

Лекция №3

Продолжение в Swift

Гибадулин Олег



образование

Организационная часть

- Отметиться - важно
- О чем пойдет речь в сегодняшнем занятии
- Оставить отзыв (после занятия)



Структура лекции

- Перечисления
- Классы
- ООП
- Свойства
- Методы
- Протоколы и расширения
- Структуры
- ARC
- Модификаторы доступа

Перечисления

Перечисление (**enumeration**) определяет общий тип для группы связанных значений

Для создания перечисления используется ключевое слово **enum**

Каждое отдельное значение в перечислении указывается после оператора **case**

Классы

Классы - это универсальные и гибкие конструкции, строительные блоки вашей программы.

Класс является **описанием** объекта, а объект представляет **экземпляр** этого класса

Для определения класса используется ключевое слово **class**

ООП

Методология программирования

- основана на представлении программы в виде совокупности **объектов**
- каждый из объектов является **экземпляром** определенного класса
- классы образуют **иерархию наследования**

Основные понятия: **наследование, инкапсуляция, полиморфизм**

Свойства

Свойства связывают значения с определённым классом, структурой или перечислением

Свойства хранения содержат значения константы или переменной как часть экземпляра

Вычисляемые свойства вычисляют значения, а не хранят их

Методы

Методы - это функции, которые связаны с определенным типом. Классы, структуры и перечисления - все они могут определять методы.

Методы экземпляра являются функциями, которые принадлежат экземплярам конкретного класса, структуры или перечисления. перечисления.

Методы типа принадлежат самому типу, а не экземплярам этого типа. Будет всего одна копия этих свойств вне зависимости от количества экземпляров.

Протоколы и расширения

Расширения добавляют новую функциональность существующему типу класса, структуры или перечисления

Протокол определяет образец методов, свойств. Протокол не предоставляет реализацию, он только описывает как реализация должна выглядеть. Реализация может быть прописана в классе, структуре или перечислении.

Структуры

Они почти как классы, но не совсем классы

- Структуры нельзя наследовать, но можно реализовать протокол
- Отсутствуют деинициализаторы
- Подсчет ссылок допускает более чем одну ссылку на экземпляр класса

Тип значения

Тип значения - это тип, значение которого **копируется**, когда оно присваивается константе или переменной, или когда передается функции.

Все **базовые типы** Swift - типы значений и реализованы они как структуры.

Все структуры и перечисления - типы значений. Это значит, что любой **экземпляр** структуры и перечисления, который вы создаете, и любые типы значений, которые они имеют в качестве **свойств**, всегда **копируются**, когда он передается по вашему коду.

Ссылочный тип

В отличие от типа значений, ссылочный тип **не копируется**, когда его присваивают переменной или константе, или когда его передают функции. Вместо копирования **используется ссылка** на существующий экземпляр.

Arc (automatic reference counting)

Автоматический подсчет ссылок - механизм в Swift для отслеживания и управления памятью вашего приложения

Уровни доступа

Swift предлагает пять различных уровней доступа для объектов вашего кода: **open**, **public**, **internal**, **fileprivate**, **private**

Эти уровни доступа относительно **объекту, исходному файлу**, в котором определен объект, и **модулю**, которому принадлежит исходный файл

Литература

<https://docs.swift.org/swift-book/>

<https://habr.com/ru/post/463125/>

Вопросы

Спасибо за внимание!



образование