



Основы Swift

Классы и структуры

Введение.

Что такое классы и структуры

- Классы и Структуры — это основные элементы для создания вашей программы, расширяются за счет методов и свойств.
- В отличие от Objective-C в **Swift** файл интерфейса не отделяется от файла имплементации.

Что общего у классов и структур

Для классов и структур можно

- определить свойства для хранения значений
- определить методы для обеспечения функциональности
- определить сабскрипты для обеспечения доступа к значениям
- определить инициализаторы для задания начальных значений
- определить операторы
- расширять функционал с помощью extensions
- поддерживать протоколы

Что могут классы, но не могут структуры

Классы могут

- наследоваться от класса для получения базового функционала
- использовать деструкторы для освобождения памяти
- использовать несколько ссылок на объект, благодаря системе подсчета ссылок (структуры при передаче копируются, а не используют ссылки)

Определение классов и структур

Определение классов и структур схоже.
Используются ключевые слова
`class` и `struct`.

```
class SomeClass {  
    // определение класса  
}
```

```
struct SomeStructure {  
    // определение структуры  
}
```

При создании определяется новый тип.
Именовывать принято используя UpperCamelCase
для соответствия стандартному Swift
написанию.

Пример определения КиС

```
struct Resolution {  
    var width = 0  
    var height = 0  
}
```

```
class VideoMode {  
    var resolution = Resolution()  
    var interlaced = false  
    func someFunc() {  
        print("Do Nothing")  
    }  
}
```

Создание экземпляров

Определение класса или структуры описывает общие признаки, но не сам объект. Для описания объекта необходимо создать экземпляр.

```
let someResolution = Resolution()  
let someVideoMode = VideoMode()
```

В общем случае для этого используются () следующие за именем класса или структуры. В этом случае создается экземпляр со значениями заданными по умолчанию. Можно задать свои.

Доступ к свойствам экземпляра

Используется (.) для доступа к свойствам.
Такой синтаксис называется **Dot Notation**
или **Dot Syntax**.

Получение свойства структуры
`someResolution.width`

Получение свойства структуры, являющейся
свойством класса
`someVideoMode.resolution.width`

Запись нового значения для свойства
`someVideoMode.resolution.width = 1280`

Инициализатор по умолчанию для структур

Для всех структур можно использовать инициализатор перечисляющий все свойства входящие в состав структуры.

```
struct Resolution {  
    var width = 0  
    var height = 0  
}
```

```
let vga = Resolution(width: 640, height: 480)
```

Классы не получают подобный инициализатор по умолчанию.