

## Введение

### 001\_Основы языка Swift

Вопрос. Уместно ли приводить краткую историческую справку о появлении языка Swift в рамках лекционного материала “ Основы языка Swift ”?

Материал изложен на доступном для новичка уровне.

Хотелось бы видеть перед началом изложения материала – базовые системные тесты. А именно запуск программы “Привет мир”. Такой запуск мог бы продемонстрировать работоспособность среду

Техническая часть вопроса обсуждался 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 очень хотелось бы иметь развилку на три разных 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... Почему вопрос нужный и неочевидный... На примере C#. System.String – значимый не изменяемый, 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 прошлого века из аппаратных особенностей ЭВМ. Для System.String – методы обработки строк давно написаны и наследуются между поколениями из языков программирования. Есть stringBuilder, который массив символов Char. Char – вроде бы как значимый, но массив может быть, как значимым так и ссылочным. По этому для stringBuilder – вопрос удаления одного символа это процедура удаления элемента массива, которая относительно быстрая. Но для stringBuilder – почти не стандартных методов обработки строк. Их писать приходится по необходимости. Допускает ли формат лекций обозрения этого момента для Swift?

Вопрос... В параграфе “Функции. Замыкания. Enum (перечисления). Можно ли по подробнее рассказать про лямбда выражения и замыкания.

Часть – возможно съехала на страницу в верх...

```
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 по подробнее. Мне почему-то показалось описание этой темы 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. По этому 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 память. 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 питоне - все параметры публичны?

## Базовые типы данных в Swift