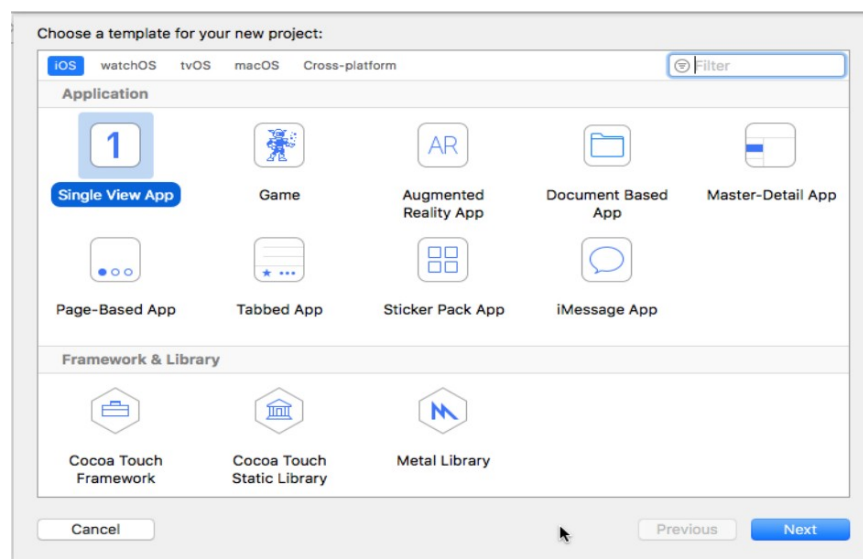
Рис.8. Изображения “bg1.jpg” и “bg2.jpg”

Ход работы:

Создание проекта Xcode.

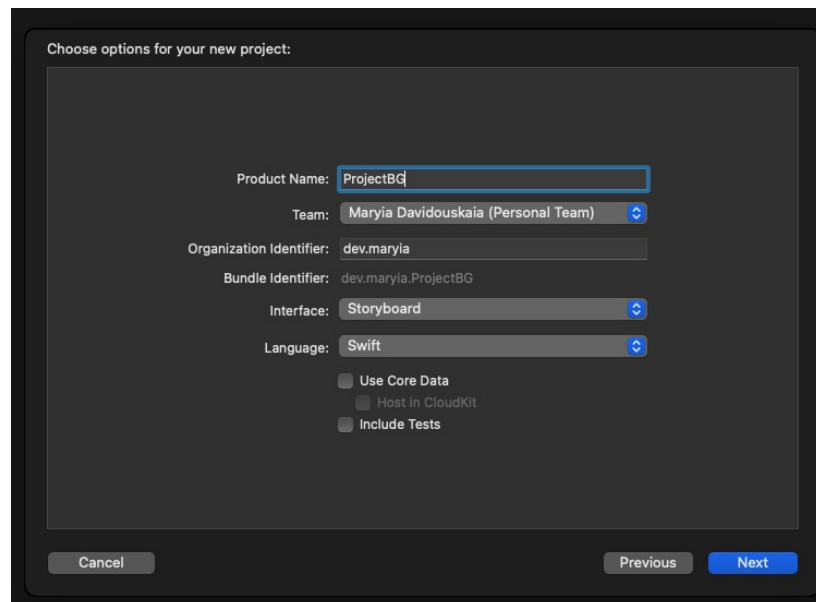
Создаем проект Xcode, аналогично предыдущим лабораторным работам. Убираем галочку Use Core Data, Unit tests, UI Tests и выбираем iOS – Application – Single View Application.

1. Для начала давайте создадим проект: запустите Xcode и нажмите комбинацию клавиш CMD+Shift+N или выберите в меню *File->New->Project*



2. Теперь выберите *Single View Application (App для Xcode 13 и выше)* и нажмите *Next*.

3. На этом этапе надо придумать имя для приложения (Product Name), нажать *Next* и выбрать папку для сохранения. Например, параметры проекта для Xcode 13:

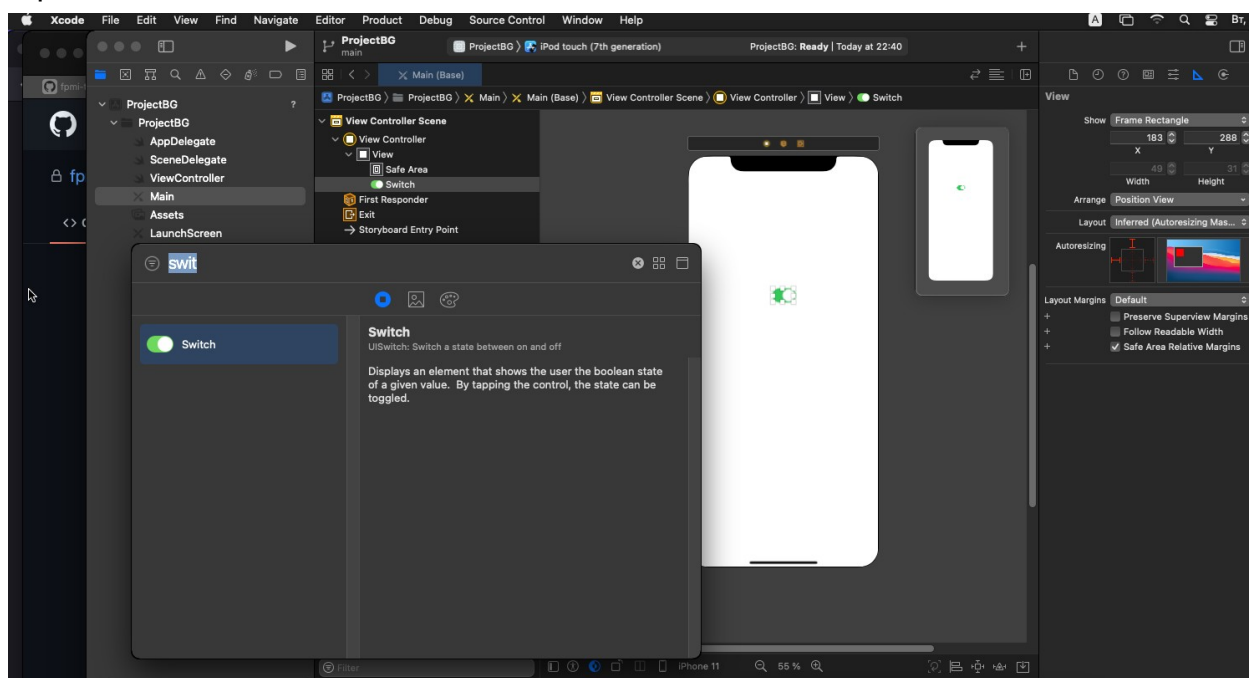


Значение в поле Language было равно **Swift**.

Разработка интерфейса приложения.

Откройте библиотеку объектов, , которая для Xcode 12 доступна на панели инспектора, а для Xcode 13 и выше в виде отдельного приложения (см. рис. Ниже). Находим в библиотеке объектов (Object library) переключатель Switch и перетаскиваем его в окно нашего View контроллера в файле Main.storyboard.

Располагаем его на экране как нам удобно. Также, добавляем объект Label, который будет индикатором переключателя Switch и располагаем его над переключателем.



Теперь необходимо связать объект Switch и Label с кодом нашей программы. Для этого открываем Assistant Editor. В одном окне выбираем файл ViewController.swift, а в другом – Main.storyboard. С зажатой клавишей ctrl перетаскиваем объект Switch в описание класса ViewController. Отпускаем ЛКМ и

задаем имя объекту: например, backgroundSwitch и нажимаем Connect. Должна появиться следующая строка кода (рис. 9):

@IBOutlet weak var backgroundSwitch: UISwitch!

Повторяем эти действия для объекта Label:

@IBOutlet weak var switchIndicator: UILabel!

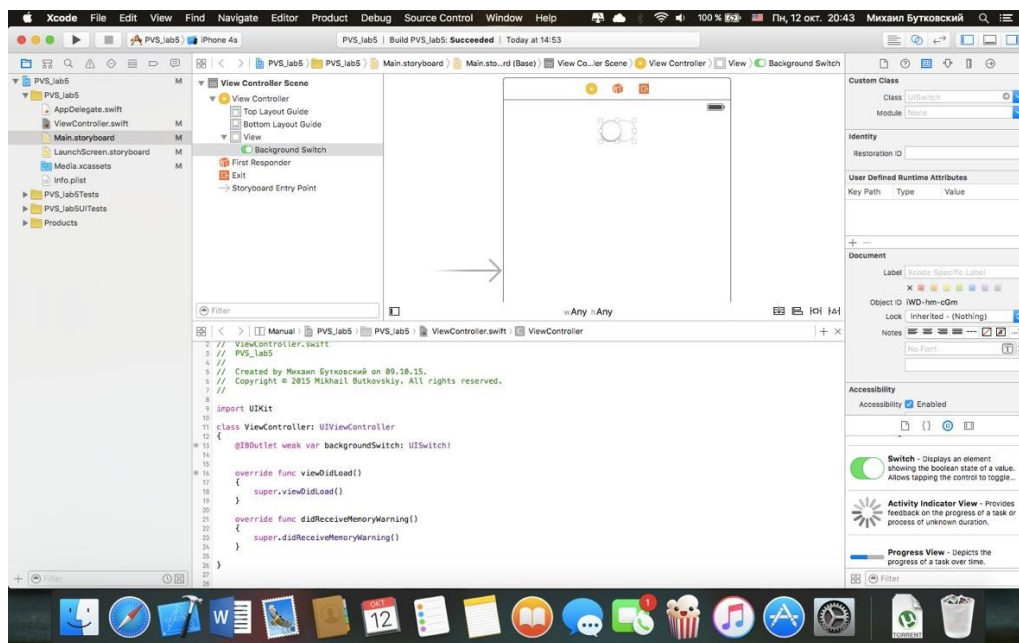


Рис. 9. Результат кода

Добавление изображений в проект Xcode.

Для того, чтобы наше приложение использовало изображения, необходимо добавить их в специальное хранилище в нашем проекте. Для этого заранее изменяем разрешение изображений под разрешение iOS устройства, для которого вы создаете приложение (например – iPhone 5s, 640x1136 пикс.). Далее, в окне навигации открываем файл Assets.xcassets и добавляем набор изображений. Для этого нажимаем “+” (Add a group or image set) внизу списка и нажимаем “New Image Set”. Появляется окно с разными масштабами, под разные устройства (рис.10). Нам необходимо перетащить наше изображение в окошко 2x. Меняем название набора изображений с “Image” на “bg1”. Аналогично создаем набор изображений для “bg2”.

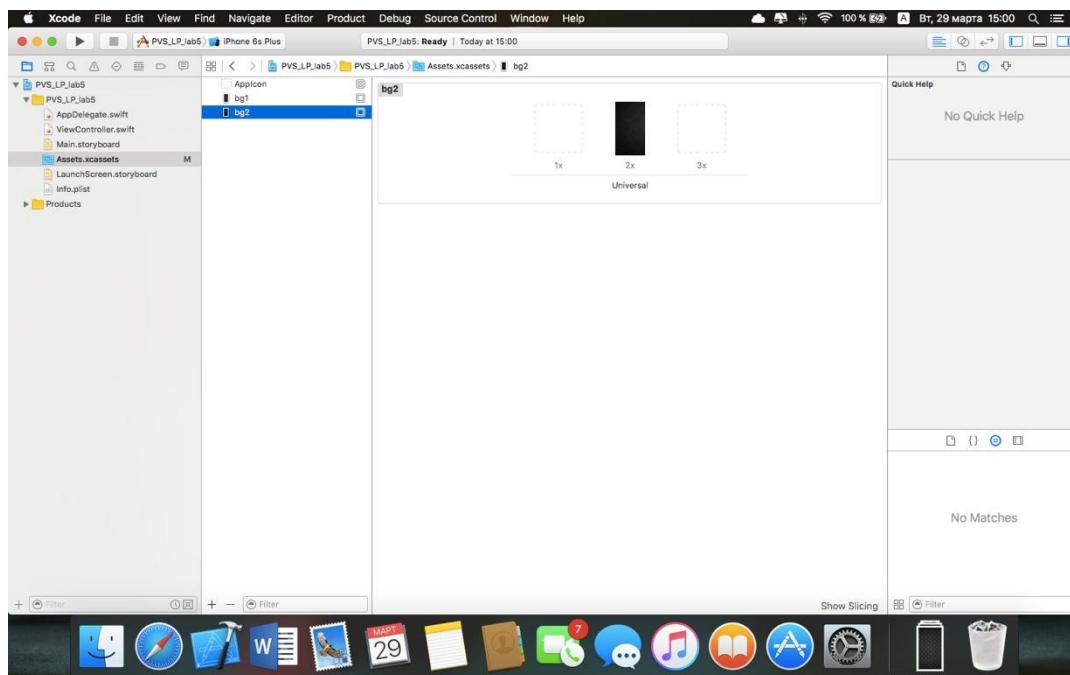


Рис.10. Окно с масштабами под разные устройства

Описание событий объекта “Switch” в коде, сборка и запуск приложения.

Осталось описать события, которые будут происходить при изменении положения выключателя. С зажатой клавишей ctrl добавляем соединение для объекта “Switch” чуть ниже, чем предыдущая связь. Выбираем тип соединения Action и задаем имя, например: “backgroundSwitchTapped” и нажимаем Connect. У нас в коде появилась функция, в которой можно описывать события для объекта “Switch”.

В теле функции опишем следующее условие:

```
@IBAction func backgroundSwitchTapped(sender: AnyObject)
{
    if backgroundSwitch.on
    {
        switchIndicator.text = "Background image: bg1.jpg"
        view.backgroundColor = UIColor(patternImage: UIImage(named: "bg1")!)
    }
    else
    {
        switchIndicator.text = "Background image: bg2.jpg"
        view.backgroundColor = UIColor(patternImage: UIImage(named: "bg2")!)
    }
}
```

Добавляем в функцию ViewDidLoad начальное положение объекта Switch и начальный статус OFF индикатора Label.

```

override func viewDidLoad()
{
    super.viewDidLoad()

    switchIndicator.textColor = UIColor.whiteColor()
    switchIndicator.text = "Background image: bg2.jpg"
    view.backgroundColor = UIColor(patternImage: UIImage(named: "bg2")!)
}

```

Сборка проекта и запуск приложения.

Выполняем команду cmd+R и проверяем работу приложения.

ЗАДАНИЕ 2. ПОКАЗ ПОГОДЫ И ДРУГИХ СВЕДЕНИЙ О ГОРОДЕ НА ЯЗЫКЕ OBJECTIVE-C

УПРАЖНЕНИЕ 2.1

Необходимо реализовать программу показа погоды для 6 городов. Цвет показателей температуры должен соответствовать погоде. Для хранения температуры и города использовать массив и/или словарь.

Входные данные

Строка, содержащая название города.

Выходные данные

Строка, содержащая информацию о температуре в городе и имеющая атрибут цвет, зависящий от температуры.

УПРАЖНЕНИЕ 2.2.

Внести изменения в спроектированное приложение согласно варианту.

Исходный код приложений должен быть оформлен согласно руководству стиля для языка Objective-C и содержать комментарии к основным блокам кода:

a) <https://github.com/DigDes/objective-c-style-guide>

b) <https://github.com/raywenderlich/objective-c-style-guide>

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Создайте новый проект вида Application → Single View Application (App в случае Xcode 13 и выше)

(Рис. 11). Затем введите название проекта, имя разработчика, название организации (Рис. 12). Устройством выберите iPhone. Отметьте Use Storyboards.

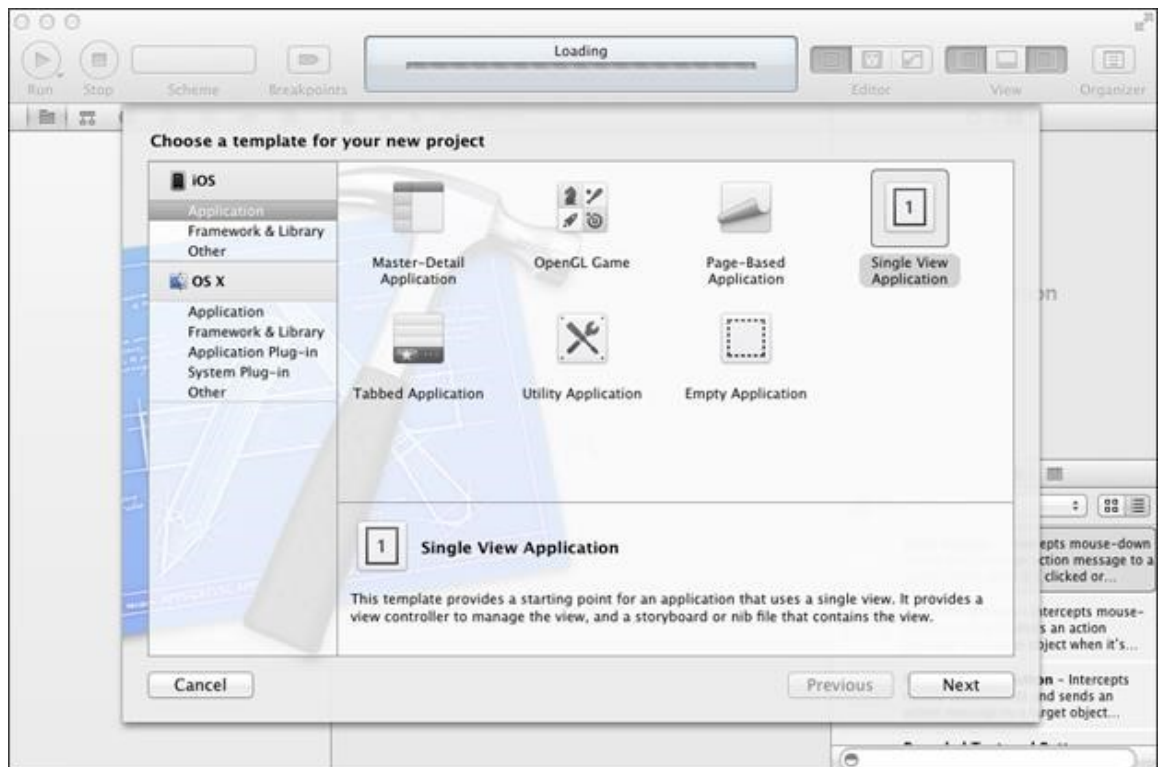


Рис. 11. Окно выбора типа создаваемого проекта

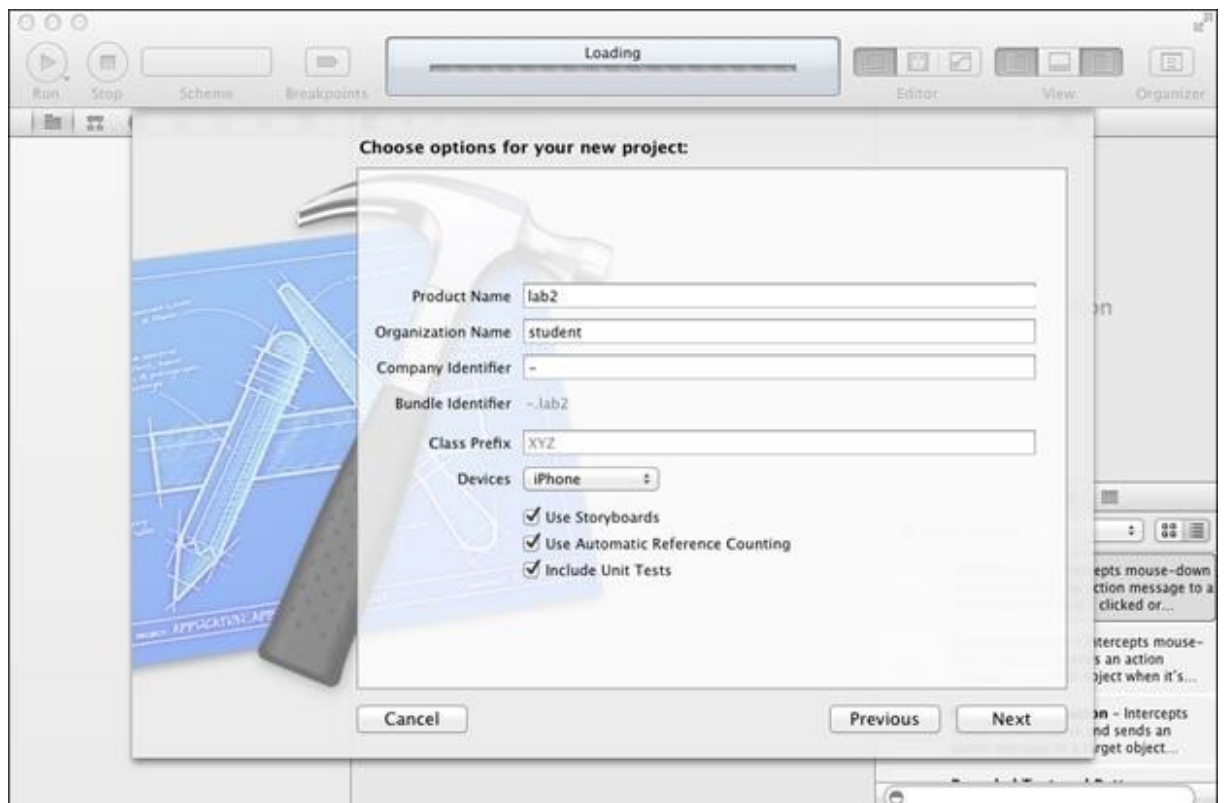


Рис. 12. Окно настройки нового проекта

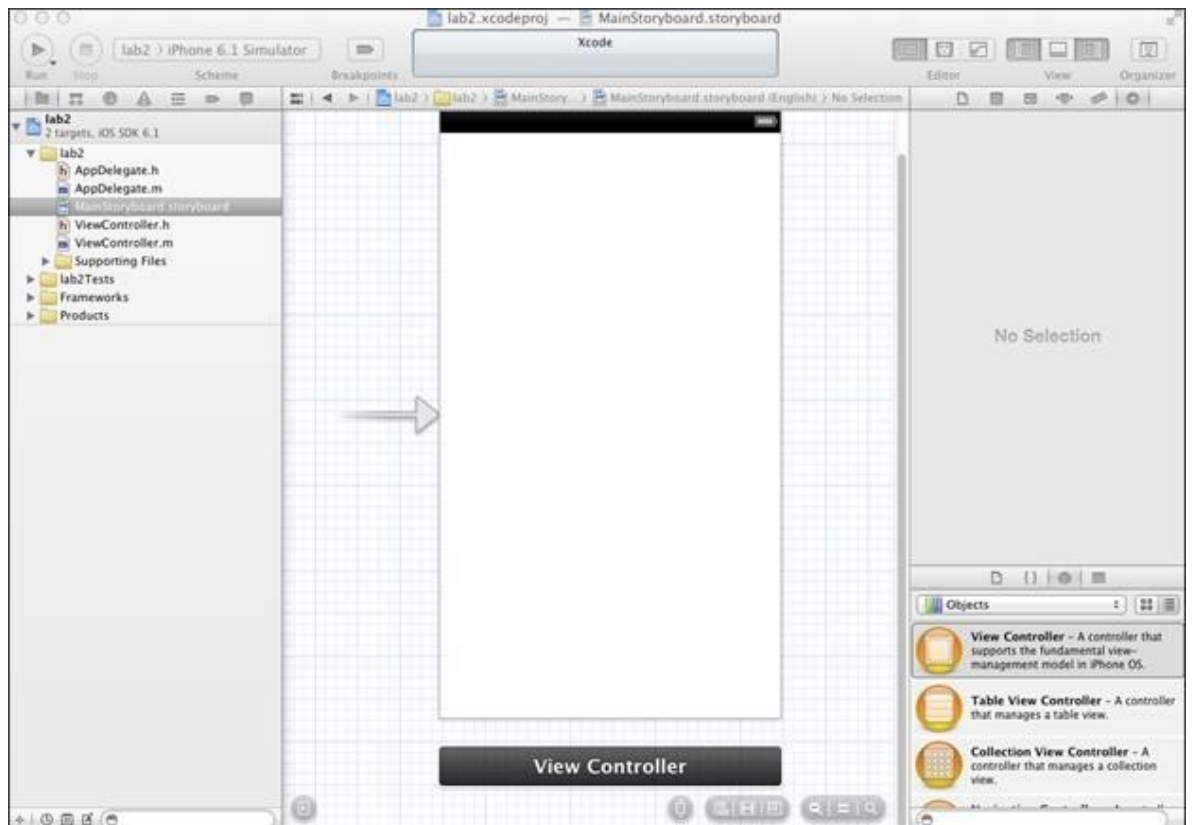


Рис. 13. Созданный проект

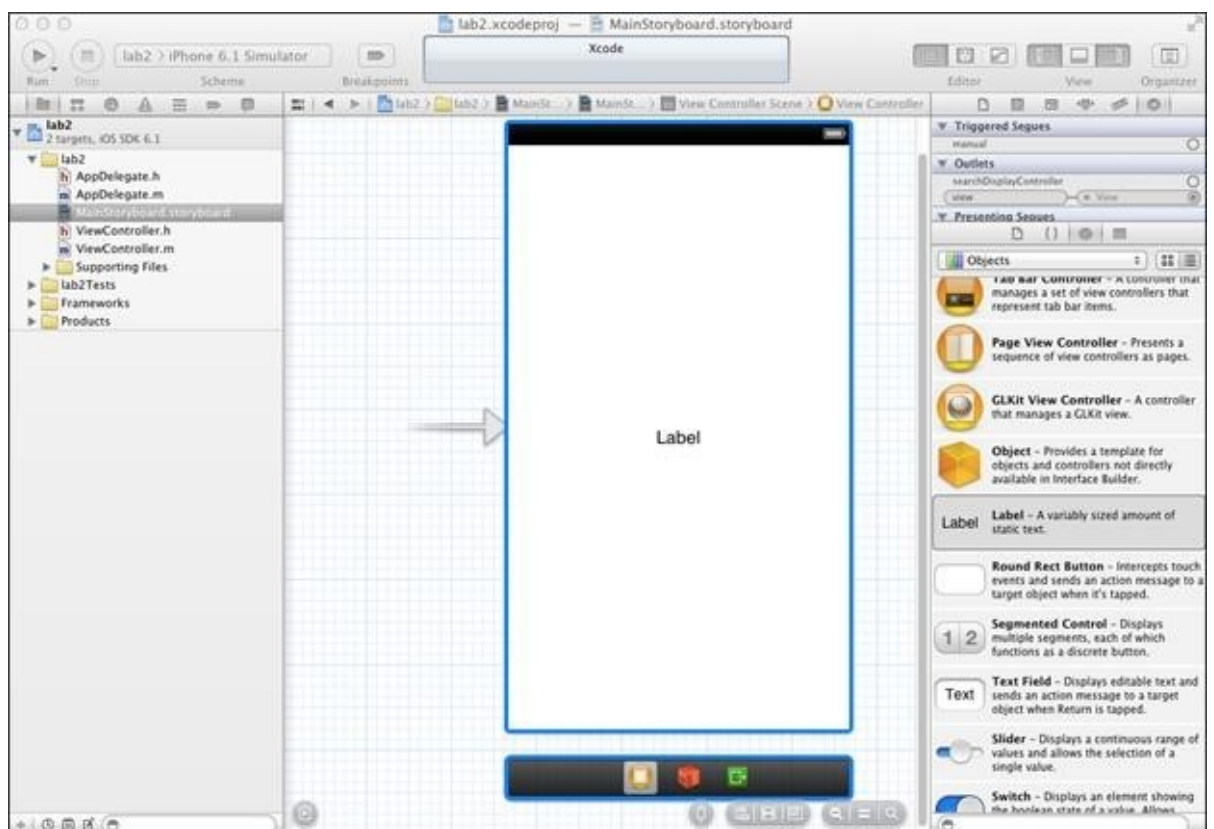


Рис. 14. Добавление надписи

Выберите папку и сохраните туда проект. В области навигации выберите MainStoryboard.storyboard. Окно редактора будет выглядеть примерно как на (Рис. 13).

На панели компонентов найдите компонент Label. Перетащите его на изображение пустого окна приложения iPhone (Рис. 14). Таким же образом добавьте кнопку (Round Rect Button).

Чтобы иметь возможность менять текст надписи в контроллере, необходимо добавить в него свойство надпись и привязать элементу интерфейса надписи (создать IBOutlet). Для этого выберите Assistant Editor верхней панели инструментов с названием Editor (иконка с изображением рубашки с бабочкой) (Рис. 15).

Ключевое слово IBOutlet (Interface Builder Outlets) предназначено для связи элемента интерфейса на форме созданным дизайнером форм и кодом.

Удерживая Ctrl перетащите надпись в файл ViewController.m между @interface и @end.

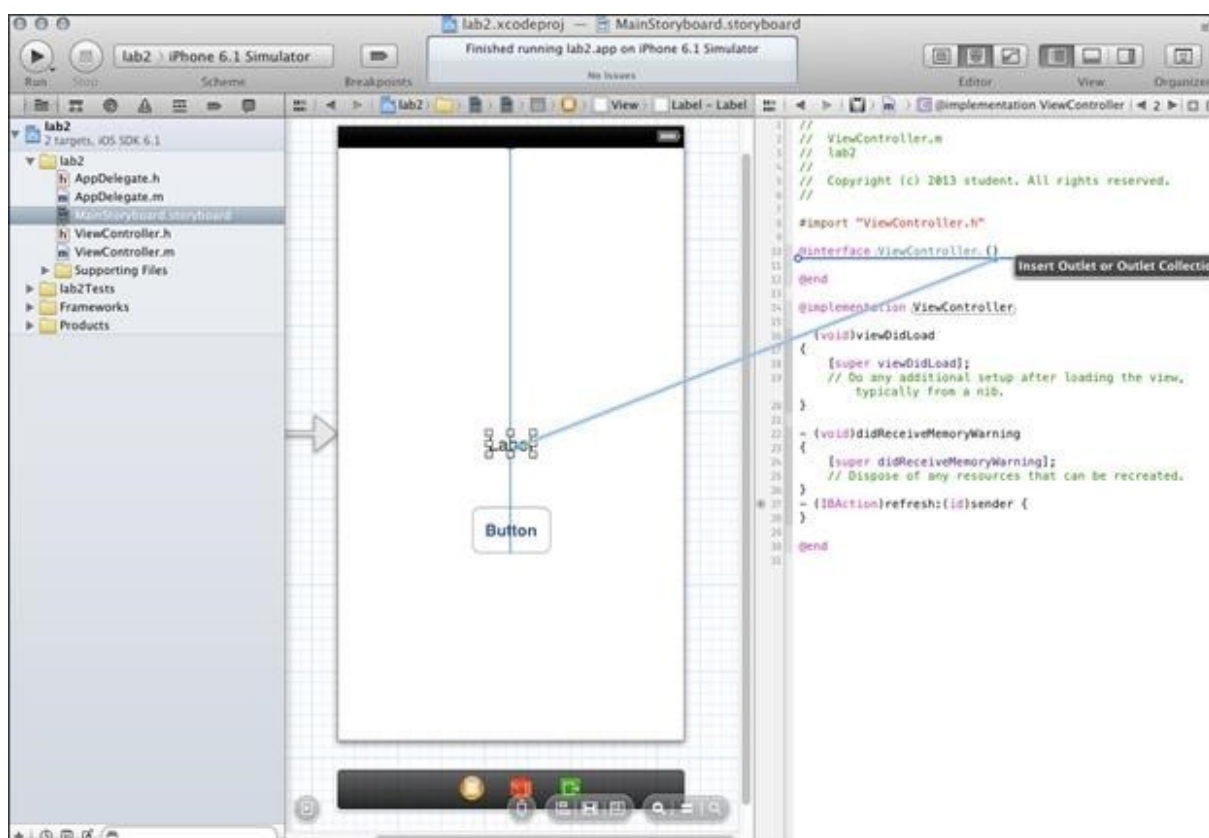


Рис. 15. Режим ассистента

Введите название свойства (Рис. 16).

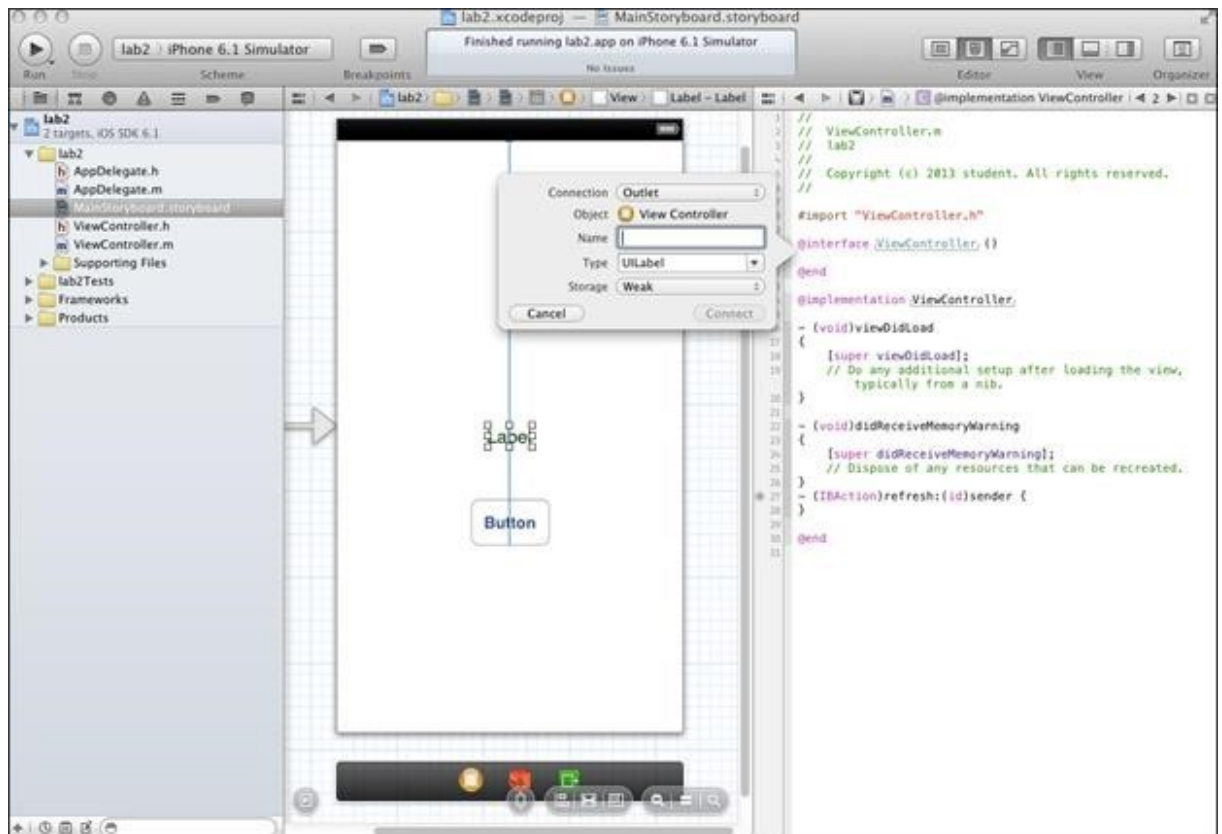


Рис. 16. Создание IBOutlet

В результате в часть интерфейса контроллера добавится объявление IBOutlet (Листинг 1) .

```
@property (weak, nonatomic) IBOutlet UILabel *indicator;
```

Листинг 1. Объявление IBOutlet

Чтобы привязать к кнопке действие необходимо также с помощью Ctrl перетащить кнопку в область реализации между @implementation @end

(Создать IBAction) (Рис. 17). Введите название метода и выберите его тип. (Рис. 18)

В результате, в часть реализации контроллера добавится код обработчика (Листинг 2):

```
- (IBAction)refresh:(id)sender {  
}  
}
```

Листинг 2. Определение обработчика

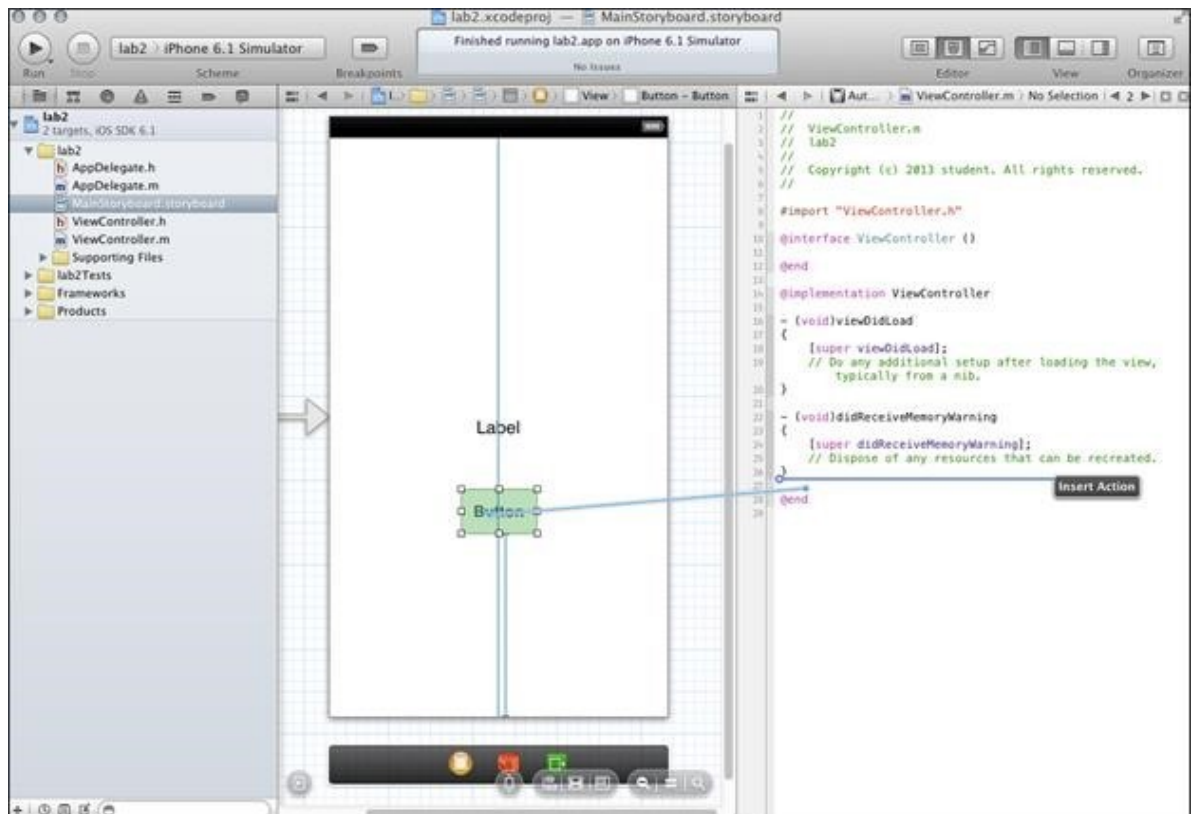


Рис. 17. Связывание объекта кнопки с обработчиком

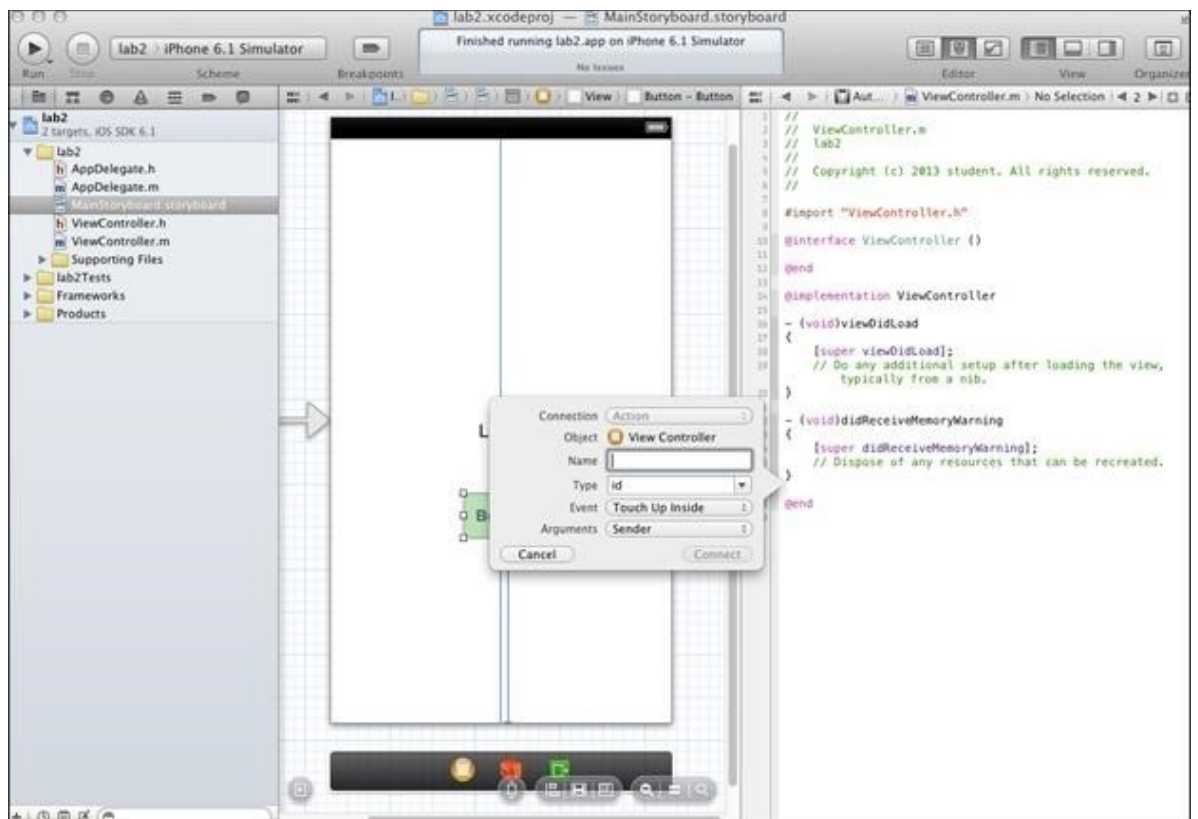


Рис. 18. Создание обработчика

Добавьте в метод refresh код установки значения надписи (Листинг 2).

```
[[self indicator] setText:@"0 C"];
```

Листинг 3. Установка значения надписи

Запустите программу.

Теперь при нажатии на кнопку надпись меняется на 0 C (Рис. 19).

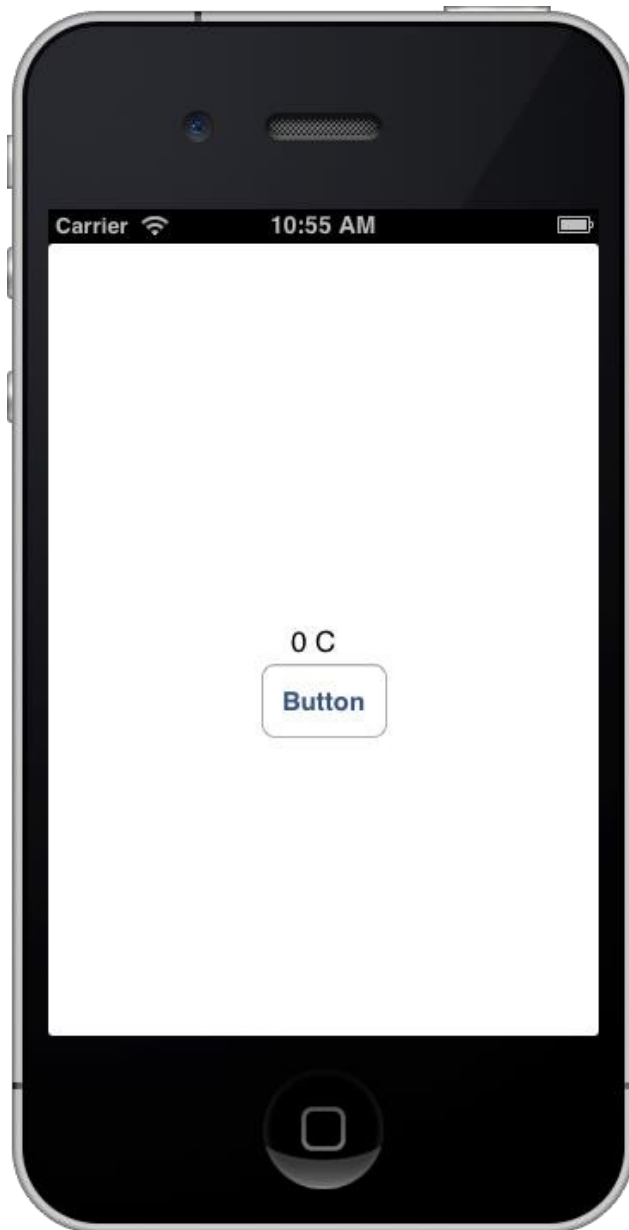


Рис. 19. Результат работы программы

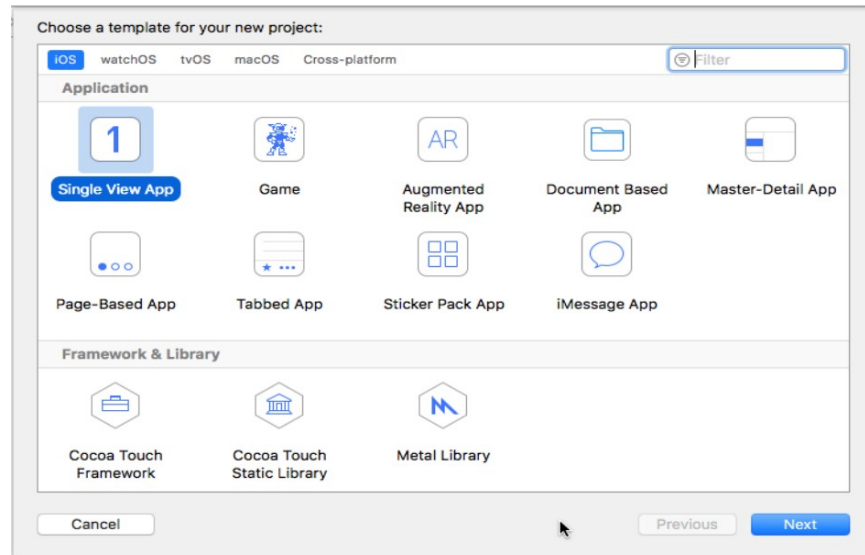
Теперь необходимо добавить функционал получения текущей температуры. (Листинг 3).

ЗАДАНИЕ 3. ПРИЛОЖЕНИЕ ДЛЯ РАСЧЕТА КОЛИЧЕСТВА КАЛОРИЙ И ИНДЕКСА МАССЫ ТЕЛА НА ЯЗЫКЕ SWIFT

В данном задании реализовать простое приложение для расчета индекса массы тела и необходимого количества калорий для поддержания веса. Для расчет индекса массы тела использовать формулы [Гарисса Бенедикта](#) и [индекса массы тела](#).

В руководстве приведён пример исходного кода для Swift 3. Адаптировать для Swift 5 и, при необходимости, исправить ошибки, связанные с версией языка.

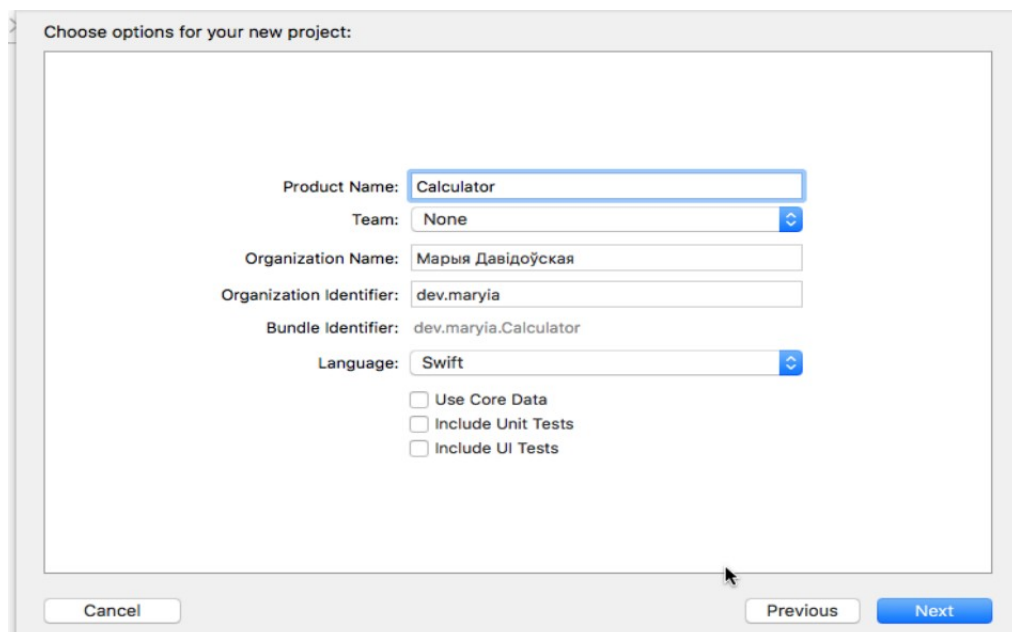
1. Для начала давайте создадим проект: запустите Xcode и нажмите комбинацию клавиш CMD+Shift+N или выберите в меню *File* → *New* → *Project*



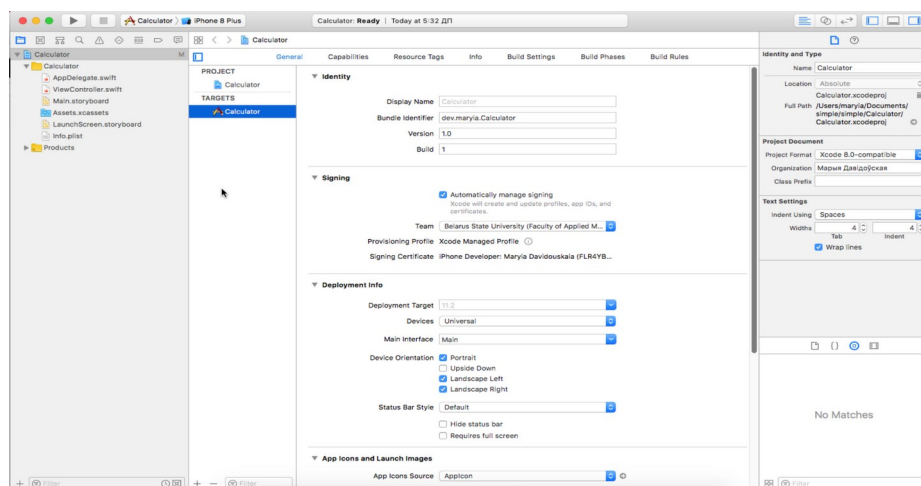
2. Теперь выберите *Single View Application (App)* и нажмите *Next*.

3. На этом этапе надо придумать имя для приложения (Product Name), нажать *Next* и выбрать папку для сохранения.

Главное, чтобы значение в поле *Language* было равно **Swift**.



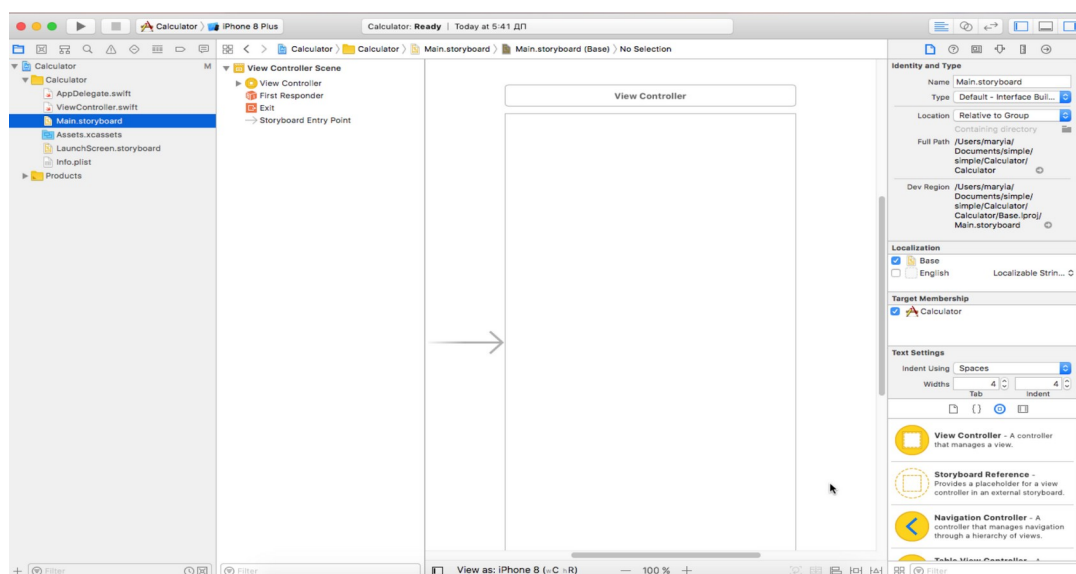
4. После того, как вы сохраните проект, перед вами появится рабочее пространство Xcode.



Слева у вас будет панель Navigator, а справа — Utilities. В каждой из этих панелей есть свои вкладки, которые позволяют получить доступ к различным функциям.

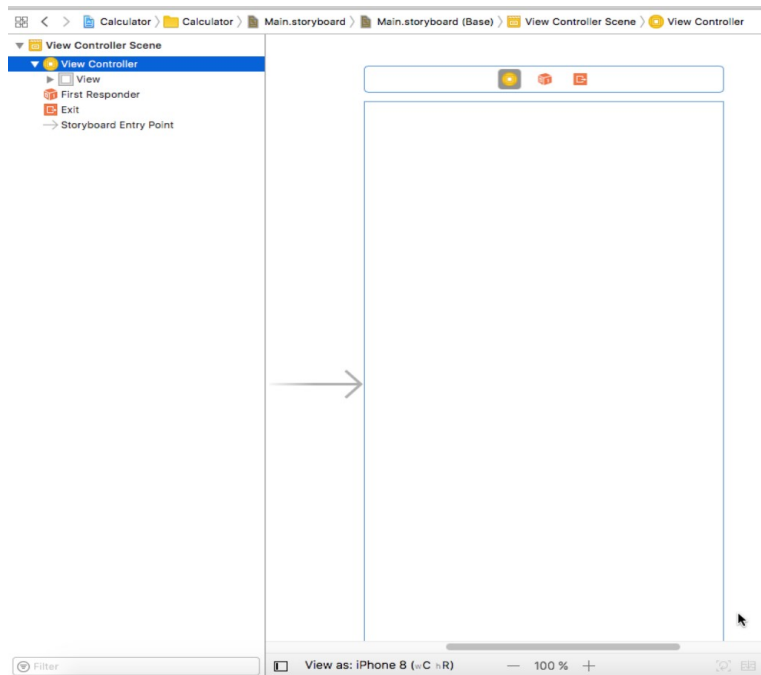
Например, первая вкладка в панели Navigator показывает список файлов проекта, третья позволяет производить поиск по проекту и так далее.

5. Найдите в списке файлов проектов файл с именем *Main.storyboard* и нажмите на него.

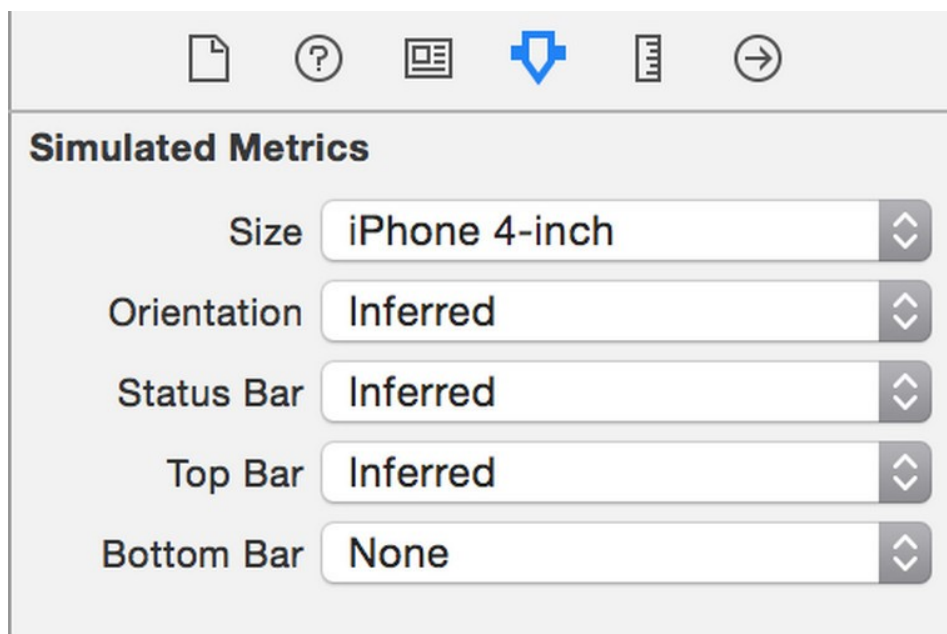


Файл *Main.storyboard* определяет, какие экраны (контроллеры) есть в приложении. Вы можете добавлять элементы на экраны, задавать связи между экранами и так далее.

Если вы выберете какой-нибудь контроллер, он подсветится синей рамкой:



После того, как контроллер выбран, его свойства начнут отображаться в панели *Utilities*. Например, контроллеру можно поменять размер, выбрав другое значение в поле *Size*.



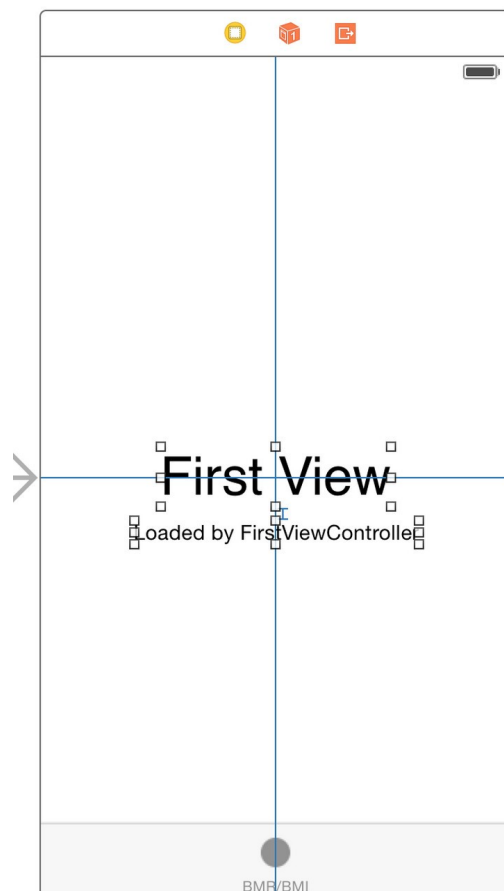
6. Запустим наше шаблонное приложение и посмотрим, как оно выглядит в симуляторе. Для этого надо в верхнем левом углу выбрать тип симулируемого устройства и нажать комбинацию клавиш **CMD+R** или кнопку с иконкой *Play*.





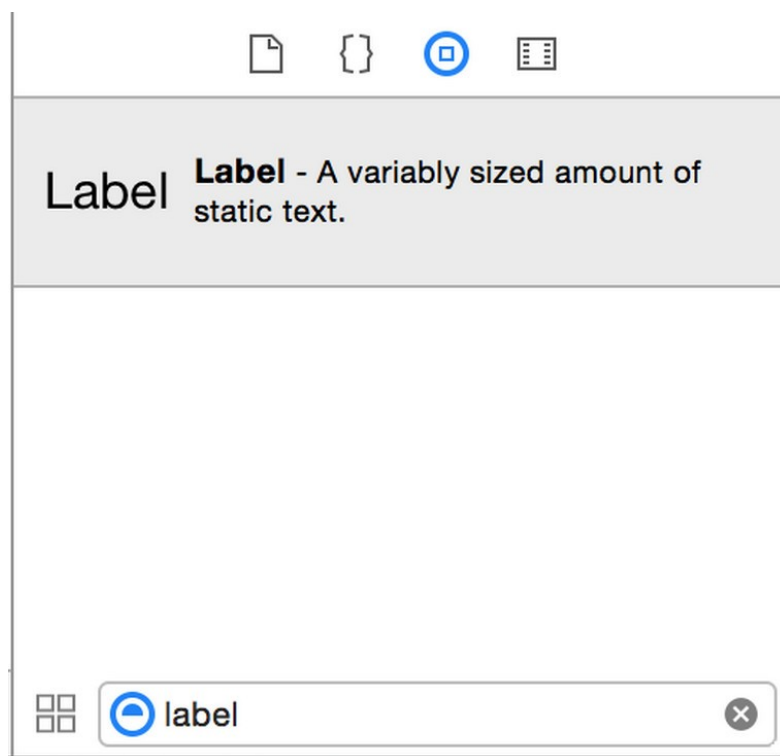
Как видите, у нас есть приложение с двумя экранами, переключение между которым осуществляется при помощи панели вкладок.

[OBJ.]

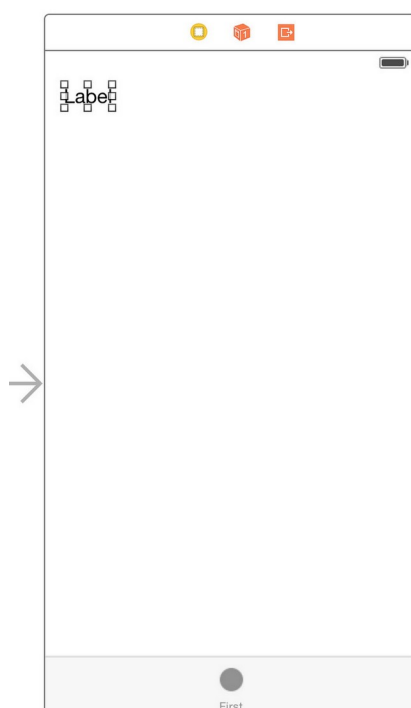


9. Теперь добавим свои элементы управления. Внизу панели Utilities есть *Object Library*, в которой можно выбрать объекты и добавить их к сцене. Найдите

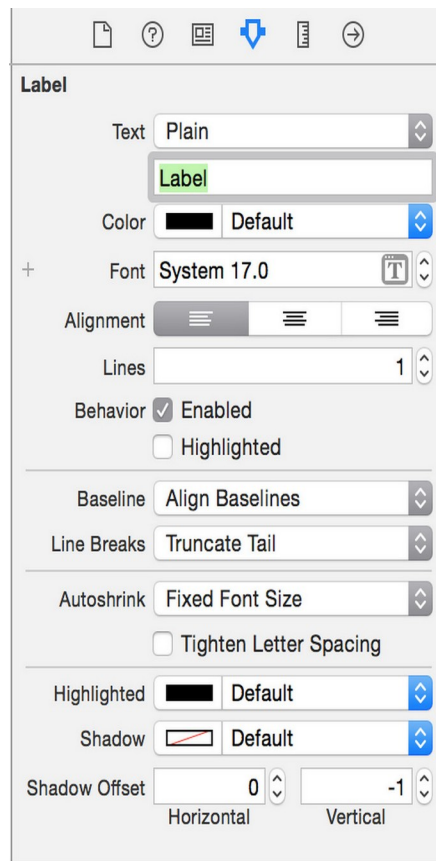
объект *Label* и перетащите его на контроллер BMR/BMI. Перед этим не забудьте два раза тапнуть на сцене, чтобы масштаб стал стандартным.



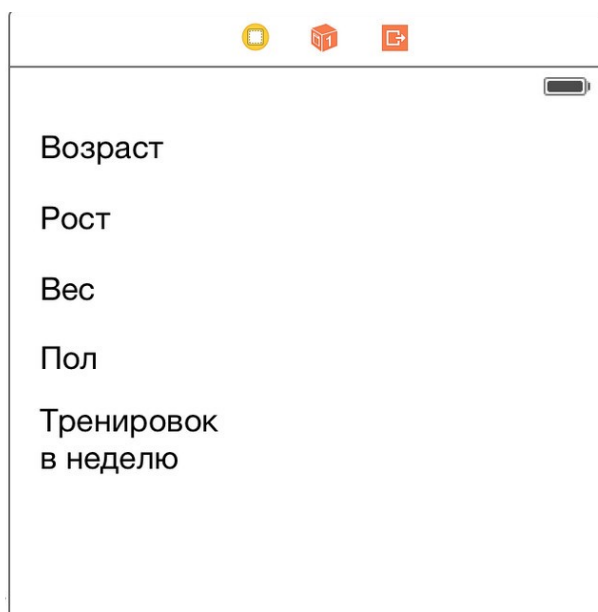
Должно получиться примерно так.



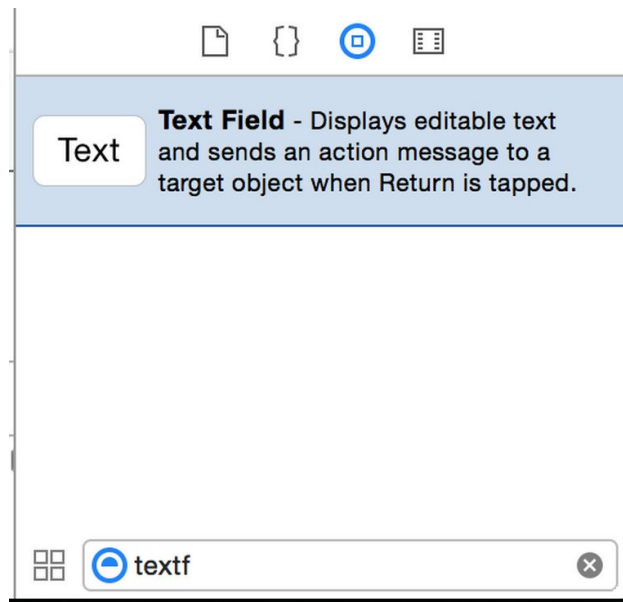
В панели Utilities можно поменять текст, размер шрифта и так далее.



10. Перетащите еще несколько лейблов и поставьте им такой же текст, как и на картинке. Для последнего лейбла нужно поставить *Lines* в 2, чтобы текст переносился на другую строку.

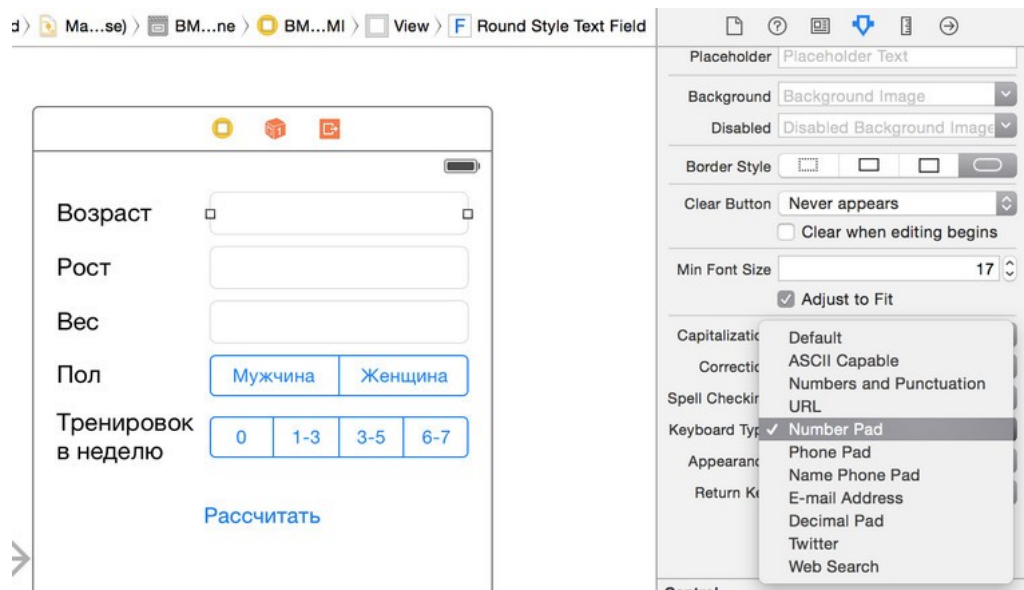


11. Теперь добавьте 3 текстовых поля (*Text Field*).

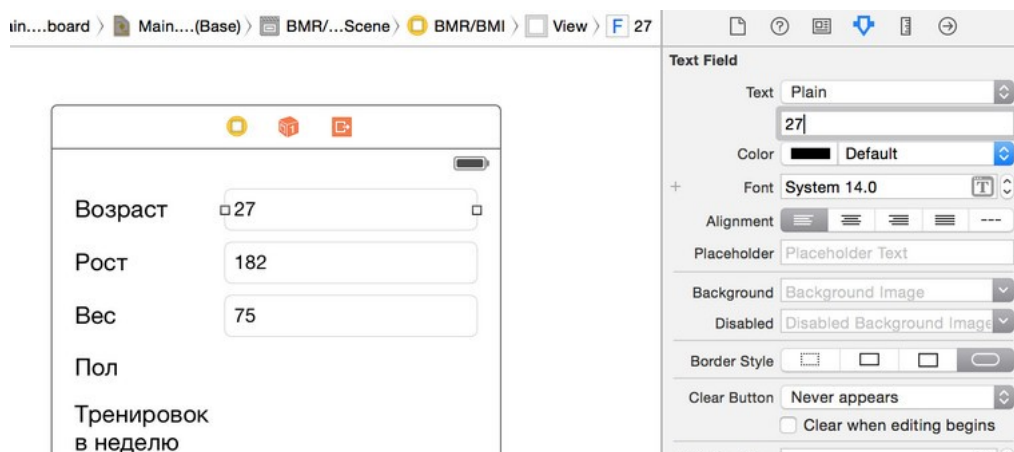


И разместите их, как на картинке.

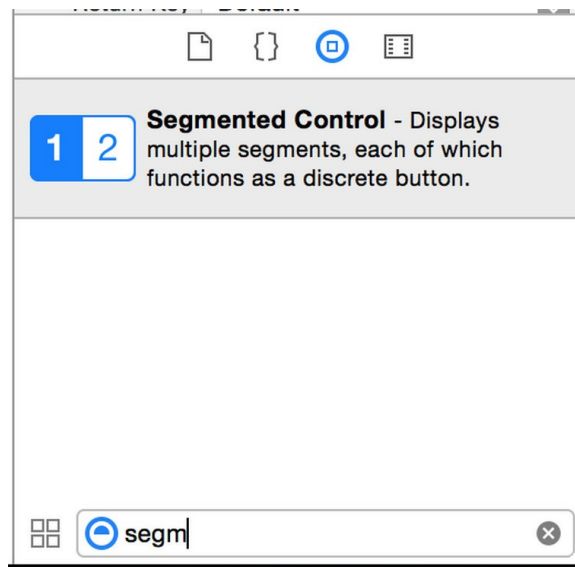
У текстовых полей поставьте параметр *keyboard type* в *Number Pad*.



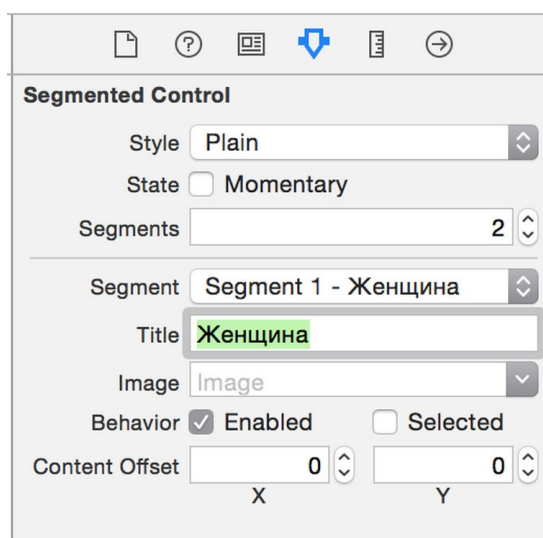
И установите текст по умолчанию.



12. Теперь добавим элементы управления для выбора пола и кол-ва тренировок в неделю. В Object Library ищите *Segmented Control* и добавляйте на экран.



У Segmented Control можно менять количество сегментов и текст для каждого сегмента.



Сделайте так, чтобы сегменты выглядели, как на скриншоте.

Возраст

Рост

Вес

Пол Мужчина Женщина

Тренировок в неделю 0 1-3 3-5 6-7

13. Теперь добавьте кнопку (*button*).

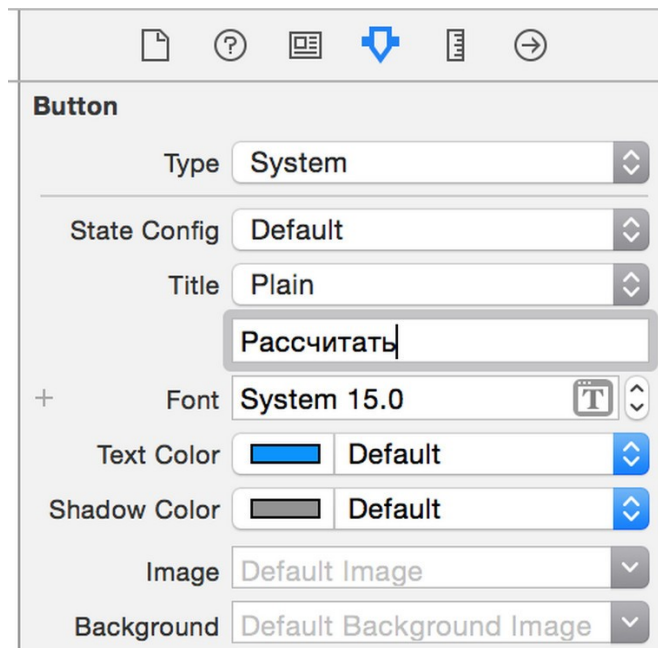
Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Bar Button Item - Represents an item on a UIToolbar or UINavigationController object.

Fixed Space Bar Button Item - Represents a fixed space item on a UIToolbar object.

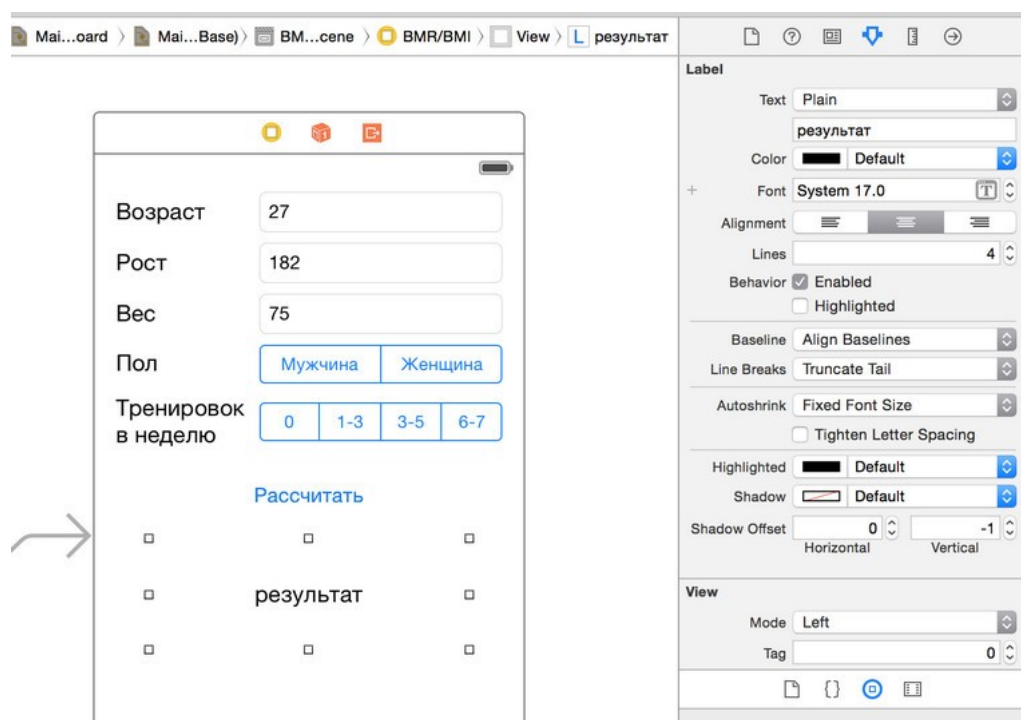
button

И установите ей заголовок.



[OBJ]

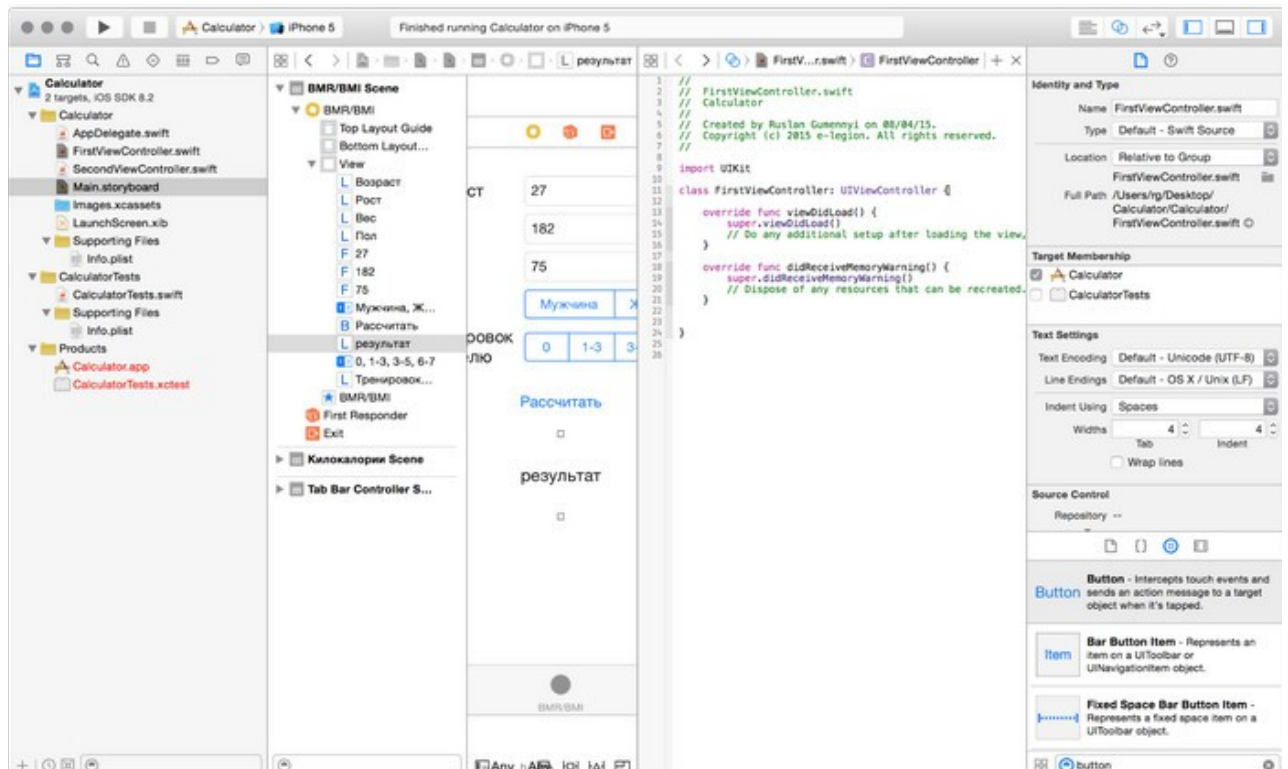
После этого добавьте еще один label с *lines* равным 4.



14. Выберите контроллер BMI/BMR, нажмите два пересекающихся кольца в правом верхнем углу и откроется *Assistant Editor*. Он показывает код, ассоциированный с этим контроллером.

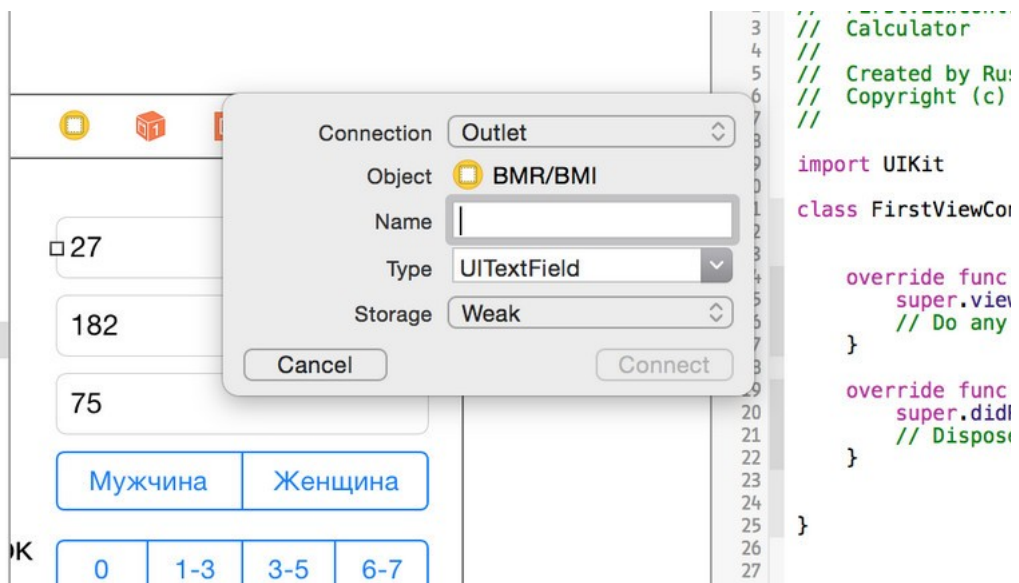


15. Протащим элементы управления.



Для этого выберите первый *text field* (в нашем случае это будет возраст), нажмите Ctrl, кликните на него еще раз и, не отпуская кнопки, перетащите внутрь класса. Если все сделано правильно, то вы увидите подсказку “*Insert Outlet, Action, ...*”.

Теперь отпустите курсор, и вы увидите окошко для создания связи. В качестве имени введите *ageTextField* и нажмите *Connect*.



16. У вас появится переменная *ageTextField*.

```

7 //
8
9 import UIKit
10
11 class FirstViewController: UIViewController {
12
13     @IBOutlet weak var ageTextField: UITextField!
14
15     override func viewDidLoad() {
16         super.viewDidLoad()
17         // Do any additional setup after loading the view
18     }
19
20     override func didReceiveMemoryWarning() {
21         super.didReceiveMemoryWarning()
22         // Dispose of any resources that can be recreated
23     }
24
25 }
26
27
28

```

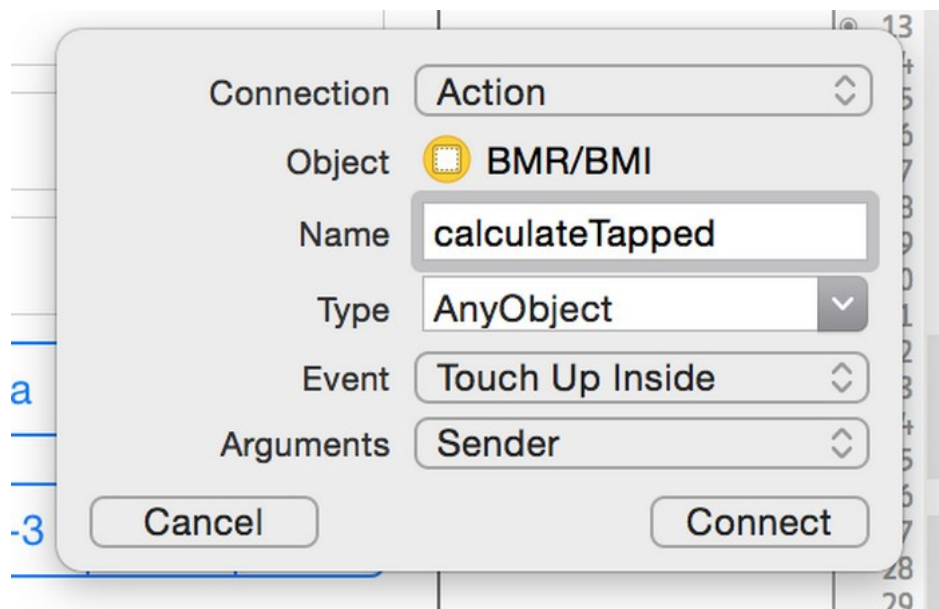
Проделайте эту процедуру для оставшихся text field, segmented controls и label с текстовым результатом. У вас должно получиться так:

```

11 class FirstViewController: UIViewController {
12
13     @IBOutlet weak var ageTextField: UITextField!
14     @IBOutlet weak var heightTextField: UITextField!
15     @IBOutlet weak var weightTextField: UITextField!
16     @IBOutlet weak var sexSegmentedControl: UISegmentedControl!
17     @IBOutlet weak var activitySegmentedControl: UISegmentedControl!
18     @IBOutlet weak var resultsLabel: UILabel!
19
20 }

```

Теперь протащите кнопку, но тип соединения укажите не Outlet, а *Action*. И в качестве имени используйте *calculateTapped*.



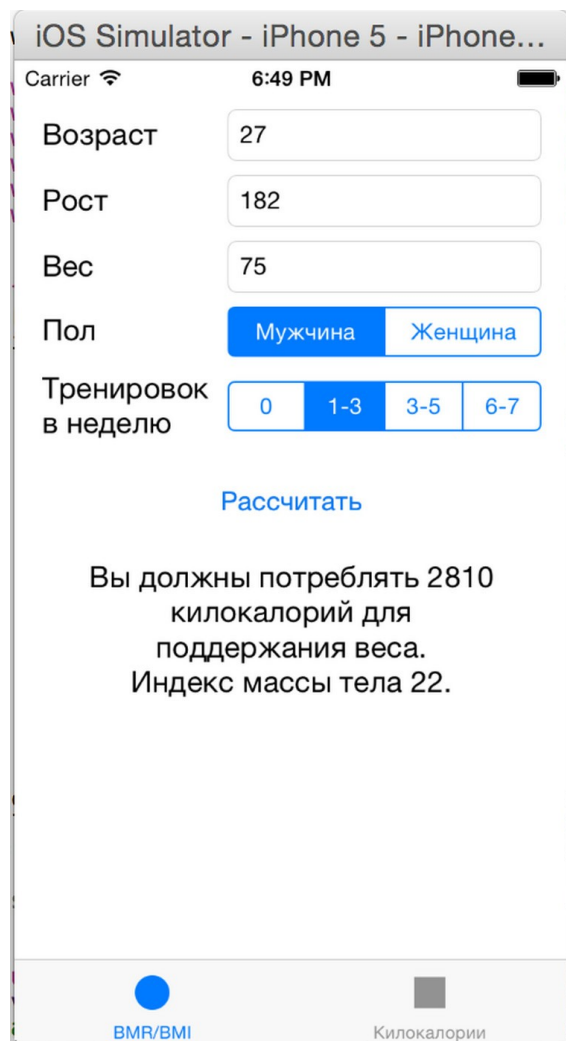

```

11 class FirstViewController: UIViewController {
12
13     @IBOutlet weak var ageTextField: UITextField!
14     @IBOutlet weak var heightTextField: UITextField!
15     @IBOutlet weak var weightTextField: UITextField!
16     @IBOutlet weak var sexSegmentedControl: UISegmentedControl!
17     @IBOutlet weak var activitySegmentedControl: UISegmentedControl!
18     @IBOutlet weak var resultsLabel: UILabel!
19
20
21     @IBAction func calculateTapped(sender: AnyObject) {
22     }
23
24     override func viewDidLoad() {

```

17. Добавьте реализацию расчёта показателя BMR и индекса массы тела в метод *calculateTapped*.

18. Проверьте результат.



ЗАДАНИЕ 4. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ С ИСПОЛЬЗОВАНИЕМ SWIFTUI НА ЯЗЫКЕ SWIFT

Разобрать и реализовать пример из статьи «SwiftUI: Getting Started» <https://www.raywenderlich.com/3715234-swiftui-getting-started>.

ЗАДАНИЕ 5. ПРОСТОЕ МОБИЛЬНОЕ ПРИЛОЖЕНИЕ НА ЯЗЫКЕ SWIFT

Используя изученные типы полей в задании 3, реализовать простое приложение на языке swift с использованием UIKit или SwiftUI (на выбор) согласно варианту из 2-х сцен. Адаптировать приложение для книжной и альбомной ориентации и под разные размеры экранов.

Продemonстрировать использование типов полей text field, segmented controls и label

При создании приложения с помощью UIKit и InterfaceBuilder продемонстрировать позиционирование элементов с помощью AutoLayout и StackView, которые обеспечивают адаптацию приложения под разные экраны и ориентацию. Пример приложения с использованием AutoLayout, StackView и нескольких сцен представлен в статье <https://www.twilio.com/blog/2018/05/xcode-auto-layout-swift-ios.html>. **Рекомендуется для изучения!!!**

При написании кода следовать рекомендациям стиля для языка Swift и добавить комментарии :

- a) <http://ilya2606.ru/?p=1846>
- b) <https://github.com/RedMadRobot/RMR-swift-style-guide>
- c) <https://github.com/raywenderlich/swift-style-guide>

КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Как зарегистрироваться как разработчик apple и добавить идентификатор организации, под которым будут создаваться приложения в профиль разработчика и xcode?
- 2) Как добавить элементы управления на экран (контроллер)?
- 3) Какие типы элементов управления есть в библиотеке объектов Xcode?
- 4) Что такое IBOutlet?
- 5) Что такое IBAction?
- 6) Опишите способы создания IBOutlet и IBAction?
- 7) Из каких компонент состоит графический интерфейс программ на Objective-C?
- 8) Что такое Outline View и Canvas в контексте InterfaceBuilder?
- 9) Что такое AutoLayout и Safe area?
- 10) Что такое scene, container, relationship connection, segue?

- 11) Что такое StackView и constraints?
- 12) В чем отличие между UIKit и SwiftUI?

Варианты

Вариант 1

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение герба города, и страну, в которой город находится. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Литва.

5. Напишите приложение, которое поможет слушателям начальной школы изучить умножение и деление для однозначных и двухзначных чисел. Тип числа и тип операции задается. Значения чисел вводятся.

Вариант 2

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение главы города, и количество населения. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Россия.

5. Напишите приложение, которое поможет слушателям начальной школы изучить сложение и вычитание двузначных и трехзначных чисел. Тип числа и тип операции задается. Значения чисел вводятся.

Вариант 3

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного памятника города, и страну, в которой город находится. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Италия.

5. Напишите приложение, которое принимает сведения об оценках по предметам Математический анализ, Геометрия и алгебра, Программирование в 10-бальной системе и переводит оценки и средний балл в выбранную систему оценивания. Перевести оценки в 5-балльную, 12-балльную и 25-балльную системы.

Вариант 4

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение университета города, и количество студентов в нем. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Напишите приложение, которое на основе выбора типа геометрической фигуры треугольник, квадрата, прямоугольник и трапеция принимает на вход параметры сторон и рассчитывает площадь фигуры.

Вариант 5

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного музея, который находится в городе, и название музея. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Армения.

5. Напишите приложение, которое выбрать тип маятника (маятник или пружинный маятник), ввести нужные параметры (массу или длину) и вычислить период колебания.

Для пружинного маятника период колебания рассчитывается по формуле:

$$T = 2 \cdot \pi \sqrt{\frac{m}{k}}, \text{ где } m \text{ – масса, } k = 100 \text{ Н/м – коэффициент упругости.}$$

Для маятника длиной L период колебания вычисляется по формуле:

$$t = 2 \cdot \pi \sqrt{\frac{l}{g}}, \text{ где } g \text{ – ускорение свободного падения (9.81 м/с}^2\text{).}$$

Вариант 6

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного ресторана, который находится в городе, и вид кухни. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Напишите приложение, которое принимает на вход дату и время в часах и минутах и в зависимости от выбора из вариантов Количество дней, Количество минут, Количество секунд вывод количество дней с начала года, количество минут с начала дня и количество секунд с начала часа.

Вариант 7

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение библиотеки города, и адрес библиотеки. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Написать приложение, которое принимает на вход значение высоты H радиуса основания R и в зависимости от типа фигуры цилиндр или конус, вычисляет объем выбранной фигуры.

Вариант 8

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение спортивного объекта города, и виды спорта, которые проводятся в этом объекте. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Польша.

5. Написать приложение для суммы выплаты по депозиту на вклад в различных валютах: до 5000 руб начисляется 20% годовых, до 1000 руб — 5% годовых, до 5000 долларов – 12% годовых, до 1000 долларов – 3% годовых.

Вариант 9

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение лучшего парка города, и название парка и его характеристики (наличие памятников, дата основания и т. д.). В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Литва.

5. Написать приложение для определения стоимости переговоров, если стоимость 1 минуты на территории Беларуси равна 0.15 руб, с Россией – 0.90 руб, а в субботу и воскресенье дополнительно предоставляется скидка 10%. Ввести данные – dt (продолжительность разговора в минутах), s (стоимость минуты разговора). День недели d (день недели от 1 до 7) выбирать в переключателе segmented controls.

Вариант 10

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение госпиталя (больницы) города, и адрес расположения больницы. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Украина.

5. Написать приложение для расчета надбавки к зарплате за стаж, если стаж от 2 до 5 лет, надбавка составляет 2%, если стаж от 5 до 10 лет — 5%. Ввести с клавиатуры зарплату, стаж выбрать в поле segmented control, вывести надбавку и сумму к выплате.

Вариант 11

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного спортсмена, проживающего в городе, и его основные достижения. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Россия.

5. Написать приложение для расчета суммы ежемесячных выплат процентов по депозиту, если договор составлен на полгода под 12% годовых в руб, 3% в долларах, 1.5% в евро или на год под 18% годовых в руб., 6% в долларах и 3% в евро С клавиатуры ввести сумму вклада и срок договора, выбрать тип валюты.

Вариант 12

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение писателя, проживающего в городе, и название одного из его произведений. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Написать приложение для определения плотности материала в зависимости от его типа. Ввести объем и массу тела, тип материала выбрать из перечня. Использовать 5 видов материалов.

Вариант 13

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение герба города, флага, и страну, в которой город находится. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Бразилия.

4. Разработать приложение: добавьте кнопку результата, текстовые поля (ввод a, c), Segmented Control (выбор целых или дробных коэффициентов). При щелчке по кнопке – вычисление функции $y=ax^2+c$. В текстовых полях: входные данные. Вывод информации на разных фонах.

Вариант 14

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного объекта (здания) города, и количество населения. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

4. Разработать приложение делающее преобразования: меры длин Американских в русские (в разных направлениях. (Мер длин не меньше 6). Вывод информации на разных фонах. Вывод информации на разных фонах.

Вариант 15

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного памятника города, и страну, в которой город находится. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Сирия.

5. Разработать приложение делающее преобразования мер старорусских и старобелорусских длин (в разных направлениях). Вывод информации на разных фонах.

Вариант 16

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение университета города, и количество факультетов и студентов в нем. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь и университет — БГУ.

4. Разработать приложение «калькулятор»: преобразование из одной системы счисления в другую ($2s/c$, $8 s/c$, $16s/c$). В разных направлениях. Вывод информации на разных фонах.

Вариант 17

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного музея, который находится в городе, и название музея. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Грузия.

5. Написать приложение для расчета скидки за покупку в зависимости от количества товаров в покупке или стоимости покупки. Предоставлять следующие виды скидок: по количеству (1% - 10 товаров, 3% - 20 товаров) и стоимости (3% - > 50 руб, но <100 руб, 5% - > 100руб). Поле segmented control использовать для выбора типа скидки.

Вариант 18

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного ресторана, который находится в городе, и вид кухни. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

4.Разработать приложение «калькулятор»: вычисление тригонометрических функций (арккосинус, арксинус, арктангенс, аркотангенс). Вывод информации на разных фонах.

Вариант 19

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение библиотеки города, и адрес библиотеки. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Великобритания.

5. Разработать приложение «калькулятор»: вычисление гиперболических функций. Вывод информации на разных фонах.

Вариант 20

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение спортивного объекта города, и виды спорта, которые проводятся в этом объекте. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Россия.

4.Разработать приложение делающее преобразования: меры морских длин в русские (в разных направлениях. (Мер длин не меньше 6) Вывод информации на разных фонах.

Вариант 21

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение лучшего парка города, и название парка и его характеристики (наличие памятников, дата основания и т. д.). В приложении

должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Латвия.

4. Написать приложение делающее преобразования 10-й системы счисления в римскую запись числе в двух направлениях. Вывод информации на разных фонах.

Вариант 22

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение госпиталя (больницы) города, и адрес расположения больницы. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Разработать приложение делающее преобразования: размеров одежды и обуви разных странах и сантиметрах (в разных направлениях). Размеров не меньше 6. Вывод информации на разных фонах.

Вариант 23

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного спортсмена, проживающего в городе, и его основные достижения. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

4. Разработать приложение делающее преобразования: меры Британских длин в русские (в разных направлениях. (Мер длин не меньше 6) Вывод информации на разных фонах.

Вариант 24

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение поэта, проживающего в городе, и название одного из его произведений. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Россия.

4. Напишите приложение, которое на основе выбора типа геометрической фигуры: пирамида, параллелепипед, шар. Принимает на вход параметры сторон и рассчитывает объем фигуры. Вывод информации на разных фонах.

Вариант 25

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение герба города, флага, и страну, в которой город находится. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Румыния.

5. Напишите приложение, которое принимает на вход дату и время в часах и минутах и в зависимости от выбора из вариантов Количество дней, Количество

минут, Количество секунд вывод количество дней с начала года, количество минут с начала дня и количество секунд с начала часа.

Вариант 26

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение музыканта или певца, проживающего в городе, и название одного из его произведений. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Украина.

4. Разработать приложение: добавьте кнопку результата, текстовые поля (ввод a, c), Segmented Control (выбор целых или дробных коэффициентов). При щелчке по кнопке – вычисление функции $y=ax^2+c$. В текстовых полях: входные данные. Вывод информации на разных фонах.

Вариант 27

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного ресторана, который находится в городе, и вид кухни. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Польша.

4. Разработать приложение делающее преобразования: меры морских длин в русские (в разных направлениях. (Мер длин не меньше 6) Вывод информации на разных фонах.

Вариант 28

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение спортивного объекта города, и виды спорта, которые проводятся в этом объекте. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Разработать приложение «калькулятор»: вычисление тригонометрических функций (косинус, синус, тангенс, котангенс). Вывод информации на разных фонах.

Вариант 29

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение музыканта или певца, проживающего в городе, и название одного из его произведений. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Украина.

4.Разработать приложение делающее преобразования: меры Британских длин в русские (в разных направлениях. (Мер длин не меньше 6) Вывод информации на разных фонах.

Вариант 30

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного спортсмена, проживающего в городе, и его основные достижения. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

4.Разработать приложение: добавьте кнопку результата, текстовые поля (ввод а,с), Segmented Control (выбор целых или дробных коэффициентов). При щелчке по кнопке – вычисление функции $y=ax^2+c$. В текстовых полях: входные данные. Вывод информации на разных фонах.

Вариант 31

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение госпиталя (больницы) города, и адрес расположения больницы. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Индия.

4.Разработать приложение делающее преобразования: меры морских длин в русские (в разных направлениях. (Мер длин не меньше 6) Вывод информации на разных фонах.

Вариант 32

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного ресторана, который находится в городе, и вид кухни. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Польша.

5. Разработать приложение делающее преобразования: размеров одежды и обуви разных странах и сантиметрах (в разных направлениях). Размеров не меньше 6. Вывод информации на разных фонах.

Вариант 33

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение главы города, и количество населения. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Китай.

4.Разработать приложение делающее преобразования: меры морских длин в русские (в разных направлениях. (Мер длин не меньше 6) Вывод информации на разных фонах.

Вариант 34

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение спортивного объекта города, и виды спорта, которые проводятся в этом объекте. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Китай.

5. Напишите приложение, которое поможет слушателям начальной школы изучить сложение и вычитание двузначных и трехзначных чисел. Тип числа и тип операции задаётся с помощью переключателя. Значения чисел вводятся.

Вариант 35

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение герба города, флага, и страну, в которой город находится. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Румыния.

4. Разработать приложение «калькулятор»: вычисление тригонометрических функций (арккосинус, арксинус, арктангенс, аркотангенс). Вывод информации на разных фонах.

Вариант 36

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного ресторана, который находится в городе, и вид кухни. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Напишите приложение, которое принимает на вход дату и время в часах и минутах и в зависимости от выбора из вариантов Количество дней, Количество минут, Количество секунд вывод количество дней с начала года, количество минут с начала дня и количество секунд с начала часа.

Вариант 37

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение известного музея, который находится в городе, и название музея. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Грузия.

4. Разработать приложение «калькулятор»: преобразование из десятичной системы счисления в другую (2с\с, 8 с\с, 16с\с). В разных направлениях. Вывод информации на разных фонах.

Вариант 38

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение университета города, и количество

факультетов и студентов в нем. В приложении должны быть не менее двух городов из каждой области и не менее трех областей, одна из которых Минская и университет — БГПУ.

5. Написать приложение для расчета скидки за покупку в зависимости от количества товаров в покупке или стоимости покупки. Предоставлять следующие виды скидок: по количеству (1% - 5 товаров, 3% - 10 товаров) и стоимости (3% - > 50 руб, но <100 руб, 5% - > 100руб). Поле segmented control использовать для выбора типа скидки.

Вариант 39

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение писателя, проживающего в городе, и название одного из его произведений. В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Беларусь.

5. Написать приложение для определения стоимости переговоров, если стоимость 1 минуты на территории Беларуси равна 0.15 руб, с Россией – 0.90 руб, а в субботу и воскресенье дополнительно предоставляется скидка 10%. Ввести данные – dt (продолжительность разговора в минутах), s (стоимость минуты разговора). День недели d (день недели от 1 до 7) выбирать в переключателе segmented controls.

Вариант 40

2.2. Дополните приложение на Objective-C из 2.1 следующим функционалом, при выборе города выводит изображение лучшего парка города, и название парка и его характеристики (наличие памятников, дата основания и т. д.). В приложении должны быть не менее **двух** городов из каждой страны и не менее **двух** стран, одна из которых Литва.

5. Написать приложение для определения плотности материала в зависимости от его типа. Ввести объем и массу тела, тип материала выбрать из перечня. Использовать 5 видов материалов.