Srint 1

Команда po (print object) если поставить breakpoint может показать информацию по любому объекту

Лекция 2

Tuples – кортеж, let httpError = (404, “error 404”), обратиться можно: httpError.0 или .1

Type Aliases - typealias AudioSample = UInt16, просто присваиваем свое название типу для дальнейшего более удобного использования

Obj C – NSUInteger – только положительные, NSInteger – любой

CGFloat – число с плавающей точкой – примитивы

NSNumber – объект

NSString, NSMutableString - ?

В obj c строки пишутся @”hello word”

Isa - ?

Id - ?

NSArray, NSMutableArra, NSSet, NSMutableSet, NSDictionary, NSMutableDictionary - ?

strNonOptional = strOptional ?? "value" – присвоит строку если переменная nil

inout параметры функции. При объявлении функции перед типом переменной ставится слово inout, при передаче параметра в фукцию ставится &, таким образом значение будет изменено в основном коде.

Closures замыкания – кусок кода, который будет выполнен в том месте, где он вызван. Функция sorted(by:) получает функцию, которая сравнивает 2 объекта, и возвращает true если первый больше второго.

Лекция 3

- перед методом в obj с означает что это метод экземпляра, а + метод класса (статический).

Имя класса начинается с @interface

Свойства класса с @property – далее идут параметры () atomic/nanatomic – atomic гарантирует что в данный момент только одному потоку имеется доступ к данному свойству. Аттрибуты доступа readonly/readwrite – доступ только на чтение или чтение/запись. Свойство может быть ли свойство nil (nullable – может, nonull – никто из этого свойства не прочитает nil, nulresettable – аналог !. атрибут памяти copy – поддерживает только классы которые реализуют NSCopy, означает что когда объявляешь/отдаешь объект – то объявляешь/отдаешь копию. Weak – слабая ссылка. Слежующий параметр – class, значит свойство класса, а не экземпляра.

Объявление метода: +- (возвращаемый тип) Имя метода:(тип первого передаваемого аргумента)имя\_передаваемого\_аргумента

Если в каком-либо файле (к примеру AppDelegate) нужно использовать методы другого класса, .h этого класса необходимо импортировать в данный файл.

GRPHuman \*human – создание указателя на объект класса GRPHuman

GRPHuman \*human = [GRPHuman new] – создание объекта типа GRPHuman

[human sayHello] – Вызов метода экземпляра класса. Точно также вызываются методы класса, только надо писать название класса.

@synthesize – синтезирование внутреннего имени для переменной. name = \_name

Категории obj c – по сути те же расширения. -?

Делегаты - ?

Диспетчеризация - ?

Лекция 4

Чаще всего ошибки объявляют в enum

Чтобы сказать коду что возникла ошибка – throw {имя ошибки}

Перед кодом где может выкинуться ошибка пишем try.

Чтобы написать что функция может кинуть ошибку надо написать throws

Propogate – это когда ошибка уходит выше, throws

Do catch:

Do { try EXPRESSION} catch pattern1 Pstatements} …

Catch is name\_enum – если ошибка принадлежит этому enum – то обработать

Catch {} – перехват всех ошибок

Let x = try? someFunc() – получаем опционал, имеем ошибку или нет

Let photo try! Loadimage() – получаем точно значение, без опционала, использовать только тогда когда на 100% уверены что ошибки там не будет

Defer {} – данный блок кода будет выполнен когда мы выходим из функции, return или throw не важно.

LocalizedError – используется для хорошего описания ошибок, имеет: errorDescripton

Generics: Чтобы использовать после названия функции пишем <T>. T – любой тип

Лекция 5

@objc позволяет увидеть Swift классы в Obj-c, происходит автогенерация интерфейса в MyProject-Swift.h

Bridging Header – создается Xcode либо прописывается в свойствах проекта

Для использования Swift классов в Obj-C необходимо:

Создать Swift файл в Obc-C проекте в нем описать класс, перед ним написать @objc

подключить файл ProjectName-Swift.h в Obj-C файле. В данном файле в конце прописываются связи классов Swift с Obc-C.

Чтобы увидеть Obc-C класс из Swift надо в файле Bridging Header прописать import хедера класса obc-c.

Для переопределение имени Swift классы для Obj-c можно использовать @objc(MyObjCSwiftClass) – в скобках можно указать любое имя класса, чтобы имена Swift и Obc-c одного и того же класса были разные

**Классы в Swift по умолчанию Internal, поэтому не произойдет автогенерации в файле заголовка project\_name-Swift.h**

Mutating func – позволяет модифицировать свойство объекта, вносить изменения в данном экземпляре объекта, вместо того, чтобы возвращать новый экземпляр с измененным значениями!?

Sprint 2

MRC/ARC:

id – тип данный который подходил для всего

id object 2 = [object retain] – увеличивает счетчик ссылок объекта

[object release] – отказ от владения объектом, счетчик ссылок уменьшается

Autorelease & drain -?

Блоки:

@(1) – конвертация числа 1 в NSNumber (объектный тип)

В init() viewController’а нельзя обращаться к view, так как он на данном этапе еще не создан

Чтобы переименовать класс: Права кнопка мыши на названии класса – refactor – rename.

Чтобы удалить mainstoryboard – сначала удалить storyboard – в настройках прилоежния на вкладке info удалить строчку mainstoryboard.

Scroll View – предназначен чтобы водить телефон по view.

Create Code Snippet – можно создавать кусок кода, который будет создан при написании сокращения (указанного в completion)

Table View: наследуется от scrollView, поэтому можно использовать методы делегата scrollView

Intrinsic content size – то, сколько места вьюшке нужно для контента

“”” text “”” = чтобы текст взялся в переносами строк