

Урок 1. Переменные

- 1) Постарайтесь по больше по практиковаться, повторите то что есть в видео.
- 2) Зайти обязательно и познакомиться с документацией - <http://swiftbook.ru/doc>.
- 3) По-создавайте свои переменные и константы
- 4) Дз сдать на дейли в gist

Урок 2. Типы данных

- 1) Постарайтесь больше попрактиковаться, повторите то, что было в видео.
- 2) Зайти обязательно и познакомиться с документацией - <http://swiftbook.ru/doc>.
- 3) Описать в гисте в комментарии что такое псевдоним типа.
- 4) Создать несколько псевдонимов на разные типы.
- 5) Создайте различные выражения с приведением типа
- 6) Так же как в математике поиграйтесь с выражениями - + /
- 7) Описать в гисте кратко что такое целые числа
- 8) Описать и привести примеры что такое числовой литерал
- 9) Описать какие в Swift существуют дополнительные целочисленные типы данных
- 10) Кратко описать что такое юникод-скаляры, написать

несколько выражений

11) Что такое Хешируемые и сопоставимые типы данных.

12) Написать примеры хешируемых типов данных. 13) Что такое хеш? Дать определение.

14) Объяснить разницу между Float и Double. 15) Прочитать книгу усова с 77 по 112 стр.

Урок 3. Строки

1) Постарайтесь по больше по практиковаться,

повторите то что было в видео.

2) Зайти обязательно и познакомиться с документацией

на тему строки - <http://swiftbook.ru/doc>.

3) Написать примеры многострочных литералов строк

4) Распечатайте каждую букву своего имени с помощью for in

5) Прибавьте к имени свою фамилию

6) С помощью интерполяции прибавьте свой возраст

7) "hello there!" удалите лишний пробел после слова "hello" 8) Теперь распечатайте "hello there! задом наперед

8) Теперь распечатайте "hello there! задом наперед

9) Проверьте есть ли в "hello there! буква "o"

10) "23:51:01" разделите эту строку на 3 элемента и распечатайте каждый элемент в формате Int.

11) Отформатируйте Float 3.14159 что бы стало 3.14

12) "AaBbCc" сделайте все буквы большие

13) "AaBbCc" удалите первую и последнюю букву.

14) "AaBbCc" вставьте в конец "!"

15) Прочитать книгу усова на тему строки

Урок 4. Массивы

1) Повторите то что было в видео.

2) Зайти обязательно и познакомиться с документацией на тему массивы - <http://swiftbook.ru/doc>.

3) Создать массив из чисел, числа это купюры в долларах посчитать количество денег и вывести в консоль.

4) - создать массив "дни в месяцах" -элементов содержащих используя цикл for вывести количество дней в каждом мес.

- используйте еще один массив с именами месяцев чтобы вывести название месяца + количество дней.

- сделайте тоже самое, но используя массив тюплов с

параметрами (имя месяца, кол-во дней).

- сделайте тоже самое, только выводите дни в

обратном порядке (порядок в массиве не меняется).

5) Создайте массив из 30 повторяющихся элементов.(не руками)

6) Создайте массив из разных типов д. (String, Int, Bool и т.д.) 7) Сформируйте новый массив состоящий только из Sting

из предыдущего массива.

8) Преобразуйте 2D массив (двумерный) в одномерный
[["Hello", "Hi"], ["Привет"], [".", "Как", "твой"], ["дела?"]] .

9) проверить на пустоту преобразованный массив и вывести
в консоль содержит ли он элемент "Привет" .

10) Создайте массив типа Int из 10 элементов используя Range
новый массив с элементами под индексами от 3 до 8.

11) Создайте массив типа Int из 10 элементов используя Slice
новый массив с элементами под индексами от 3 до 8.

12) Создайте массив типа String из 8 элементов с помощью
sorted() и sort() отсортируйте и в коментах напишите в чем
разница этих двух методов.

13) Дан массив [22, 41, 23, 80, 33, 83, 56, 42, 11, 98, 120, 6, 84, 1]
выведите максимальный и минимальный элемент. выведите
первый и последний элемент.
удалите первый и последний элемент.

опишите в чем разница dropFirst() dropLast() от removeFirst() и
removeLast().

14) Усов читаем Диапазоны и Массивы стр 132 - 154 (Конспект)

Урок 5. Dictionary

- 1) Повторите то что было в видео.
- 2) Зайти обязательно и познакомиться с документацией на тему словари - <http://swiftbook.ru/doc>.
- 3) Создайте пустой словарь **responseMessages**, для ключа тип числовой, для значения строка.
- 4) Добавьте в пустой словарь ключи 500, 404, 403, 200, а значения "OK", "Access forbidden", "File not found", "Internal server error"
- 5) Создайте массив `httpResponseCodes` с числами 200, 403, 301
- 6) Переберите массив `httpResponseCode` в цикле чтоб при каждой итерации вы проверяли код массива в словаре `responseMessage` и если код такой есть то выводить в консоль ключ и значение, если нет писать неизвестный код и выводить код.
- 7) Отсортируйте все ключи по порядку 200, 403 и тд.
- 8) Распечатайте в цикле все ключи и значения.
- 9) Создайте словарь с ключами `primes`, `triangular`, `hexagonal`, у каждого ключа значение это массив [2, 13, 15, 7, 11, 3, 5], у второго значения массив [21, 6, 3, 10, 15, 1, 28], у третьего [1, 91, 15, 28, 45, 66, 6]

10) Отсортируйте содержимое каждого массива по возрастанию.

11) Создать 2 словаря в 1 ключ и значение «a»: «b», у второго «c»: «d»
нужно объединить эти 2 словаря не используя +=.

12) Создать переменную a = 1 и вторую b = 2, поменять местами значения у 2х переменных 2мя способами.

13) Опишите что такое словарь, почему его иногда называют хеш мап, и это class или struct.

14) Напишите какие протоколы комформит словарь и для чего они (только кратко).

15) Прочитать книгу Усова о словарях

Урок 6. Кортежи

- 1) Повторите то что было в видео.
- 2) Зайти обязательно и познакомиться с документацией
- <http://swiftbook.ru/doc>. и выпишите оттуда около 5 примеров
- 3) Создайте 1 пустой тюпл.
- 4) Создайте тюпл где первое значение 1, второе элемент массива "threee". Распечатайте все значения.
- 5) Создайте тюпл с Int, Bool, String значениями. Распечатайте все значения.
- 6) Создайте тюпл где первое значение строка а второе предыдущий тюпл. Распечатайте все значения.
- 7) Создайте тюпл из значений "Apple", «IBM»
- 8) Добавьте в предыдущий тюпл через = "Google"
- 9) Удалите "IBM"
- 10) Если не получилось сделать 8 и 9 пункт 😊 создайте тюпл внутри которого словарь и повторите упражнение 7, 8, 9 но уже со словарем внутри тюпла.
- 11) Создайте 3 константы имя, фамилия, отчество и объедините их в новом кортеже.
- 12) Написать является ли тюпл коллекцией как массив и словарь.
- 13) Почитать Усова о тюплах.

Урок 7. Optional Type

- 1) Постарайтесь больше практиковаться, повторите то, что

есть в видео.

2) Зайти и обязательно познакомиться с документацией.
Выпишите в ДЗ 5 примеров.

3) Создать 5 констант со СТРОКАМИ которые содержат цифры и посчитайте их используя новые инструменты.

4) Создать 3 константы со значением nil.

5) Создать 5 опциональных типов констант и установите им значения.

6) Перечислите способы извлечения опционала с примерами (guard не использовать)

7) Создайте опциональный массив Int'ов с значениями от 1 до Преобразуйте в новый массив добавив к каждому элементу букву «а». Выведите получившийся массив

8) Дан массив number: [Int]? = [5, 3, 6, 8, 1, 2, 4]. Найти сумму элементов массива и вывести в консоль. Для вычисления суммы использовать функцию высшего порядка.

9) Создайте несколько опциональных значений. С помощью «оператора объединения с nil» извлеките значение.

10) Дан массив numbers: [String]? = ["5", "3", "шесть"]

Если количество элементов в массиве больше 2 тогда преобразовать данный массив в массив [Int] и вывести в консоль

11) Распечатайте свое имя в цикле for.

12) Создайте массив с возрастом всех членов вашей семьи и распечатайте в консоли через цикл for.

13) Читаем Усова про Опционалы и главу 13 в Swift notes for professional

Урок 8. Условные операторы

- 1) Повторите то, что было в видео.
- 2) Зайти обязательно и познакомиться с документацией. Выпишите около 5 примеров.
- 3) Создать цикл в цикле. В первом цикле интервал 200 раз во втором как дойдем до 15 выйти со всех циклов сразу.
- 4) Создать обработку возраста в switch, если человеку от 6 до 17 то выводить, что ему пора нужно в школу, если от 18 до 25, то в универ и так далее, аж до пенсии.
- 5) Сделать 4 е задание через if конструкцию.
- 6) Создать switch, который принимает сразу точку по x и y например (1, 1) и написать условие если x,y по 0 то выводить x,y и написать, что это точка **origin**, если значение (1, 0), то выводим точку и говорим вы на оси x, (0, 1) вы на оси y, если x и y от -2 до 2 то писать вы внутри координат.
- 7) Написать switch в котором 3 кейса 1,2,3 с принтами этих индексов. Когда мы передаем 1 индекс печатать принт 1 го кейса, а когда передаем 2 й индекс то печатать сразу принт 2-го и 3-го кейса.
- 8) Придумать свой switch в котором будет 3 кейса и в одном из них доп. условие с оператором **where**.
- 9) Создайте переменную с значением 10 и через цикл while печатайте свое имя пока значение не станет 0.
- 10) Почитать Усова на эту тему. .(Конспект)

Урок 9. Function

- 1) Повторите то, что было в видео.
- 2) Зайти обязательно и познакомиться с документацией. Выпишите 5 примеров в плейграунд.
- 3) Создать 4 функции:
 - Которая ничего не принимает и ничего не возвращает.
 - Принимает параметры но не возвращает.
 - Принимает параметры и возвращает.
 - Не принимает параметры но возвращает тупл из 3х параметров.
- 4) Создать журнал для учителя, который будет принимать имя студента , профессию и оценку и записывает это все в массив. И внесите 10 студентов туда и распечатаете через цикл for
- 5) Создать функцию которая принимает имя и фамилию, потом положить это в массив и отсортировать и вывести результат в консоль.
- 6) Создать функцию которая принимает параметры и вычисляет площадь круга.
- 7) Создать те пустой массив кошелек, создайте функцию которая принимает массив в параметрах и в теле функции добавляет в этот массив значение 100. Далее вызовите массив и потом распечатайте значение массива.
- 8) Написать функцию в которой можно передавать в параметры каждый раз разное количество Int например иногда 1,5,9 а следующий раз 5,10, 11, 20, 50 функция все это посчитает и вернет сумму этих чисел.
- 9) Написать любую функцию которая вернет throws.
- 10) Почитать Усова на эту тему. (Конспект)

Урок 10. Замыкания

1) Прочитать документацию:

<https://developer.apple.com>

swiftbook.ru Выпишите около 5 примеров

2) Повторить то, что есть в видео.

3) Написать сортировку массива с помощью замыкания, сначала в одну сторону, затем в другую. Вывести всё в консоль.

4) Создать метод который запрашивает имена друзей, после этого имена положить в массив.

Массив отсортировать по количеству букв в имени.

5) Что такое closure

6) Написать что такое хвостовое замыкание. Привести пример.

7) Написать что такое сбегующее замыкание. Привести пример.

8) Написать что такое autoclosures. Привести пример.

9) Создайте переменную типа `()->>()`

В теле кложуры создайте словарь (типы значений на ваш выбор) из 10 элементов и распечатайте пары (ключ: значение)

Вызовите ниже вашу кложуру

10) Создать массив из 10 элементов с помощью `map` каждый элемент умножить на 10, преобразовать в массив `String` с

помощью `compactMap`, отфильтровать с помощью `.filter`.

Оставить значения с количеством символов больше 2

Вывести результат в консоль.

11) Создать функцию которая принимает коэффициенты квадратного уравнения и возвращает его корни.

для примера решаем это уравнение: $x^2 - 8x + 12 = 0$;

12)

```
func closureExample(saySomething: () -> ()) {  
    saySomething()  
}
```

Вызовите ниже функцию и в ее теле вывести произвольный текст. Преобразовать функцию с функцией на скрине к функции с автофункцией. Вызвать и в теле вывести произвольный текст. Описать разницу между двумя вариантами.

13) Написать функцию принимающую два параметра:

первый параметр `[Int]` второй замыкание `(_ result: [String] -> ())`

переданный массив преобразовать к массиву `String` и вернуть в замыкании.

Вызвать функцию и распечатать полученный результат в

консоль.

Функцию вызвать 3 раза:

1) В **ПОЛНОМ** синтаксисе, с указанием типа замыкания, замыкание внутри скобок функции)

2) С неявным возвратом значения (без указания типа), выносом за скобки

3) С сокращенным именем параметра (\$0), выносом за скобки

14) Написать функцию которая будет принимать ключи, выводить ключи и значения словаря (Dictionary). Тип ключа и значения выбирайте сами.

15) Написать функцию, которая принимает пустые массивы(один строковый, второй - числовой) и проверяет их: если пустой - то добавить значения и вывести в консоль.

16) Читаем Усова и заметки программиста про Closure (Конспект)

Урок 11.

- 1) Больше практиковаться, повторите то, что есть в видео.
- 2) Обязательно познакомиться с документацией. Выписать около 5 примеров
- 3) Создать по 1-2 enum разных типов.
- 4) Создать несколько своих enum, названия какие хотите: например, анкета сотрудника - пол, возраст, ФИО, стаж. Используйте switch как в видео.
- 5) Создать enum со всеми цветами радуги. Создать функцию, которая содержит названия разных предметов или объектов. Пример результата в консоли "apple green", "sun red" и т.д.
- 6) Создать функцию, которая выставляет оценки ученикам в школе.
- 7) Создать Enum Vehicles. Добавить в него 3 вложенных Enum: Cars, Ships, Bikes. В каждом Enum добавить по 3 значения.
Каждое значение имеет 2 ассоциативных типа (String, Int).
- 8) Создать функцию, принимающую в качестве параметра элемент Vehicles.Cars.

С помощью switch вывести для найденного кейса ассоциативные значения с припиской «Печать из switch» и вызвать closure, в который передать ассоциативные значения. Closure должен выводить на печать эти значения с припиской «Печать из Closure».

После этого провалиться в следующий элемент switch и вызвать secondClosure(), которая выводит в консоль «Печать из secondClosure»

```
Печать из switch Black, 340  
Печать из closure color Black, speed 340  
Печать из secondClosure closure
```

если передаете элемент, который в свиче последний. Третьей строчки быть не должно.

9) Создать enum House: Int с 5-ю case –ми и задайте им исходные значения.

Создать в House вычисляемое свойство houseHeight, в котором выводить текущее значение * 100

Создать в House функцию в которой выводить на печать значение из вычисляемого свойства.

Урок 12.

1) Повторить то что было в уроке.

2) Создать класс родитель и 2 класса наследника.

3) Написать программу, в которой создать класс *House* в нем несколько свойств - *width*, *height* и несколько методов - *create*(выводит произведение свойств), *destroy*(отображает в принте что дом уничтожен).

4) Создайте класс с методом, который сортирует массив учеников.

- 5) Написать свою структуру и класс, пояснить в комментариях - чем отличаются структуры от классов.
- 6) Объяснить в комментариях есть ли у классов множественное наследование, и пояснить что это.
- 7) Объяснить может ли структуры наследоваться.
- 8) Выписать по 1 пример из документации, 1 из Усова, 1 из заметок профи. Примеры на свое усмотрение.

Урок 13.

- 1) Повторить то, что есть в видео.
- 2) Создать класс "Students", добавить ему property: dateOfBirth, skills. Для тренировки.

Создавайте свои структуры с разными видами свойств:

сохраняемые/не сохраняемые, вычисляемые, свойства типа и т.д.

- 3) Написать структуру "CreateTriangle", с двумя свойствами - угол А, угол С. И создать 2 вычисляемых свойства - те же угол А, угол С.

И если мы записываем значения в эти углы - результатом должно быть значение третьего угла.

Для простоты можно взять прямоугольный треугольник, сумма углов которого равна 180 градусов. 4) Создать структуру "Резюме", у которой есть такие свойства:

- Фамилия,
- Имя.
- Должность,
- Опыт,

- Контактные данные(телефон, емейл),
- О себе(можно записать какие-то пару предложений на выбор)

5) Задание на закрепление предыдущих тем: Создать журнал, что б можно было писать имя, фамилию и оценку
Поставить ограничения: имя и фамилия не выше 15 символов каждое.

И оценка не выше 5.

Если везде значения превышаться выдавать об этом
Сообщения в консоль.

6) Написать класс, в котором есть 4 функции: -пустая, которая просто выводит сообщение через print,
- которая принимает параметры и выводит их в консоль,
- которая принимает и возвращает параметры.
- которая принимает замыкание и распечатывает результаты в консоль

Урок 14. Конспект

- 1) Повторить то, что есть в видео.
- 2) Создать класс журнал с 1 массивом и 2 мя методами, удалить оценку и поставить оценку.
- 3) Создать в этом же классе метод расписание который в параметр принимает енам из 3 х кейсов утро, обед, вечер.

- 4) Внутри метода добавить свитч который смотрит на параметр в котором энам и говорит во сколько прийти в школу.
- 5) Почитать Усова и документацию на тему свойства. (Конспект)

Урок 15. Свойства

- 1) Повторите что было в уроке.
- 2) Зайти в документацию и Выпишите окло 5 примеров.
- 3) Создать класс в котором будет массив в котором – «мама» «папа» и «дети». Создать subscript для этого класса что б у класса был доступ как у массива по индексу.
- 4) Создать сабскрипт который будет принимать число умножать его на 100 и делить на 2.
- 5) Создать свой subscript у которого внутри есть оператор if else.
- 6) Создать класс Human и в нем 3 проперти имя , возраст пол.
- 7) Создать ещё 2 класса которые наследуются от класса Human и переопределить методы и свойства .
- 8) Создать дикшинари который хранит все типы данных которые вы знаете, а ключ должен быть только строка.
- 9) Почитать книгу Усова (Конспект)

Урок 16. ООП

- a).Найти информацию что такое наследование в Google, перефразировать эту информацию своими словами и вставить это в плейграунд в закомментированом виде.
- b).Сделать то же самое с понятием Инкапсуляция.

с).И так же с понятием полиморфизм.

- 1) Создать класс "люди", в этом классе 4 property - "имя", "рост", "вес", "пол", а также будет иметь метод "say(говорит)".
- 2) Создать subClass (наследники) "повар", "менеджер", "борец" и переопределить метод "say(говорит)" в каждом из этих классов.
- 3) Создать по одному объекту(экземпляру) каждого класса и объединить их в массив.
- 4) В цикле "for in" пройти по всем элементам массива и вывести в консоль все характеристики каждого объекта (имя, рост и тд) и у каждого вызвать метод "say(говорит)".
- 5) Создать еще одного наследника от super Class, добавить в него пару новых свойств, добавить в массив и в цикле вывода вывести его свойства как дополнение к свойствам People.
- 6) Метод "say(говорит)" реализовать таким образом, чтобы перед тем как выполнить собственный method say, выполнялся сначала метод say класса People.
- 7) Вывести все это в обратном порядке(Google в помощь).
- 8) Создать класс "Марсианин" (не наследник класса people!) со своими собственными property (отличными от people) и методом "say" (отличным от people).
- 9) Унаследоваться от него и создать пару других классов (Инопланетян) с переопределенным методом "say".
- 10) Объединить всех people и Марсианинов) в один массив.

Урок 17.

1. Повторите то, что есть в видео.

2 . Описать несколько структур – любой легковой автомобиль и любой грузовик.

2.1 Структуры должны содержать марку авто, год выпуска, объем багажника/кузова, запущен ли двигатель, открыты ли окна, заполненный объем багажника.

2.2 Описать перечисление с возможными действиями с автомобилем: запустить/заглушить, двигатель, открыть/закрыть окна, погрузить/выгрузить из кузова/багажника груз определенного объема.

2.3 Добавить в структуры метод с одним аргументом типа перечисления, который будет менять свойства структуры в зависимости от действия.

2.4 Инициализировать несколько экземпляров структур. Применить к ним различные действия.

2.5 Положить объекты структур в словарь как ключи! названия как строки, например:
`var dict = [structCar: "structCar"]`

3. Почитать о Capture List (см ссылку ниже) - и описать своими словами и сделать скрин своего примера и объяснения

[capture-lists-in-swift](#) <- [Ссылка](#)

4. Набрать код на белом скрине, понять в чем там проблема и решить эту проблему

5. У нас есть класс мужчины и его паспорт. Мужчина может родиться и не иметь паспорта, но паспорт выдается конкретному мужчине и не может выдаваться без указания

владельца. Чтобы разрешить эту проблему, ссылку на паспорт у мужчины сделаем обычной опциональной, а ссылку на владельца у паспорта – константой. Также добавим паспорту конструктор, чтобы сразу определить его владельца. Таким образом, человек сможет существовать без паспорта, сможет его поменять или выкинуть, но паспорт может быть создан только с конкретным владельцем и никогда не может его сменить. Повторить все, что на черном скрине и решить проблему соблюдая все правила!

6. Создать класс «Ученик».

Добавить свойство класса, в котором будет текущая дата.

Добавить метод класса, которое выводит текст этого задания в виде многострочного литерала.

7. Создать класс «FlowControl»

Добавить в класс 5 функций, каждая из которых будет использовать отдельный оператор управления потоком.

(В одной массив прогоняется через цикл `for`, в другой через `while`, в третьей через `repeat while`) и т.д.

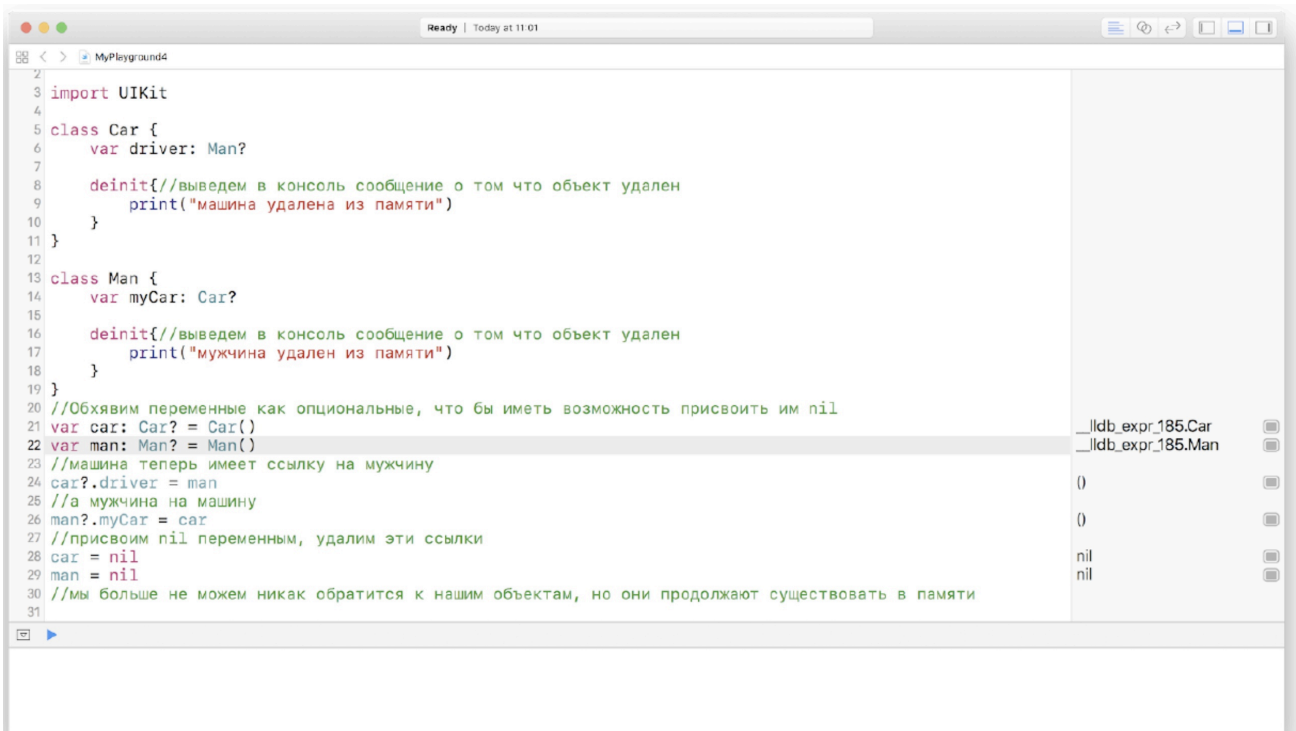
Каждая функция возвращает результат своей работы.

Создать `enum` с кейсами под каждую функцию.

Создать экземпляр класса и продемонстрировать работу каждой функции используя значения перечисления.

Функции напрямую управления потоком должны быть недоступными для вызова вне класса.

8) Читаем Усова, пишем конспект.



```
12 class Man {
13     var passport: Passport? // По заданию weak или unowned сюда запрещено
    ставить!
14
15     deinit {
16         // выведем в консоль сообщение о том, что объект удален
17         print("мужчина удален из памяти")
18     }
19 }
20
21 class Passport {
22     let man: Man
23
24     init(man: Man) {
25         self.man = man
26     }
27
28     deinit {
29         // выведем в консоль сообщение о том, что объект удален
30         print("паспорт удален из памяти")
31     }
32 }
33
34 var man: Man? = Man()
35 var passport: Passport? = Passport(man: man!)
36 man?.passport = passport
37 passport = nil // объект еще не удален, его удерживает мужчина
38 man = nil // теперь удалены оба объекта
39
```

Урок 18.

1) Прочитать документацию по уроку из:

[https:// developer.apple.com](https://developer.apple.com)

swiftbook.ru.

Выписать 5 примеров

2) Повторить то, что есть в видео.

По возможности используем в заданиях Optional chaining

3) Сделать класс Люди, у класса будут проперти родственники, соседи и т.д. (все опциональные).

3.1) Создать нужно от 15 человек.

3.2) Посчитать, сколько у этого человека двоюродных братьев, троюродных сестёр, тетя, дядя, и т.д.

4) Создать класс животные и проперти - корова, коза, собака и т.д.

Создать класс растения и проперти - трава, цветы и т.д.

5) Положить все классы массив и отсортировать по алфавиту и по классу - Люди - животные - растения

6) Создать 3 класса Rectangle, Circle, Radius

Класс Квадрат свойства:

circle: Circle?

Класс Radius свойства:

radiusName: String

Класс Circle свойства:

area = [Radius]()

6.1 Для класса Circle добавить вычисляемое свойство, выводящее количество элементов в area

6.2 Для класса Circle добавить сабскрипт

геттер возвращает запрашиваем элемент

сеттер добавляет новый элемент в area

6.3 Для класса Circle добавить функцию выводящую количество элементов в area

6.4 Создать метод класса, для каждого из 3 классов, который описывает для чего этот класс предназначен.

Создать экземпляры классов

Показать различные опциональные цепочки используя экземпляры классов, их методы и функции.

Перед выводом в консоль результатов избавляемся от опционала.

7. Читаем Усова, конспект.

Урок 19.

- 1) Постарайтесь по больше по практиковаться, повторите то, что есть в видео.
- 2) Зайти обязательно и познакомиться с документацией.
- 3) Я как заказчик даю вам разработчику задания сделать :

Библиотеку книг и видео библиотеку фильмов

Реализовать:

3.1 Возможность добавлять новые книги и фильмы и после добавления должна происходить автоматическая сортировка по алфавиту в пределах жанра.

3.2 Метод- "Sort by mood" - фильмы должны быть отсортировать по настроению.

3.3 Такой же метод придумать для книг.

3.4 Возможность удалять книги.

3.5 Выводить количество фильмов и книг в библиотеке

3.6 Выводить количество книг по жанрам

3.7 Выводить количество фильмов по жанрам

3.8 Выводить уведомление о местонахождении книги, фильма (на руках, в библиотеке)

3.9 У каждой книги и фильма должны быть запись о людях, бравших ее ранее с указанием имени, даты выдачи и даты возврата.

3.10 Метод, который выводит адрес библиотеки и контактные данные.

3.11 Предусмотреть механизм, запрещающий выдачу отсутствующих и забранных книг.

3.12 Одному человеку запрещается иметь на руках более 5 книг, и более 3 фильмов.

4. Читаем Усова, конспект.

Урок 20.

1) Повторить то, что есть в видео.

2) Зайти в документацию открыть тему extension и руками набрать пол этой главы, разобраться как это работает

<https://swiftbook.ru/content/languageguide/extensions/>

3) Расширить String, чтобы принимал сабскрипт вида `s[0..<3]` и мог также устанавливать значения используя его

4) Расширить Int и добавить 3 метода:

возведение в квадрат, в куб, в степень

5) Расширить еще 3 структуры (написать свои, либо взять на свой вкус и в комментариях обосновать для чего это было сделано

- 6) Добавьте проперти, которое возвращает количество символов в числе
- 7) Добавить метод, который говорит, что число положительное или отрицательное
- 8) Усов, конспект * ДЗ до 00:00 конспект не требуем

Урок 21.

- 1) Повторить то, что есть в видео.
- 2) Зайти в документацию открыть тему generic и руками набрать пол главы разобраться как это работает <http://swiftbook.ru/doc/swift-tour/generics>
- 3) Создать любой generic для классов, структур, методов, перечислений (по 1 на каждый).
- 4) Написать функцию, которая принимает Generic объект и складывает в массив/словарь (на выбор)
- 5) Создать класс, который сортирует массивы значений (на ваш выбор) 3 разными способами.
- 6) Написать класс на свой вкус (любые методы, проперти) универсального типа. Используя extension, расширить класс, добавить сабскрипт.
- 7) Усов, конспект * ДЗ до 00:00 конспект не требуем

Урок 22. Протоколы

- 1) Повторить код из урока.
- 2) Зайти и почитать документацию на тему protocol.
- 3) Посмотреть видео о copy on write (спрашивают на собесед!) - https://www.youtube.com/watch?v=66g_pD3s7TY.
- 4) Реализовать структуру IOSCollection и создать в ней copy on write как в уроке - <https://youtu.be/QsoqHGgX2rE>
- 5) Объяснить в коментах кратко что такое:
 - a) Copy on write.
 - б) isKnownUniquelyReferenced().
 - в) strong, weak, unowned ссылки.
- 6) Создать протокол переключатель с окончанием «able» как принято в swift сообществе.
- 7) В протокол переключатель добавить метод «переключать». (Все названия только на английском!).
- 8) Создать enum Switcher который конформит (conform) протокол переключатель и у которого 2 кейса off, on.
- 9) Создать объект с выбранным кейсом, далее вызвать у объекта метод переключения который меняет off на on например.
- 10) Создать протокол Movable с методом run, протокол Shootable с методом shoot.
- 11) Создать протокол HumanProtocol у которого есть свойство name и который конформит протоколы Movable и Shootable.

12) Создать класс Human подписаться на протокол HumanProtocol и реализовать его, в методы добавить принты.

13) Создать 2 класса и одну структуру от каждого по 1 экземпляру, и все это сложить в 1 массив. Внимание наследование и any и anyObject не использовать!

Урок 23. Делегаты

1) Изучить раздел "Протоколы -> Делегирование" в документации.

2) Проработать код из видео.

3) Создать протокол BatmanProtocol у которого свойство name.

4) Добавить дефолтную реализацию протоколу BatmanProtocol через экстеншен добавить свое имя.

5) Создать класс Batman подписаться на протокол BatmanProtocol, создать объект класса и распечатать name.

6) Создать протокол с 2мя методами 1 из них должен быть опциональным (optional).

7) Создать протокол с ассоциативным типом и создать 2 разных класса которые реализуют этот протокол. В 1м классе 1 тип в другом другой тип.

8) Создать протокол GoToShopDelegate с методом run.

9) Создать класс мама и класс сын. Использовать паттерн delegate что б мама отправляла сына за хлебом.