

Classes IOS – Swift

Parte 2

X-Code

Prof. Agesandro Scarpioni

Classes – Métodos

- No Swift, criar métodos com muitos parâmetros e passar estes parâmetros para os métodos é bem mais fácil que Obj-C. Vamos ver 4 exemplos sem o return, ou seja, sub rotina :

- Java

- `public void CalcularImcComPesoeAltura(float peso, float altura) { código aqui };`

- VB

- `Public Sub CalcularImcComPesoeAltura(Byval peso as Single, Byval altura as Single) (código aqui) End Sub`

- Objective-C

- `- (void) calcularImcComPeso:(float) peso eAltura:(float) altura { código aqui }`

- Swift

- `func calcularImc(peso:Float, altura:Float) { código aqui }`

Classes – Métodos

- Em nosso exemplo de classes no Swift o método tem o nome `calcularImc`, veja que o nome foi adaptado para receber dois parâmetros, e não há indicação de retorno pois se trata de uma sub rotina, mesmo com a palavra `func` no início do método. Veja a chamada abaixo:
- `a calcularImc(peso: 100 Altura: 1.85)`

Classes – Métodos

- Vamos implementar uma sub rotina e uma function na classe Atleta.swift.

```
26
27 //Sub-rotina (não tem retorno) - funciona como o Void em várias linguagens.
28 func calcularIMC(peso:Float, altura:Float){
29     let imc = peso / (altura * altura)
30     print("O IMC de \(self.nome) é \(imc)")
31 }
32
33 //Função (tem retorno) - funciona como a Function em várias linguagens.
34 func calcularIMC_2(peso:Float, altura:Float)->Float{
35     // return peso / (altura * altura)
36     //ou
37     return peso / pow(altura,2)
38 }
39
```

DICAS: Caso prefira use pow para trabalhar com potência → `imc = peso / pow(altura,2);`

Classes – Métodos

- No Main vamos chamar os 2 métodos criados, calcularIMC (sub rotina) e calcularIMC_2 (function)

```
1 //
2 //  main.swift
3 //  ClasseExemplo
4 //
5 //  Created by Agesandro Scarpioni on 12/02/17.
6 //  Copyright © 2017 Agesandro Scarpioni. All rights reserved.
7 //
8
9 import Foundation
10
11 var a = Atleta()
12
13 a.nome = "Ana"
14 a.idade = 22
15
16 print("NOME: \(a.nome)")
17 print("IDADE: \(a.idade)")
18
19 print("Iron Man \(a.nome) \(a.idade) anos")
20
21 var a2 = Atleta(nome: "Carlos Gomes", idade: 25)
22 print("Iron Man \(a2.nome) \(a2.idade) anos")
23
24 //chamando uma Sub-rotina que exibe um texto
25 a2.calcularIMC(peso: 100, altura: 1.85)
26
27 //imprimindo os dados de uma função que retorna um float
28 print(a.calcularIMC_2(peso: 80, altura: 1.75))
29
```

NOME: Ana
IDADE: 22
Iron Man Ana 22 anos
Iron Man Carlos Gomes 25 anos
O IMC de Carlos Gomes é 29.2184
26.1224
Program ended with exit code: 0

Métodos de Instância

- Foi criado nos slides anteriores dois métodos de instância, criaremos mais um, só que desta vez será sem uma função sem parâmetro que retorna um texto. Vá na classe Atleta.swift e digite o método abaixo:

```
39  
40 //métodos de instancia, no Obj-C se usa o símbolo (-) no início do nome do método  
41 func descricao() ->String{  
42     return "O Atleta " + self.nome  
43 }  
44
```

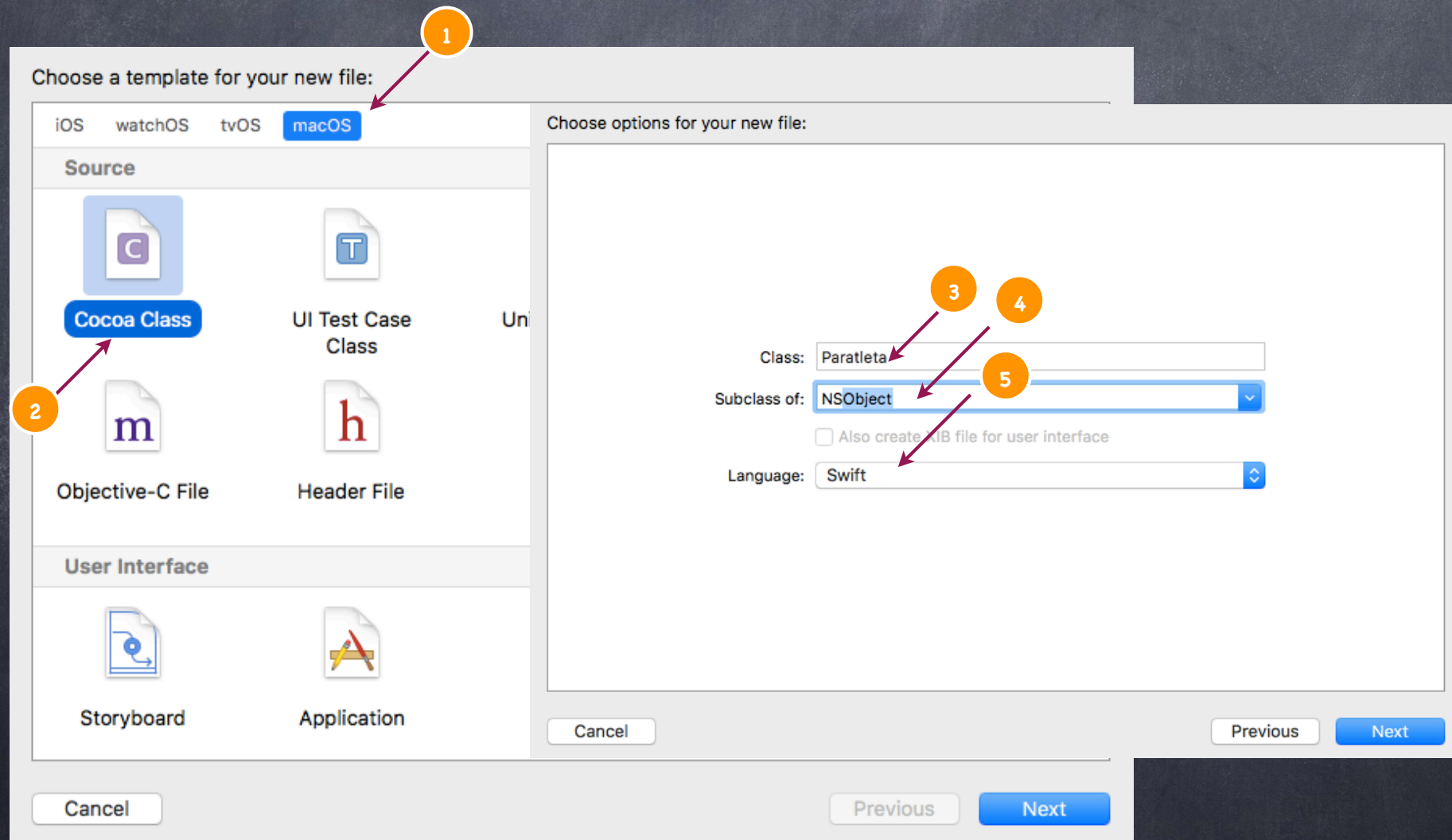

Métodos de Classe

- O método de Classe funciona como o método estático do Java, porém ao invés de iniciar com a palavra func da página anterior o nome do método é iniciado com a palavra class.

```
44  
45 //métodos de classe, no Obj-C se usa o símbolo (+) no início do nome do método  
46 //O método de classe é como o método estático no Java  
47 class func alerta()->String{  
48     return "Se beber não dirija"  
49 }  
50
```


Métodos Estáticos

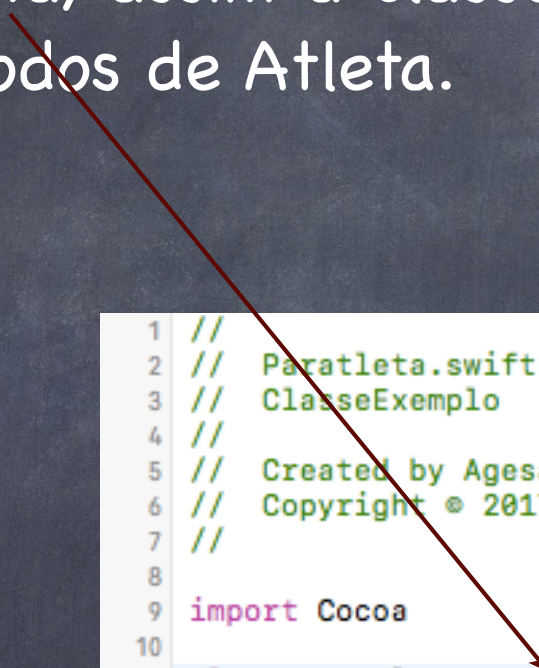
- Criar a classe Paratleta subclasse de NSObject. Clique em File → New File → macOS → double clique em Cocoa Class, ou, Command + N.



Métodos Estáticos

- Na classe Paratleta, altere a subclasse NSObject para Atleta, assim a classe Paratleta irá herdar os atributos e métodos de Atleta.

```
1 //  
2 // Paratleta.swift  
3 // ClasseExemplo  
4 //  
5 // Created by Agesandro Scarpioni on 12/02/17.  
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 import Cocoa  
10  
11 class Paratleta: Atleta {  
12  
13 }  
14
```



Métodos Estáticos

- Crie o atributo "modalidade" em Paratleta e já crie um construtor padrão, claro que esse atributo é um atributo que caberia perfeitamente em Atleta e não somente em Paratleta.

```
1 //  
2 // Paratleta.swift  
3 // ClasseExemplo  
4 //  
5 // Created by Agesandro Scarpioni on 12/02/17.  
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 import Cocoa  
10  
11 class Paratleta: Atleta {  
12  
13     var modalidade: String  
14  
15  
16     override init(){  
17         modalidade=""  
18         super.init()  
19     }  
20 }  
21
```


Métodos Estáticos

- Crie um método descricao na classe Paratleta que retorne uma string, use override pois esse método será sobreposto ao método descricao da classe Atleta.

```
1 //  
2 // Paratleta.swift  
3 // Teste de Classes  
4 //  
5 // Created by agesandro scarpioni on 02/05/15.  
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 import Foundation  
10  
11 class Paratleta:Atleta {  
12     var modalidade: String  
13  
14  
15     override init(){  
16         modalidade=""  
17         super.init()  
18     }  
19  
20  
21     override func descricao() -> String {  
22 |  
23     }  
24  
25 }  
26
```


Métodos Estáticos

- Na implementação da linha 22, observe que estamos invocando o método descricao da classe Atleta, por isso usamos super, e para isso não criamos um objeto para invocar o método, isso porque em Atleta o método descricao é um método estático, ou seja, iniciou com a palavra "class"

```
1 //  
2 // Paratleta.swift  
3 // ClasseExemplo  
4 //  
5 // Created by Agesandro Scarpioni on 12/02/17.  
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 import Cocoa  
10  
11 class Paratleta: Atleta {  
12  
13     var modalidade: String  
14  
15  
16     override init(){  
17         modalidade=""  
18         super.init()  
19     }  
20  
21     override func descricao() -> String {  
22         return super.descricao() + " da \((modalidade) é paratleta "  
23     }  
24 }  
25
```


Métodos Estáticos

- No main (1), crie um novo objeto para Paratleta (2), faça a carga de algumas informações nos atributos (3) e chame o método estático descricao(4) da classe Paratleta.

```
19 print("Iron Man \(a.nome) \(a.idade) anos")
20
21 var a2 = Atleta(nome: "Carlos Gomes", idade: 25)
22 print("Iron Man \(a2.nome) \(a2.idade) anos")
23
24 //chamando uma Sub-rotina que exibe um texto
25 a2.calcularIMC(peso: 100, altura: 1.85)
26
27 //imprimindo os dados de uma função que retorna um float
28 print(a.calcularIMC_2(peso: 80, altura: 1.75))
29
30 let p = Paratleta()
31 p.nome = "Jonas"
32 p.idade = 30
33 p.modalidade = "natação"
34 print(p.descricao())
35
```

NOME: Ana
IDADE: 22
Iron Man Ana 22 anos
Iron Man Carlos Gomes 25 anos
O IMC de Carlos Gomes é 29.2184
26.1224
O Atleta Jonas da natação é paratleta
Program ended with exit code: 0

Prática

Criação de um programa para testarmos todos os conceitos deste tópico.

- Ainda no projeto da classe Enfermeira criado anteriormente, desenvolva um método do tipo void e três métodos que retornem respectivamente um bool , um String e um Int.
- Como dica, vocês podem criar um método que entre com a temperatura do corpo e retorne um booleano para indicar que o paciente está com febre, ou ainda um método que receba dois parâmetros e retorne a temperatura da água para um banho em graus celsius. Faça uso do método construtor com pelo menos 3 atributos.