**Вариант 4.**

2. Напишите программу, которая вычисляет сумму цифр целого числа. Например, сумма цифр числа 2155 равна 2+1+5+5=13. Допускается ввод любого числа.

3. Написать программу, которая при вводе числа в диапазоне от 1 до 99 добавляет к нему слово "копейка" в правильной форме. Например, 1копейка, 5 копеек, 42 копейки

5. Словарь с данными об абитуриентах enrolee(name, average\_mark), и вузах vuz(name, passing\_score).

**ЗАДАНИЕ 1. ЗНАКОМСТВО СО СРЕДОЙ ПРОГРАММИРОВАНИЯ XCODE»**

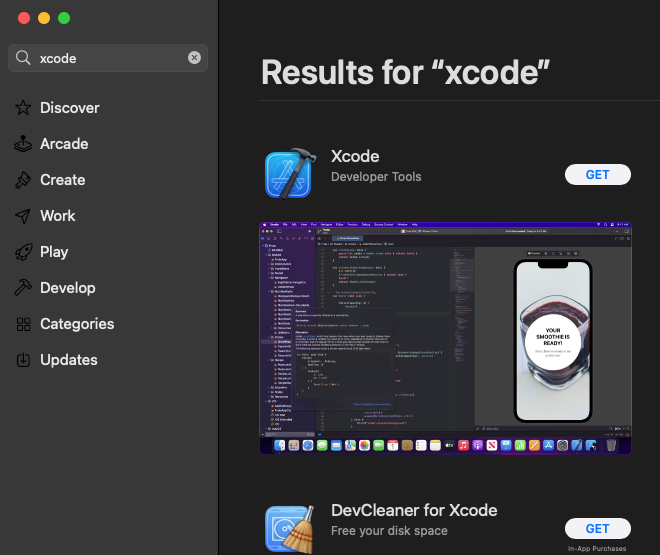
**Цель задачи:** Изучить среду Xcode и научиться создавать проект iOS.

**Задачи:** Создать проект iOS на языке swift. Научиться запускать iOS Simulator. Подключить свой Apple ID в настройках Xcode. Включить результаты выполнения в отчет лабораторной работы 6.

**Ход работы:**

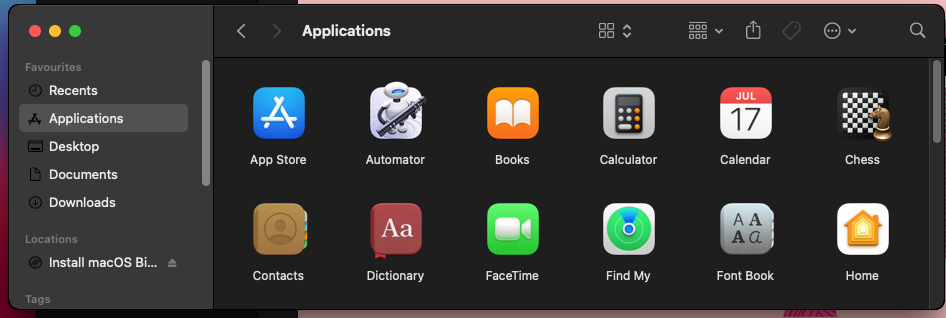
* 1. *УСТАНОВКА СРЕДЫ ПРОГРАММИРОВАНИЯ XCODE*

Для того, чтобы установить среду разработки Xcode, открываем приложение App Store. В поиске набираем Xcode и открываем страницу приложения (рис.1). В левом верхнем углу, под иконкой приложения, нажимаем кнопку “Загрузить”. Ждем окончания загрузки.



* 1. *ЗАПУСК СРЕДЫ ПРОГРАММИРОВАНИЯ XCODE*

После завершения загрузки переходим в папку “Программы” на вашем компьютере Mac, либо нажимаем клавишу F4. Находим приложение Xcode и запускаем его (рис 2).



Появляется окно запуска (рис. 3), где мы можем:

- Быстро открыть недавние проекты (таблица в правом углу), либо открыть уже существующий проект;

- Открыть Playground (“песочница” для языка программирования Swift, где мы можем поэкспериментировать с ним) ;

- Создать новый проект Xcode

- Просмотреть созданные проекты (начать работу с созданным ранее проектом)

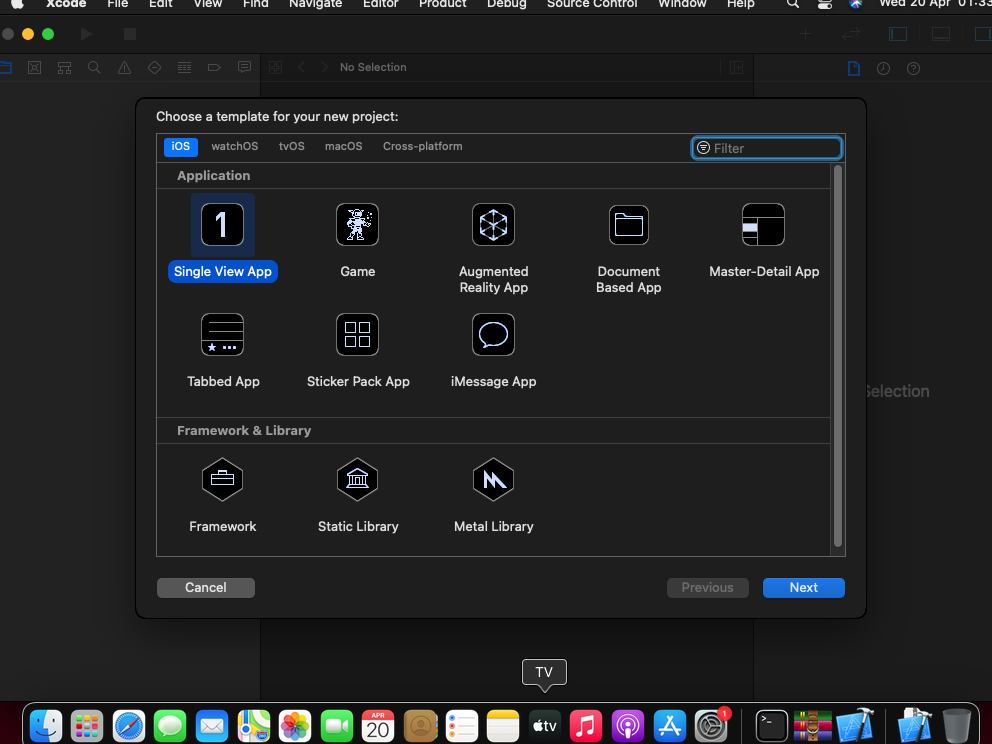
Для того, чтобы это окно больше не открывалось при запуске, можно убрать галочку с параметра “Show this window when Xcode launches”.



* 1. *СОЗДАНИЕ ПРОЕКТА В СРЕДЕ ПРОГРАММИРОВАНИЯ XCODE*

Создаем новый проект, нажав на “Create a new Xcode project”.(рис.4)

Здесь мы можем выбрать шаблон, по которому будем создавать приложение для iOS (iPhone, iPad, iPod touch), watchOS (смарт-часы Apple Watch) или OS X (MacBook, MacBook Air, MacBook Pro, Mac mini, Mac Pro, iMac). Для каждой операционной системы имеется несколько шаблонов.



Выбираем iOS – Application – Single View Application (приложение с одним контроллером View Controller).

Нажимаем на кнопку “Next” и попадаем в окно выбора свойств нашего приложения (рис.5).

Здесь мы видим следующие параметры:

1) Product Name – название приложения

2) Organization Name – название компании разработчиков, либо имя индивидульного разработчика

3) Organization Identifier – идентификатор организации, он же используется как идентификатор сборки (Bundle Identifier)

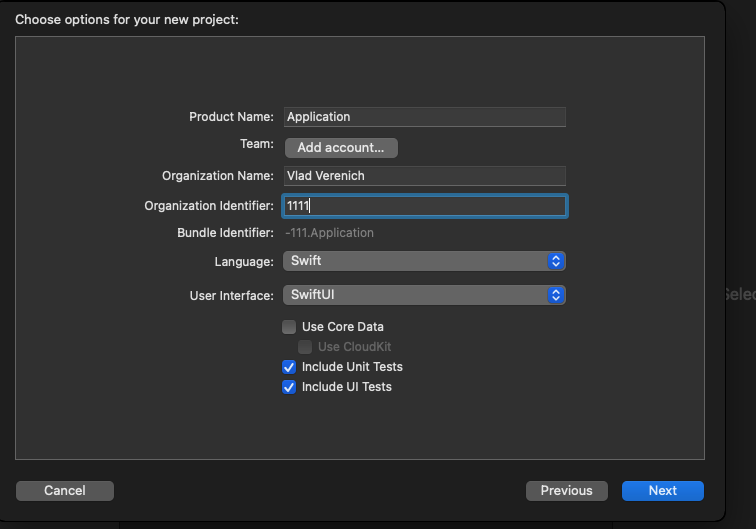
4) Language – язык программирования

5) Devices – устройство, на которое будет нацелен проект (проект может быть нацелен сразу на iPhone и iPad – параметр Universal)

6) Use Core Data – использовать базу данных Core Data (встроенное в Xcode средство для работы с базами данных)

7) Include Unit Tests – включить модульное тестирование (процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы)

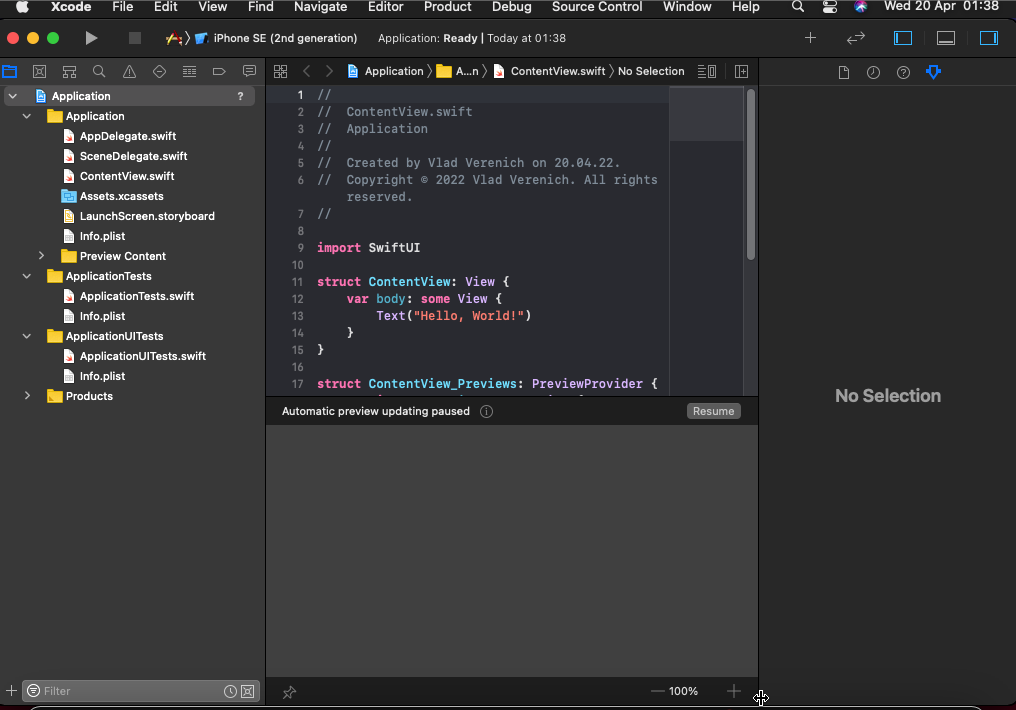
8) Include UI Tests – включить тестирование интерфейса пользователя



После того, как мы задали параметры для нашего приложения, нажимаем “Next”. Попадаем в окно выбора месторасположения нашего проекта, выбираем его и нажимаем “Create”.

* 1. *ОБЗОР ИНТЕРФЕЙСА СРЕДЫ ПРОГРАММИРОВАНИЯ XCODE И IOS SIMULATOR*

Интерфейс среды программирования Xcode



Все параметры приложения заданы, шаблон выбран – мы попадаем в сам интерфейс Xcode (рис.6).

Главная часть интерфейса – это три области: (слева направо) область навигации, область редактирования и область утилит. В верхней части: (слева направо) кнопка компилирования и запуска приложения, выбор iOS симулятора и параметры схемы компилятора, строка состояния, кнопки редактора и кнопки вида.

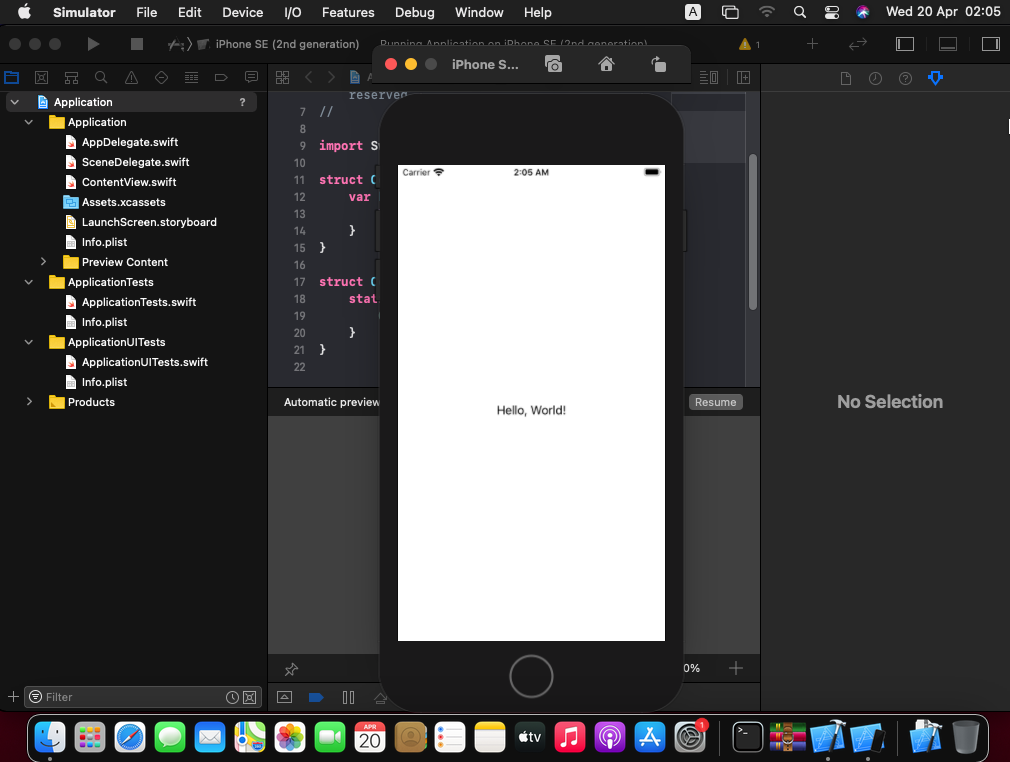
iOS Simulator

iOS симулятор позволяет запускать написанные приложения на симуляторах различных устройств. Компания Apple настаивает на том, чтобы разработчики использовали инструменты только новейших операционных систем, в данном случае – iOS 9.

Для выбора устройства откройте список устройств в схеме компилятора. (рис.7). iOS Device в самой верхней части списка позволяет запустить приложение на вашем устройстве.

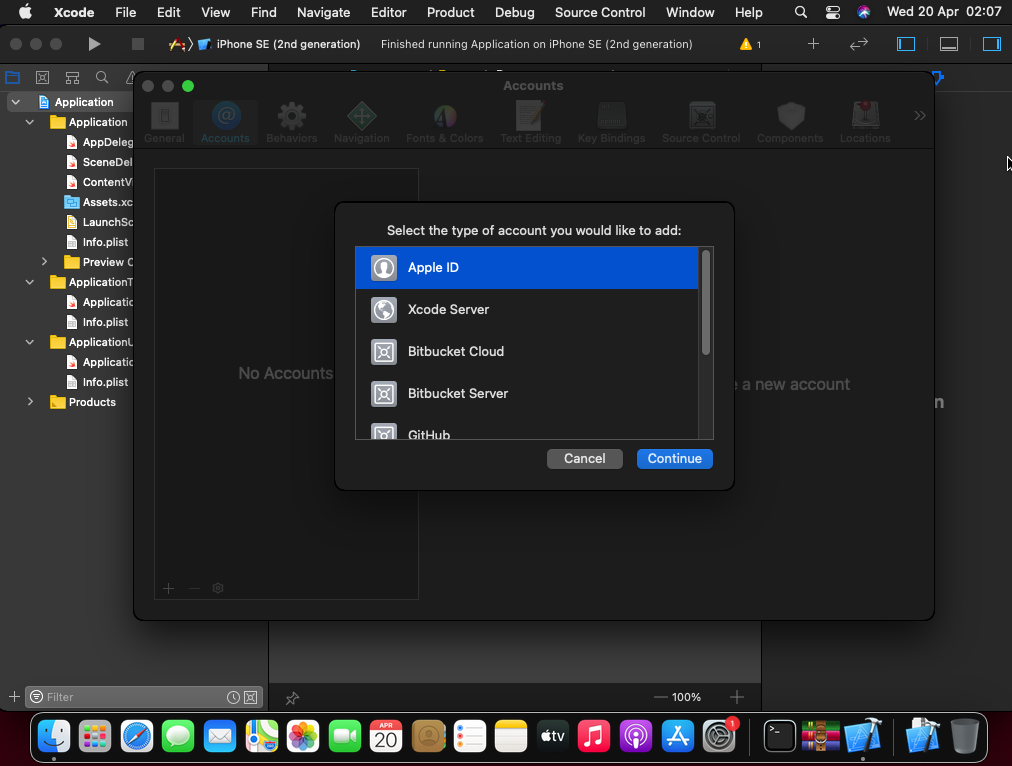
* 1. *КОМПИЛЯЦИЯ И ЗАПУСК ПРИЛОЖЕНИЯ НА IOS SIMULATOR*

Для запуска приложения в симуляторе iOS-устройств выполните команду cmd+R на клавиатуре, либо нажмите на кнопку сборки и запуска проекта (Build and then run the current scheme) в правом верхнем углу интерфейса (рис.8).

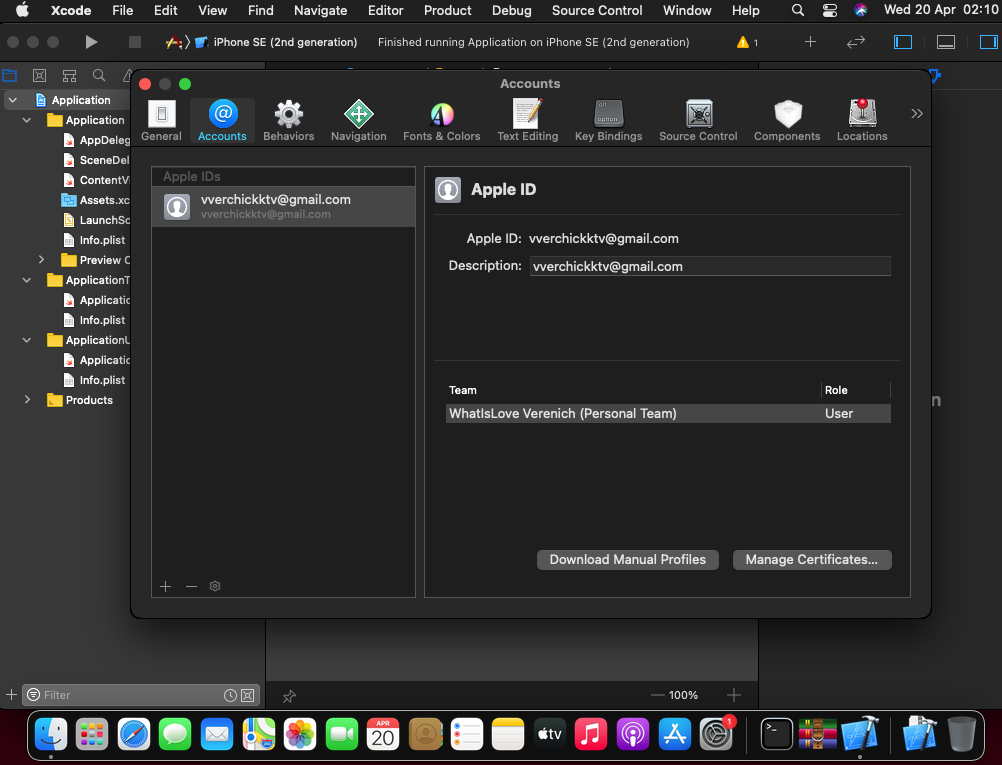


*1.6. ПРИВЯЗКА СВОЕГО APPLE ID К XCODE*

Для этого выбираем в верхней строке “Xcode” – “Preferences” – “Accounts”.



В левом нижнем углу списка Apple IDs нажимаем “+” – “Add Apple ID…”, вводим данные своего Apple ID и нажимаем кнопку “Add”. Мы видим, как в списке Apple IDs появился наш аккаунтом (рис.10). Тем самым мы привязали наш Apple ID к Xcode и можем запускать приложения на любых iOS устройствах, которые привязаны к этому же Apple ID.



*ВОПРОСЫ*

1. Какие способы запуска симулятора iOS возможны?

**Ответ:**

Для запуска приложения в симуляторе iOS-устройств выполните команду cmd+R на клавиатуре, либо нажмите на кнопку сборки и запуска проекта (Build and then run the current scheme) в правом верхнем углу интерфейса

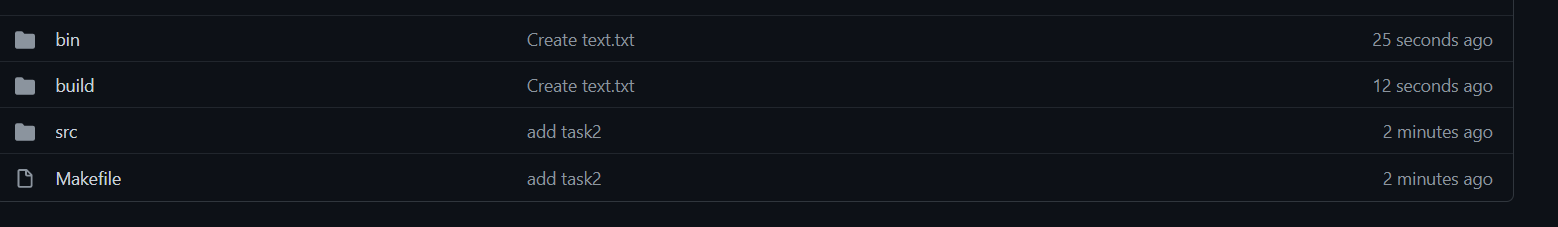
1. Как управлять симулятором?

**Ответ:**

Нажатием на контролы в появившемся окне приложения

**ЗАДАНИЕ 2. КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ НА OBJECTIVE-C**

**Цель** — познакомиться с синтаксисом языка Objective-C и разработать простейшее консольное приложение **с использованием модели КИС** для проектов на языке Objective-C. Продемонстрировать использование объектов.



1) Изучить учебные материалы по созданию консольных приложений на языке Objective-C в Xcode:

1. <https://www.raywenderlich.com/2690-learn-to-code-ios-apps-1-welcome->toprogramming — Изучить обязательно!!!

2. Подборка книг по Objective-C — Изучить обязательно!!!

3. https://iphonelearning.wordpress.com/2011/08/25/creating-first-console-app-usingxcode/

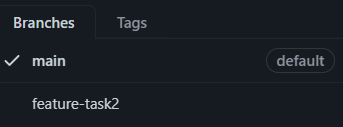
4. <https://andybargh.com/create-your-first-objective-c-application/>

5. <http://www.informit.com/articles/article.aspx?p=1315356>

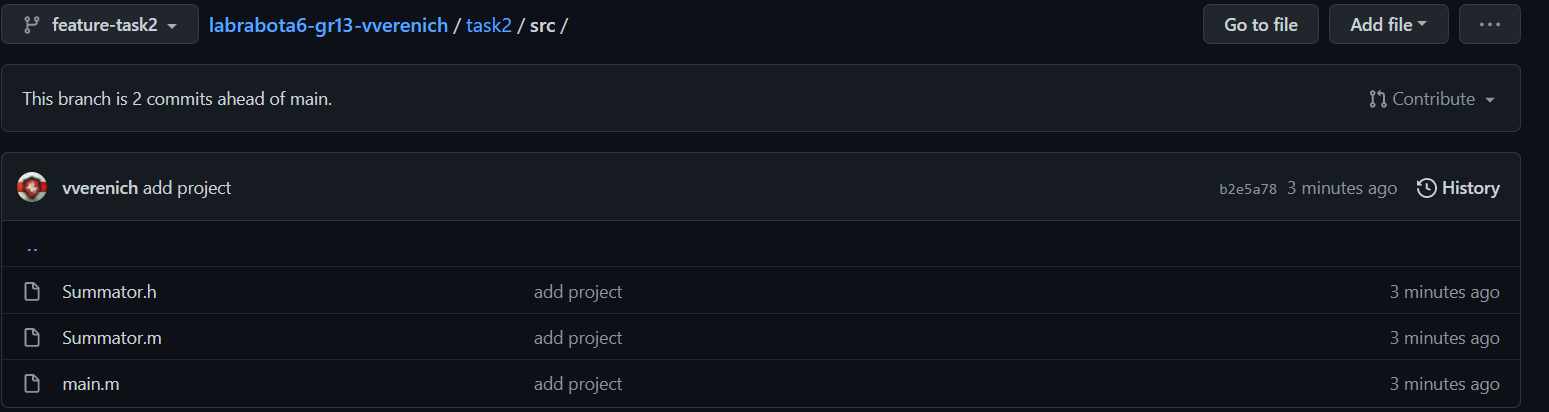
2) Реализовать приложение на Objective-C согласно варианту.



3) Опубликовать код приложения в git-репозиторий в каталоге task2 и в ветке feature-task2.







4) Предоставить протоколы тестов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Входные данные | Ожидаемые выходные данные | Действительные выходные данные | Тест пройден |
| 1 | 234 | 9 | 9 | Да |
| 2 | 3245 | 14 | 14 | Да |
| 3 | 55 | 10 | 10 | Да |
| 4 | 100 | 1 | 1 | Да |

5) Ответить на вопросы.

ВОПРОСЫ

1. Какие расширения файлов у следуюших типов файлов?

• C language source file **ОТВЕТ: .c**

• C++ language source file **ОТВЕТ: .cpp**

• Header file **ОТВЕТ: .h**

• Objective-C source file **ОТВЕТ: .m**

• Objective-C++ source file **ОТВЕТ: .mm**

• Object (compiled) file **ОТВЕТ: .o**

1. Способы вывода на экран в Objective-C

**ОТВЕТ:**

NSLog, printf(from C)

**ЗАДАНИЕ 3. КОНСОЛЬНЫЕ ПРИЛОЖЕНИЯ НА SWIFT: СПОСОБЫ СОЗДАНИЯ**

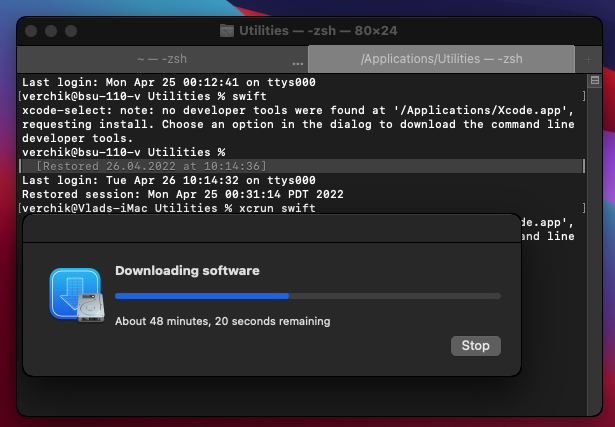
**Цель** — познакомиться с консолью Swift REPL, компиляцией консольных приложений с помощью swiftc, созданием консольных приложений в xcode в playground; продемонстрируйте навыки работы с консолью Swift REPL и Playground2 , компиляцию приложений с помощью swiftc, создание и тестирование приложений в playground и создание консольных приложений.

1. Изучите примеры

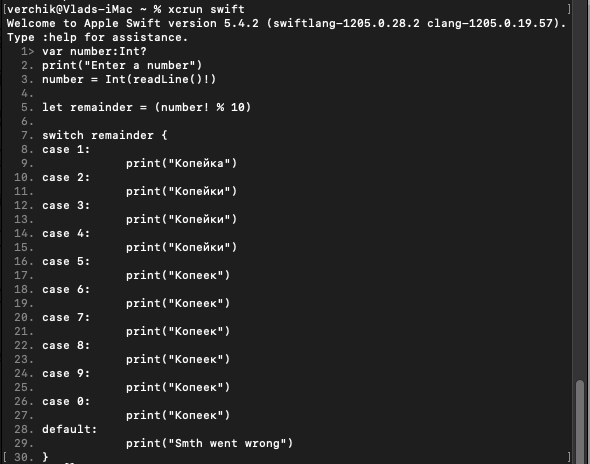
◦ из главы 2 книги Learn Swift http://www.aidanf.net/learn-swift/running\_code.

◦ <https://developer.apple.com/swift/blog/?id=18>

◦ <https://swift.org/blog/swiftpm-repl-support/>

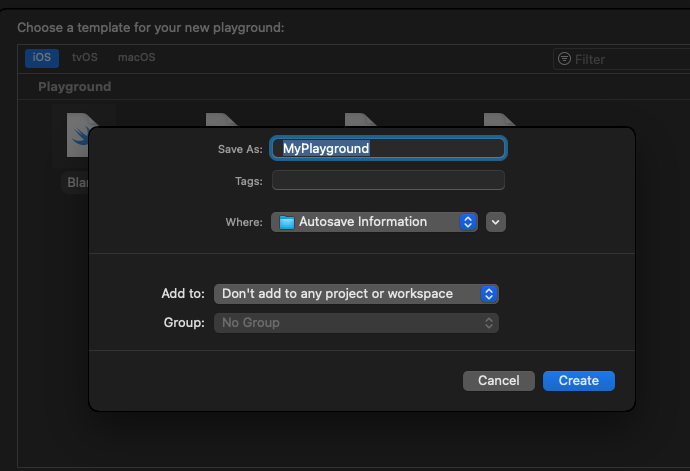


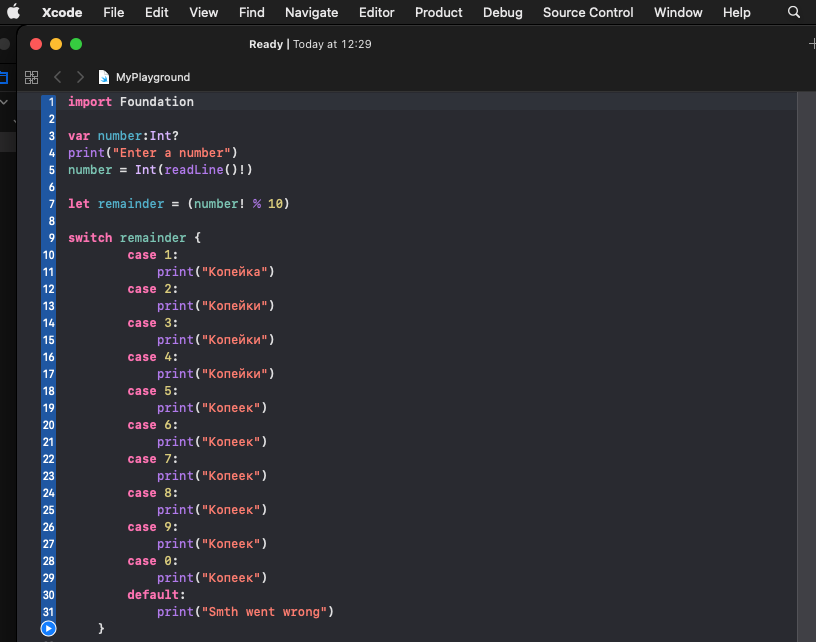
1. Разработать приложение на языке Swift согласно варианту в консоли Swift REPL и в Xcode Playground.



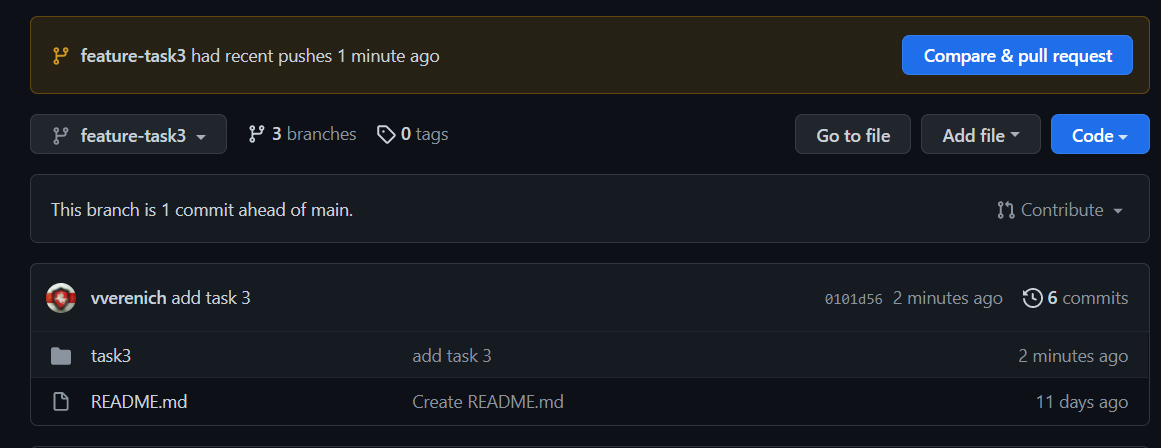


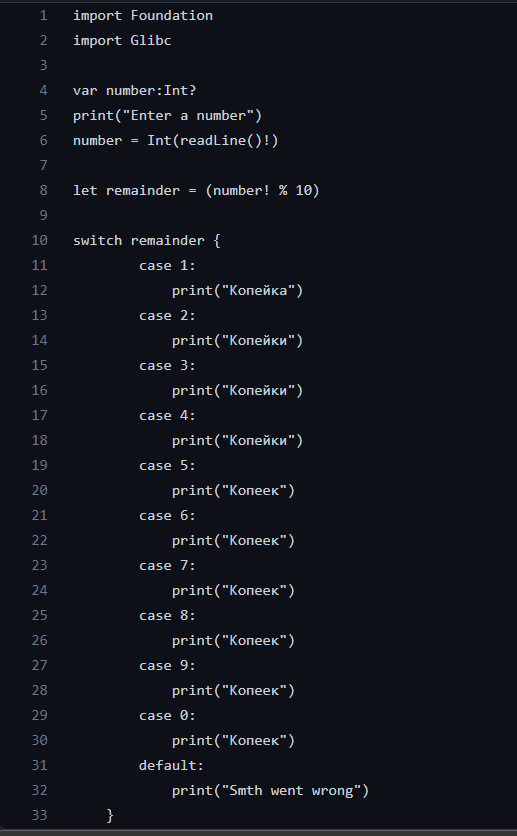
1. Продемонстрировать навыки работы с консолью Swift REPL, компиляцию приложений с помощью swiftc, создание и тестирование консольных приложений в playground.





4. Опубликовать код приложения в git-репозиторий в каталоге task3 в ветке feature-task3.





5. Предоставить протоколы тестов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Входные данные | Ожидаемые выходные данные | Действительные выходные данные | Тест пройден |
| 1 | 1 | Копейка | Копейка | Да |
| 2 | 2 | Копейки | Копейки | Да |
| 3 | 5 | Копеек | Копеек | Да |
| 4 | 0 | Копеек | Копеек | Да |

6. Включить скриншоты и код приложений в отчет.

*Всё включено!*

7. Ответить на вопросы.

ВОПРОСЫ

1. Что такое swift REPL и в каких случаях рекомендуется использовать?

**ОТВЕТ:**

Он позволяет нам сделать разбор написанного кода в режиме runtime. Удобно экспериментировать с небольшим по размеру кодом и выполнять debug.

1. Что такое playgraund? Какие задачи (проекты) можно создавать в playground?

**ОТВЕТ:**

Playground, или в переводе на русский «песочница», представляет из себя отличное окружение для того, чтобы изучить Swift. Выглядит это так: вы вводите строку кода и моментально видите результат его выполнения. После того, как вы убедитесь, что написанный вами код дает нужный результат, вы можете просто взять и перенести его в свой проект.

Например, при помощи Playground вы можете решить следующие задачи: освоить синтаксис Swift поэкспериментировать с новыми API, чтобы улучшить навыки программирования разработать новый алгоритм и понаблюдать за каждым шагом его выполнени провести простые арифметические вычисления.

1. Как компилировать приложение на языке swift в консоли bash?

**ОТВЕТ:**

## % swiftc -o file file\_name.swift

## % ./file\_name

**ЗАДАНИЕ 4. ПОДКЛЮЧЕНИЕ РЕПОЗИТОРИЯ И ОРГАНИЗАЦИЯ РАБОТЫ С ВЕТКАМИ В XCODE**

Цель — изучить как работать с git-репозиторием в Xcode и подключить репозиторий для публикации кода в задании 5.

Изучить примеры из статей:

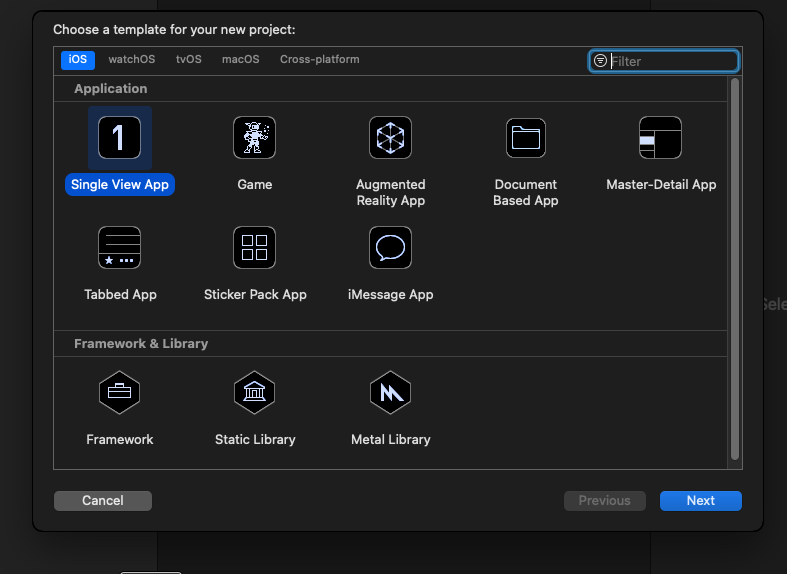
1. <https://gist.github.com/quoha/4587902>

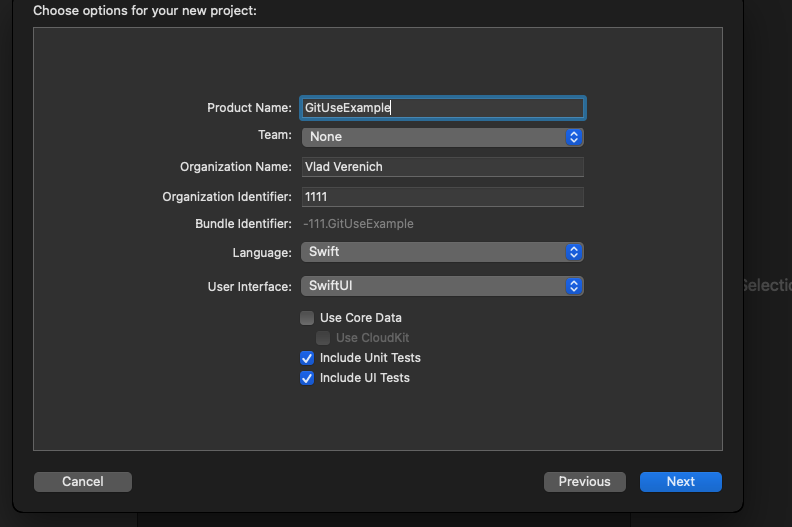
2. <https://www.raywenderlich.com/675-how-to-use-git-source-control-with-xcode-9>

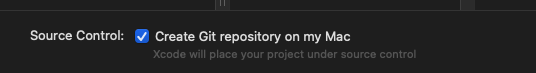
3. https://medium.com/xcblog/the-marriage-of-github-and-xcode-9-at-wwdc821b78eb6797

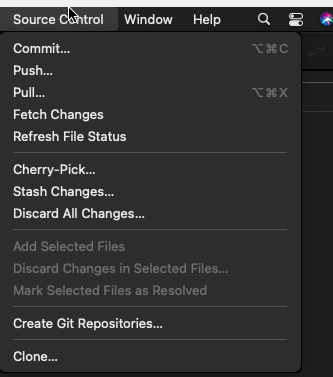
4. <http://blog.chrishannah.me/putting-your-xcode-project-on-github-the-easy-way/>

5. https://medium.com/technology-nineleaps/integration-of-xcode-9-with-githuba350bf563278









ВОПРОСЫ

1. Как создать локальный git-репозиторий для проектов в Xcode?

**ОТВЕТ:**

При запуске проекта выбрать опцию “ создать репозиторий”

1. Как добавить внешний репозиторий в Xcode?

**ОТВЕТ:**

Зайти  в source control , выбрать clone.

1. Как создать ветку репозитория в Xcode?

**ОТВЕТ:**

Выбрать Branch from master(если текущая ветка - master)

1. Как отменить commit в Xcode?

**ОТВЕТ:**

ишем git hist, получая список всех коммитов в ветке

Далее находим по комментарию нужный коммит и копируем 7 символов хэша этого коммита.

 Далее git reset --hard имя\_коммита\_перед\_неверным

Таким образом мы сбросим ветку до последнего верного коммита.

1. Как слить ветки в Xcode?

**ОТВЕТ:**

Выбрать merge into Branch

**ЗАДАНИЕ 5. КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ОБРАБОТКИ ТЕКСТОВЫХ ДАННЫХ**

Цель — разработать консольное приложение для обработки текстовых данных с помощью словаря.

**Входные данные**

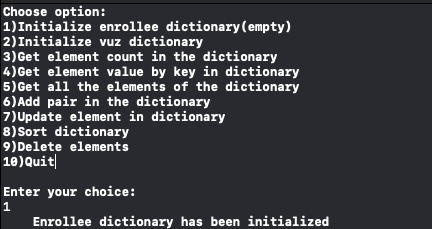
На вход программе подается список слов, разделенных пробелами.

**Выходные данные**

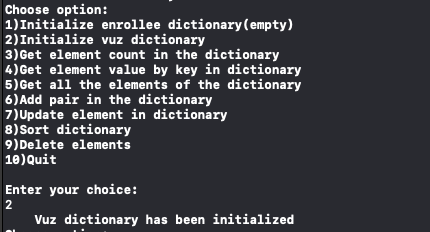
Согласно варианту задания.

Требуется реализовать приложение, которое будет

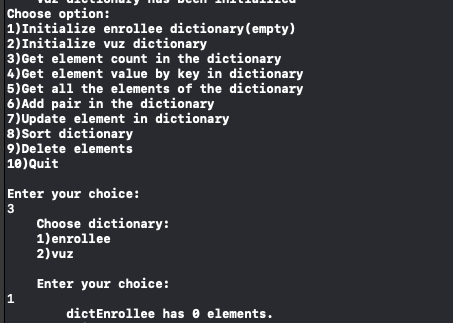
• инициализировать пустой словарь (для первого словаря из задания);

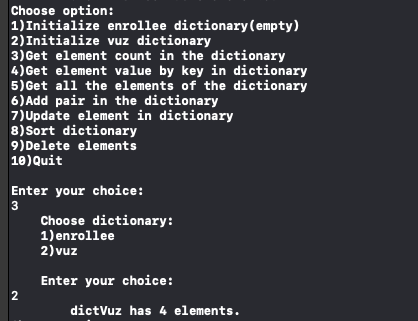


• инициализировать словарь с данными (для второго словаря из задания);

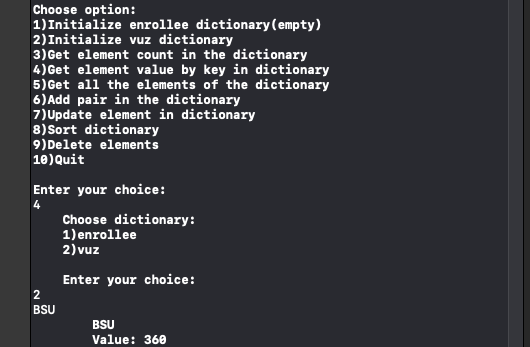


• подсчитывать количество элементов в словаре

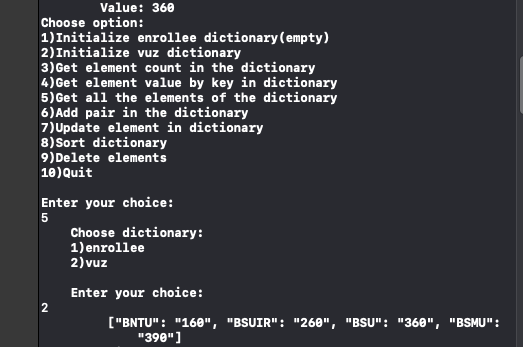




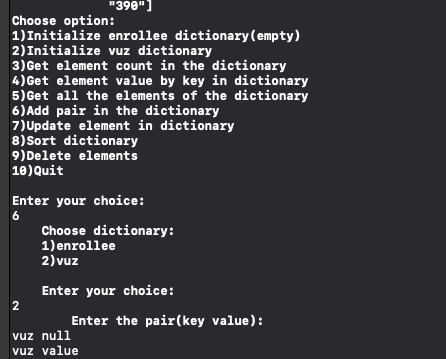
• выводить значение и ключ из словаря по ключу;



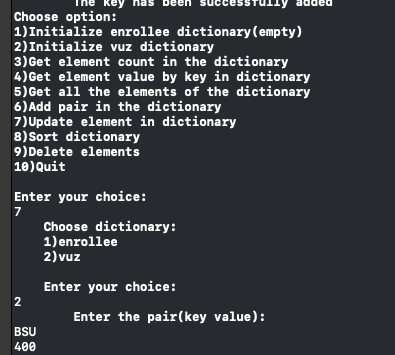
• выводить все элементы (ключ-значение) из словаря;

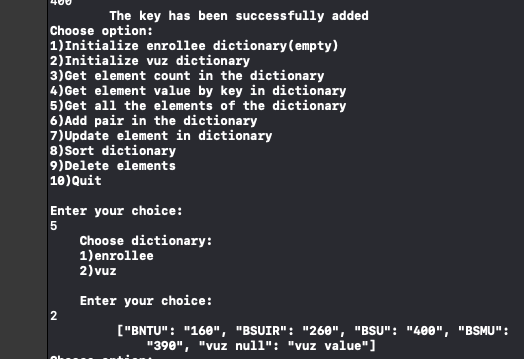


• добавлять элемент в словарь;

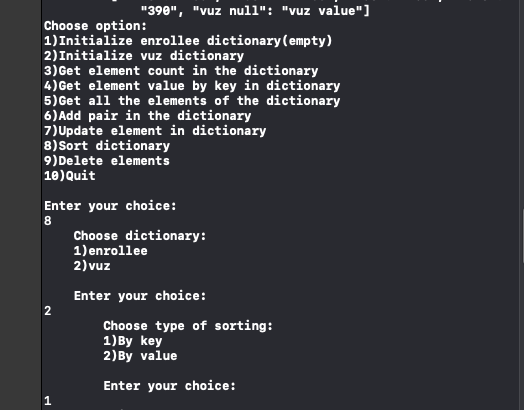


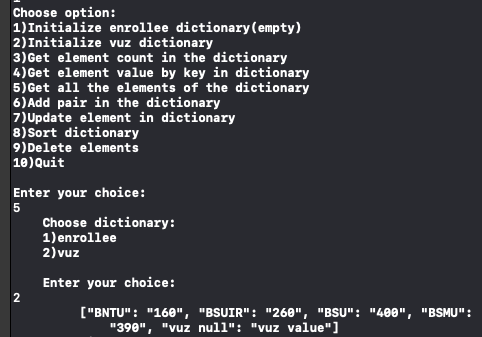
• модифицировать элементы словаря;



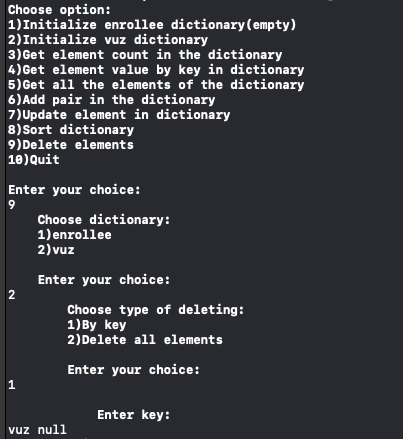


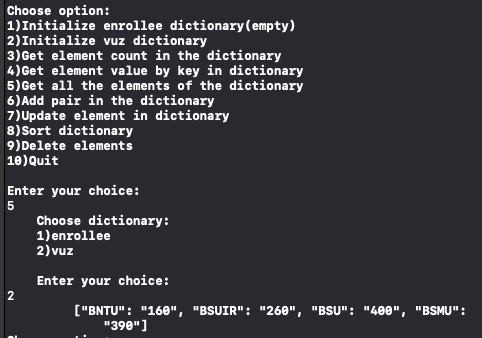
• сортировать словарь по ключу и по значению;





• удалять один и все элементы словаря;





Для вывода можно использовать функцию print(), это аналог функции printf().

Для решения задачи:

1. Подборка книг по разработке приложений на языке Swift

2. Изучите примеры консольных приложений из статей:

◦ https://www.raywenderlich.com/163134/command-line-programs-macos-tutorial2

◦ <https://o7planning.org/en/10573/swift-tutorial-for-beginners>

2. Изучите типы коллекций согласно документации <https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/CollectionTypes.html>

<https://developer.apple.com/documentation/swift/dictionary> и статьям <https://makeapppie.com/2016/06/24/how-to-use-dictionaries-in-swift-3-0/> — Обязательно для изучения!

<https://www.dev2qa.com/how-to-use-dictionary-in-ios-swift/> — Обязательно для изучения!

<https://habr.com/ru/post/275911/> — Обязательно для изучения!

<https://andybargh.com/dictionaries-in-swift/> — Обязательно для изучения!

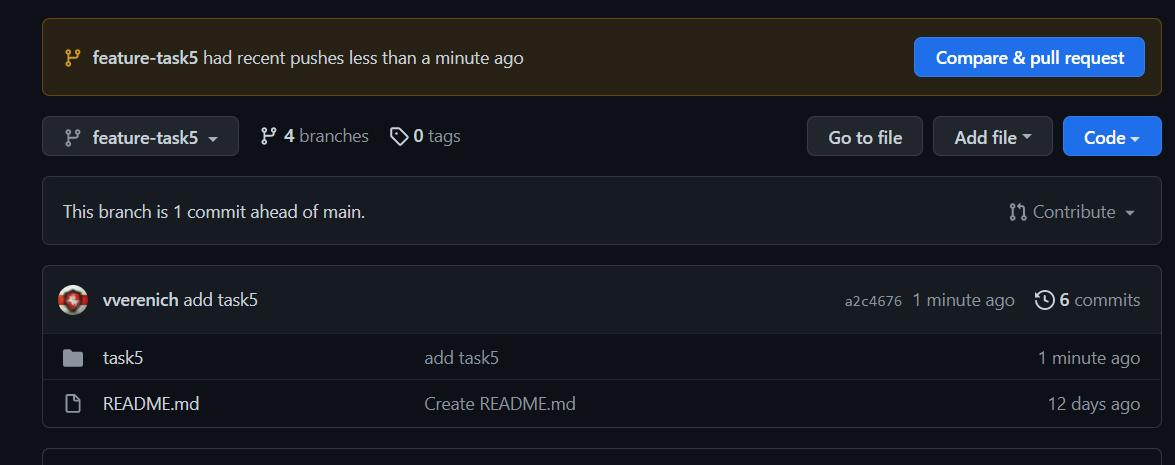
<https://iosrevisited.blogspot.com.by/2017/08/dictionariesios-in-swift-4-howtouse.html>

<https://swift.org/blog/dictionary-and-set-improvements/> <https://www.agnosticdev.com/content/how-work-dictionaries-swift-4>

3. Изучите методы сортировки строк <https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/Strings/Articles/SearchingStrings.html>

4. Создайте консольный проект в Xcode согласно варианту.

5. Опубликуйте код приложения в репозитории.



6. Предоставить протоколы тестов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Входные данные | Ожидаемые выходные данные | Действительные выходные данные | Тест пройден |
| 1 | 2, 5 2 | [“BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”] | [“BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”] | Да |
| 2 | 6 2 BarSU 200 | [“BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”,  “BarSU”:”200”] | [“BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”,  “BarSU”:”200”] | Да |
| 3 | 8 2 1 | [“BarSU”:”200”, “BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”] | [“BarSU”:”200”, “BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”] | Да |
| 4 | 9 2 BarSU | [“BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”] | [“BNTU”:”160”, “BSUIR”:”260”, “BSU”:”400”, “BSMU”:”390”] | Да |

7. Включить скриншоты и код приложений в отчет.

Включил

8. Ответить на вопросы.

ВОПРОСЫ

1. В чем отличия языка программирования Swift от C и С++.

**ОТВЕТ:**

Отсутствуют указатели (то есть язык является более безопасным). Присутствуют именованные параметры

1. Какие способы отладки программ поддерживает среда xCode?

**ОТВЕТ:**

Breakpoint, Breakpoint - это точка, которая позволяет остановить программу в некоторый момент времени и выполнить некоторые действия над запущенной программой.

LLDB - консольный отладчик

1. Какие операторы языка swift Вы знаете?

**ОТВЕТ:**

Унарные, бинарные, тернарные (условный оператор: выражение ? действие1 : действие2). Операторы сравнения, арифметические операторы, логические операторы.

1. Какие типы коллекций применяются в языке Swift и в чем их особенности?

**ОТВЕТ:**

Массивы, Множества и Словари

1. Какие типы проектов xCode Вы знаете?

**ОТВЕТ:**

Single View Application – шаблон с одним готовым окном

Empty Application – пустой шаблон для создания приложения с нуля Tabbled Application – для проекта с несколькими вкладками OpenGL Game – для игр на основе OpenGL.