

Prática em Swift



Professor Pedro Henrique
prof.pedrohenrique.iossdk@gmail.com

Conceitos importantes do dia



- Palavra reservada guard e sua função
- Palavra reservada defer e sua função
- Inicialização de Structs
- Mutação de Structs
- Diferenças Fundamentais entre Classe e Struct

Links do ❤️



- <http://nshipster.com/guard-and-defer/>
- <https://www.codebeaulieu.com/40/How-to-use-Guard-in-Swift-2-to-parse-JSON>
- <https://www.natashatherobot.com/swift-guard-better-than-if/>
- <http://radex.io/swift/guard/>
- <http://www.codingexplorer.com/structures-swift/>

guard



- O que é?
 - Um novo operador condicional (desde Swift 2.0) que exige que a execução saia do escopo se a condição não for atendida.
- Para que serve?
 - O principal uso do guard é no tratamento de erros com maior clareza, majoritariamente quando se está criando um método construtor.
- Como usar?

```
guard let _ = string.rangeOfString("^\\d+$", options: [.RegularExpressionSearch])  
    else { throw ConversionError.InvalidFormat }
```


defer



- O que é?
 - Uma nova forma de fazer com que um bloco de código seja executado somente quando o fluxo de execução sair do escopo atual.
- Para que serve?
 - Serve para evitar tratamentos repetitivos e, principalmente, além de ajudar a evitar memory-leaks.
- Como usar?

```
let destData = UnsafeMutablePointer<UInt8>.alloc(dataSize)
defer {
    destData.dealloc(dataSize)
}
```

Inicialização de Structs



- Ao contrário de outras linguagens, no Swift as Structs podem ter um método construtor (init)
- A inicialização de uma Struct é escrita da mesma forma que se faz com uma classe. Vejamos:

```
1 struct XYPoint {
2     var x: Int
3     var y: Int
4 }
init(x: Int, y: Int) {
    self.x = x
    self.y = y
}
```

Mutação de Structs



```
1 struct Rectangle {  
2     var width = 1  
3     var height = 1  
4 }
```


Mutação de Structs



```
1 struct Rectangle
2     var width = 10
3     var height = 5
4
5     func area() -> Int {
6         return width * height
7     }
8 }
```

A função [area] não faz qualquer modificação nos dados da struct

Mutação de Structs



A função [`scaleBy`], por outro lado, **modifica** o valor interno da struct.

```
1 struct Rectangle {  
2     var width = 1  
3     var height = 1  
4  
5     mutating func scaleBy(value: Int) {  
6         width *= value  
7         height *= value  
8     }  
9 }
```

Mutação de Structs



- Observe a palavra reservada mutating
- Lembre-se! **Uma struct é um tipo de dado.**
 - Em outras palavras, uma struct, normalmente é um tipo imutável, sendo declarado com [let], portanto.
 - Para usar a mutação de structs, a variável deve ser declarada como [var].

Classe X Struct



Classe	Struct
Pode herdar de outras classes	Não pode ter herança
O objeto pode passar por typecast	Não pode passar por typecast
Pode declarar conformidade com protocolos (e ter a declaração atestada)	Pode declarar conformidade com protocolos (e ter a declaração atestada)
Pode ter um método desinicializador (dealloc)	Não pode ter método desinicializador
Pode ter múltiplas referências para a mesma instância	Não pode ter múltiplas referências para a mesma instância

ONE DOES NOT SIMPLY

SKIP PRACTICE

memes.com

