Предварительная рецензия на лекцию 001_Основы языка Swift.

001_Основы языка Swift.

Материал изложен на доступном для новичка уровне.

В рамках лекционного материала «Основы языка Swift» можно привести краткую историческую справку о появлении языка Swift.

Хотелось бы видеть перед началом изложения материала – базовые системные тесты. В частности, запуск программы: «Привет, мир». Такой запуск мог бы продемонстрировать работоспособность среды разработки и ее окружения.

Техническая часть вопроса обсуждалась с автором курса. Было принято решение, что для большей части курса достаточно Swift онлайн. Также возможно исполнение базовых программ на Windows. Работа на родном для Swift железе и программном обеспечении – потенциально проблематична, по причине низкой распространенности. В первой лекции хотелось бы видеть три разных варианта запуска программы «Привет, мир». На данном моменте не настаиваю, но по опыту чтения литературы прошлых лет – программа «Привет, мир» может стать проблемой для новичков.

Вопрос: Swift - строки – ссылочный или значимый тип? Являются ли строки изменяемыми. Есть ли класс для быстрой работы строк? Какие плюсы и минусы…?

Поясню – почему этот вопрос кажется нужным и неочевидным на примере C#. System.String – значимый неизменяемый тип, то есть удаление одного символа приводит к полному копированию символов из старой строки в новую в оперативной памяти компьютера, что очень требовательно по времени. Пришло это из Ассемблера из аппаратных особенностей ЭВМ. Для System.String – методы обработки строк давно написаны и наследуются между поколениями из языков программирования. Есть stringbilder, который массив символов Char. Char – вроде бы значимый, но массив может быть, как значимым так и ссылочным. Поэтому для stringbilder – вопрос удаления

одного символа – это процедура удаления элемента массива, которая идет относительно быстро. Но для stringbilder –нет стандартных методов обработки строк. Их писать приходится по необходимости. Допускает ли формат лекций обсуждение этого вопроса для Swift?

Вопрос: в параграфе «Функции. Замыкания. Enum (перечисления)» можно ли по подробнее рассказать про лямбда-выражения и замыкания?

Часть текста на скриншоте возможно съехала на страницу вверх?

```
func greet(name: String) -> String {
    return "Привет, \(name)!"
}

let greeting = greet(name: "Алексей")
print(greeting)  // Привет, Алексей!
```

Есть функции, есть лямбда-выражения (которые могут быть переменными), есть замыкания (на контекст и на область видимости, и на область существования переменных вокруг лямбда-выражений и функций). Можно ли этот момент подробнее? Описание этой темы показалось несколько смазанным. (В Swift – терминология может отличаться).

С точки зрения некоторых современных языков функционального программирования (Julia) у функции может быть только одна переменная – это картеж. Поэтому его описание может быть к месту именно здесь…

Вопрос. Перечисления (Enum) – Есть ли возможность вернуть все перечисления списком или массивом? Есть ли возможность получить integer id значения перечисления и, наоборот, получение значения перечисления по integer id. Часть ответа уже приведена в виде ассоциированного значения.

```
enum Temperature {
    case hot(Double)
    case cold(Double)
}

let todayTemperature = Temperature.hot(30.0)
```

Вопрос по поводу классов и ООП… Деинициализация, деструктор или

финализатор? Если «деинициализация» – специфический для данного языка термин – то все в порядке.

В тексте увидел «управление памятью». Вопрос: есть ли сборщик мусора? Есть ли указатели? Есть ли выделение памяти системными или библиотечными вызовами? Есть ли в языке понятие сегмента памяти? Возможны ли утечки памяти?

Вопрос: есть ли в языке модификаторы доступа к параметрам, методам и классам, как в C++? Или как в Python – все параметры публичны?

С уважением к Вам

Сибирев Иван Валерьевич.