

# Classes IOS – ObjC

Parte 3 – Exceptions e Protocolos

X-Code

Prof. Agesandro Scarpioni

[agesandro@fiap.com](mailto:agesandro@fiap.com)



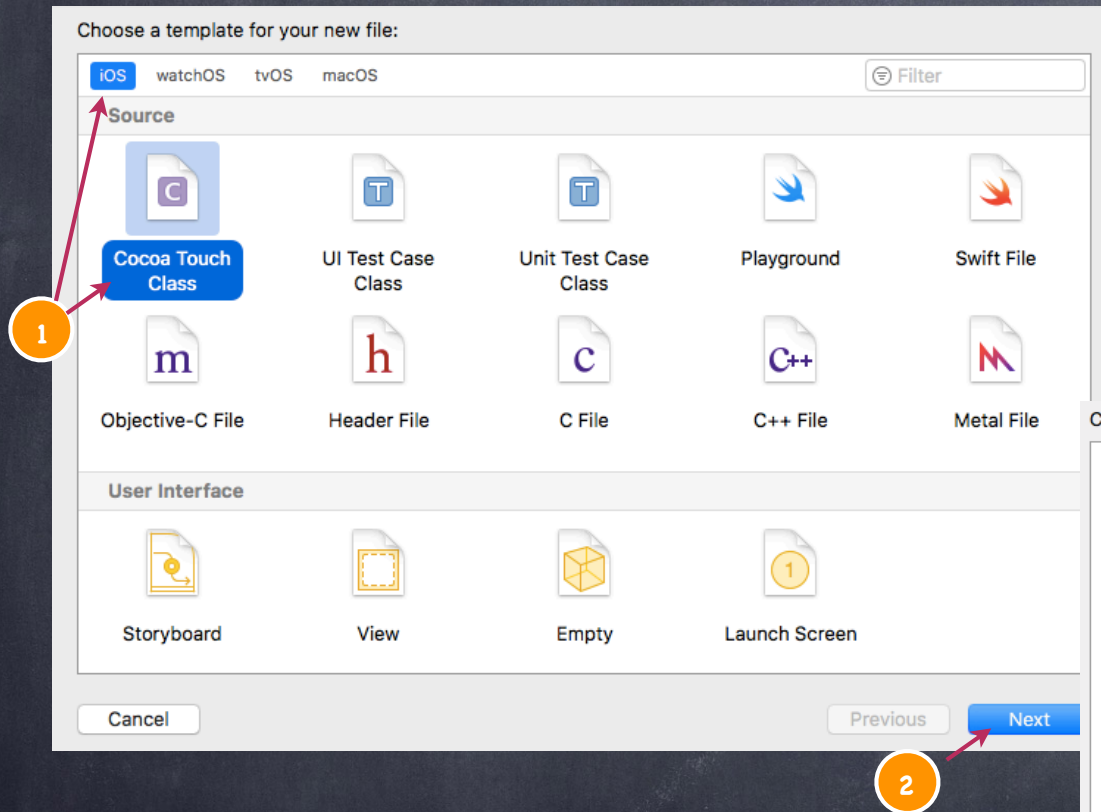
# Exceptions

- A forma de criar uma exception em Obj-C é muito semelhante a forma de criar excessões em outras linguagens. Abra o programa da classe Atleta.
- Inclua mais uma Classe chamada ImcException filha de NSErrorException.

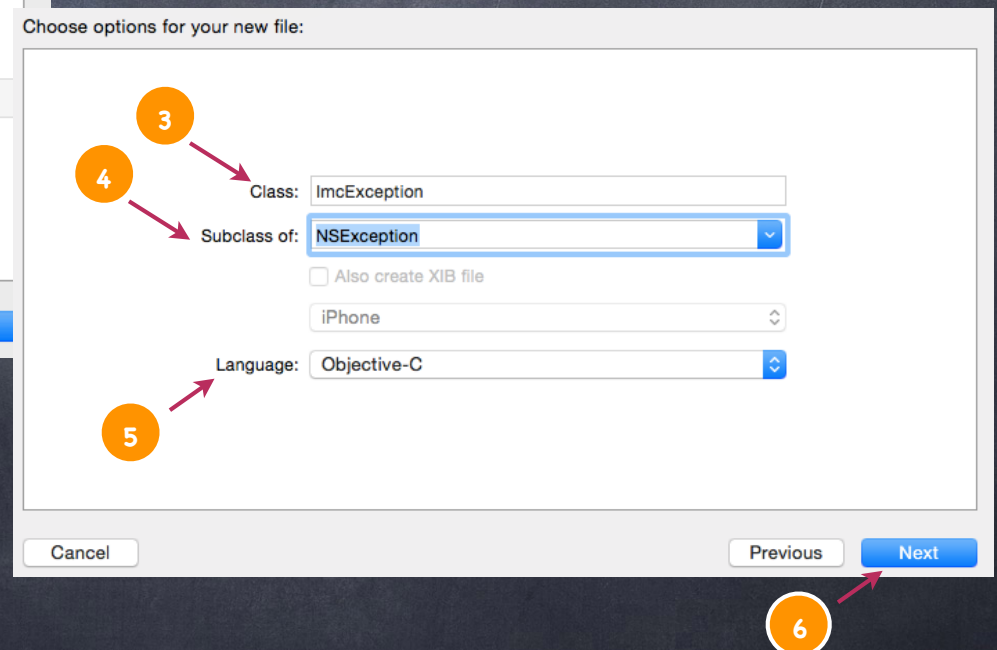


# Exceptions

- Crie uma classe "ImcException". Clique em File->New->File-> IOS -> Cocoa Touch Class.



- Nomeie a classe como ImcException subclasse de NSError





# Exceptions

- Na classe Atleta.m faça o import da classe ImcException.h

```
1  //
2  //  Atleta.m
3  //  ClasseExemplo
4  //
5  //  Created by agesandro scarpioni on 08/03/15.
6  //  Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7  //
8
9  #import "Atleta.h"
10 #import "ImcException.h"
11
12 @implementation Atleta
13
14 -(void) setName: (NSString *)_nome{
15     nome=_nome;
16 }
17 -(NSString *) getName{
18     return nome;
19 }
```



# Exceptions

- Ainda na classe Atleta.m faça o lançamento da exception dentro do método CalcularImcComPesoeAltura.

```
27 -(void) calcularImcComPeso:(float) peso eAltura:(float) altura{
28     float imc;
29     if (altura > 2){
30         //throw exception
31         NSString *motivo=@"Altura não pode ser maior que 2 metros";
32         NSError *e = [imcException exceptionWithName:@"imcException" reason:motivo userInfo:nil];
33         @throw e;
34     }
35     imc = peso / (altura * altura);
36     //ou use a função para potência chamada pow, veja abaixo:
37     //imc = peso / pow(altura,2);
38     NSLog(@"O imc de %@ é %0.2f", self.getNome, imc);
39 }
```



# Exceptions

- O método `exceptionWithName` serve para criar uma exception e iniciá-la com um nome e um reason, o reason será uma mensagem que explica o erro.
- O parâmetro `userInfo` é um `NSDictionary` do Foundation que representa uma estrutura de chave e código e pode conter mais informações sobre o erro.

```
27 -(void) calcularImcComPeso:(float) peso eAltura:(float) altura{
28     float imc;
29     if (altura > 2){
30         //throw exception
31         NSString *motivo=@"Altura não pode ser maior que 2 metros";
32         NSException *e = [imcException exceptionWithName:@"imcException" reason:motivo userInfo:nil];
33         @throw e;
34     }
35     imc = peso / (altura * altura);
36     //ou use a função para potência chamada pow, veja abaixo:
37     //imc = peso / pow(altura,2);
38     NSLog(@"O imc de %@ é %.2f", self.getNome, imc);
39 }
```



# Exceptions

- Na classe main.m faça o import da classe ImcException.h

```
1 //  
2 //  main.m  
3 //  ClasseExemplo  
4 //  
5 //  Created by Agesandro Scarpioni on 05/02/17.  
6 //  Copyright © 2017 Agesandro Scarpioni. All rights reserved.  
7 //  
8  
9 #import <Foundation/Foundation.h>  
10 #import "Atleta.h"  
11 #import "MassagemAtleta.h"  
12 #import "ImcException.h"
```



1



# Exceptions

- Ainda na classe main.m chame um método com uma altura não permitida, por exemplo: 2.20m.

```
23
24     Atleta *a2 = [[Atleta alloc] initWithNome:@"Maria Mendes" andIdade:18];
25     NSLog(@"Iron Man %@ %d anos", [a2 getNome], [a2 getIdade]);
26     [a2 calcularImcComPeso:83.8 eAltura:1.87];
27     NSLog(@"%@", [a2 calcularSeuRendimentonaAguawithTempoemHoras:1.5 andMetros:8000]);
28
29
30     @try {
31         [a2 calcularImcComPeso:83.8 eAltura:2.20];
32     }
33     @catch (ImcException *exception) {
34         NSLog(@"Erro: %@", [exception reason]);
35     }
36     @finally {
37
38     }
39
```



# Exceptions

- Utilizando o bloco try/catch/finally, o método é chamado com uma altura inválida e uma exceção será lançada.

```
@try {  
    [a2 calcularImcComPeso:83.8 eAltura:2.20];  
}  
@catch (ImcException *exception) {  
    NSLog(@"Erro: %@", [exception reason]);  
}  
@finally {  
}
```



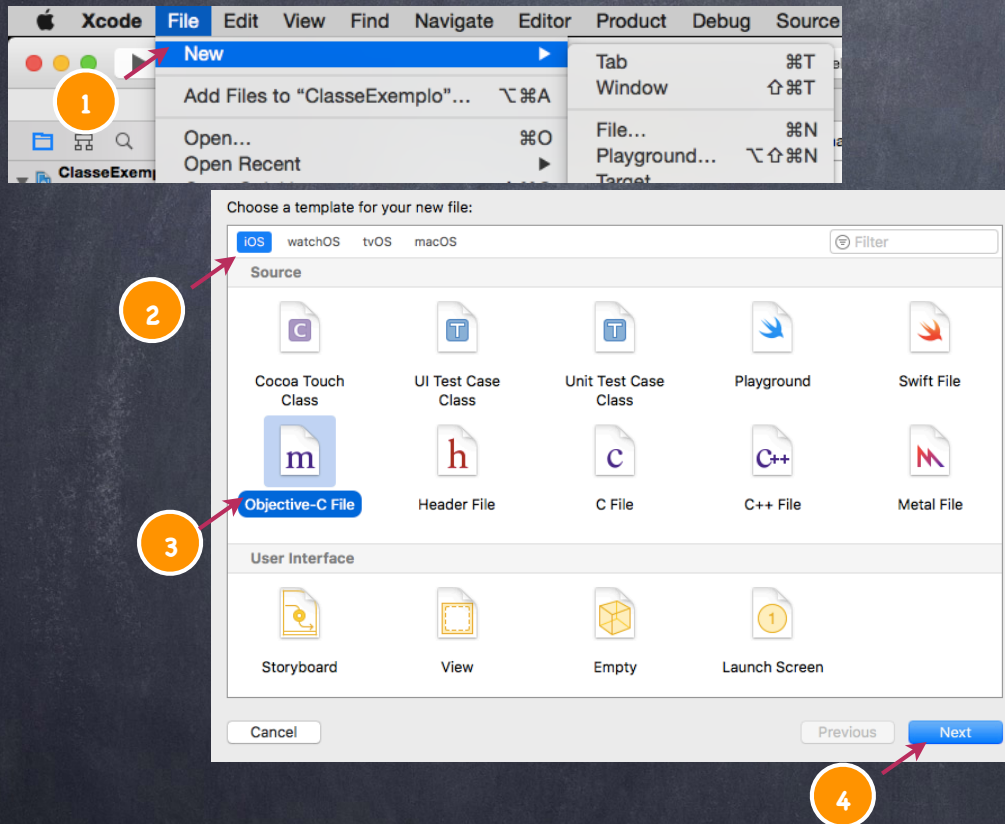
# Protocolos

- O protocolo funciona como as interfaces do Java ou do VB ou aos métodos virtuais do C++, é declarado com a notação @protocol e uma classe pode implementar um ou mais protocolos, desde que quando tivermos mais que um protocolo, a declaração entre um protocolo e outro deve ser separado por vírgula, exemplo: <protocolo1,protocolo2, protocolo3>.
- O protocolo necessita apenas do arquivo .h, crie 3 protocolos para a alimentação do atleta.

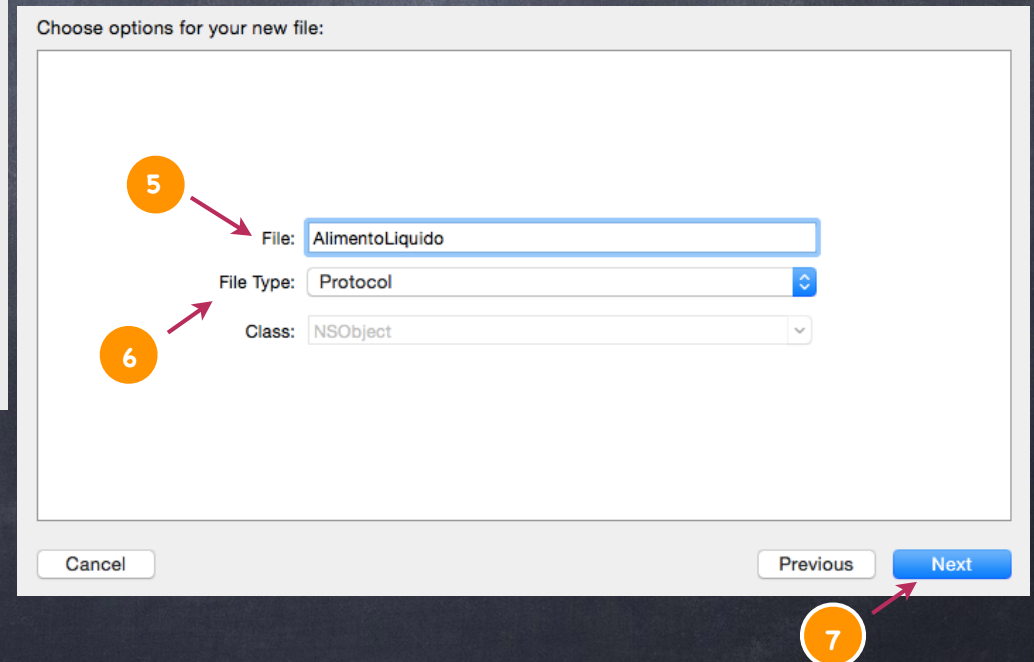


# Protocolos

- Crie 3 protocolos para alimentação do atleta, AlimentoLiquido, AlimentoSolido, AlimentoLiquidoeSolido, este último vai herdar os outros dois protocolos. Clique em File->New->File->IOS->Source->Objective-C File.



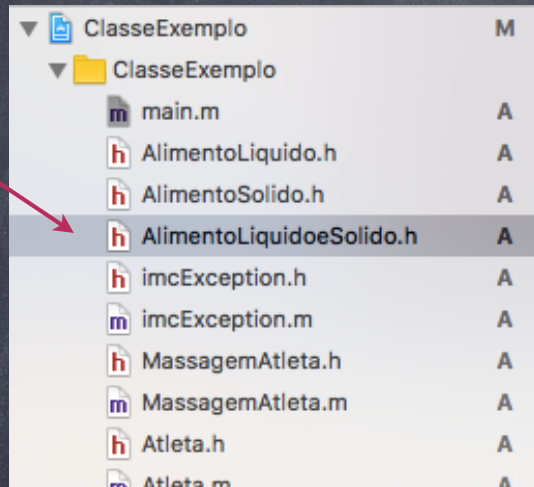
- Nomeie o primeiro como AlimentoLiquido escolha Protocol e repita o mesmo passo para os outros.





# Protocolos

- Com os 3 protocolos criados como mostra na imagem 1, declare os métodos destacados nos pontos 2 e 3.



```
1 //
2 // AlimentoLiquido.h
3 // ClasseExemplo
4 //
5 // Created by agesandro scarpioni on 08/03/15.
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10
11 @protocol AlimentoLiquido <NSObject>
12
13 -(void) beberIsotonico;
14
15 @end
```

```
1 //
2 // AlimentoSolido.h
3 // ClasseExemplo
4 //
5 // Created by agesandro scarpioni on 08/03/15.
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10
11 @protocol AlimentoSolido <NSObject>
12
13 -(void) comerCarboidrato;
14
15 @end
```

```
1 //
2 // AlimentoLiquidoeSolido.h
3 // ClasseExemplo
4 //
5 // Created by agesandro scarpioni on 08/03/15.
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10 #import "AlimentoLiquido.h"
11 #import "AlimentoSolido.h"
12
13 @protocol AlimentoLiquidoeSolido <AlimentoLiquido, AlimentoSolido>
14
15
16 @end
```

**Obs:** Não esqueça dos dois imports no ponto 4. O protocolo AlimentoLiquidoeSolido herda os métodos dos 2 protocolos, veja que no ponto 5 para implementar mais de um protocolo em uma classe separamos os nomes por virgula < \_\_ , \_\_ > .



# Protocolos

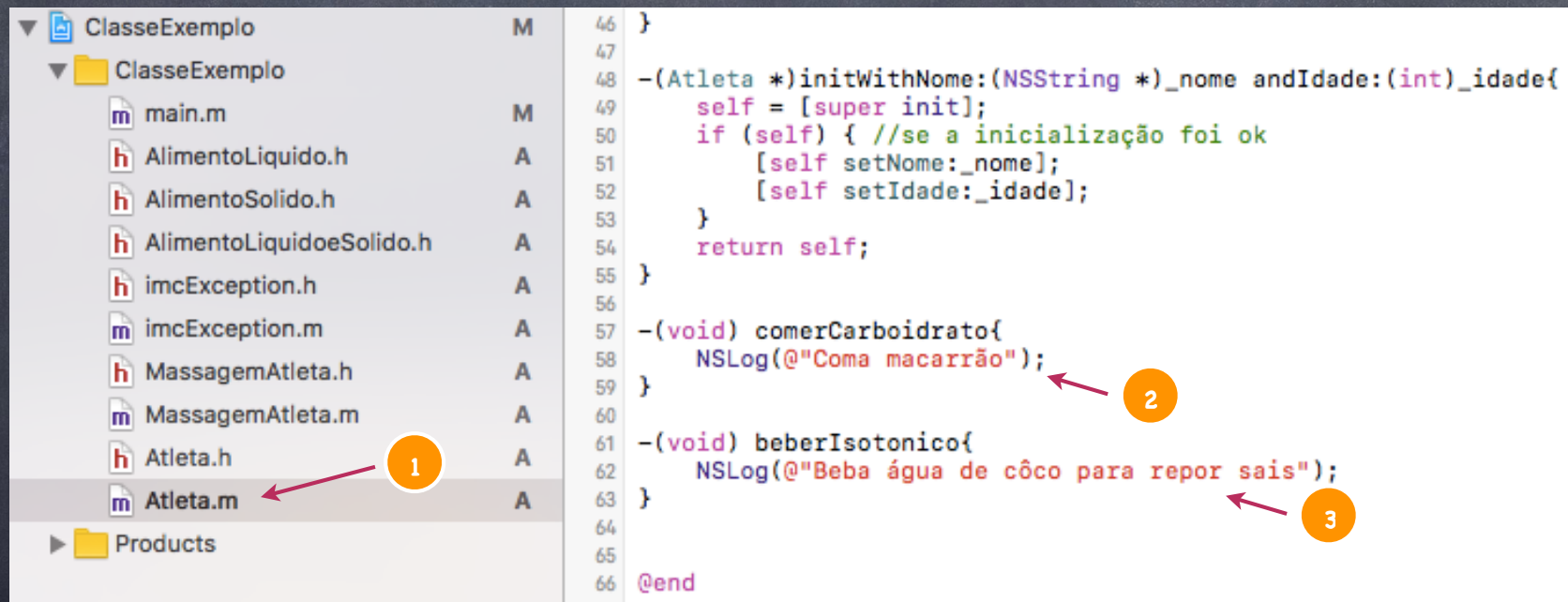
- Para ser mais rápido vamos apenas implementar o protocolo: AlimentoLiquidoeSolido, vá para a classe Atleta.h. faça o import (1) e declare a interface AlimentoLiquidoeSolido (2).

```
1 //
2 // Atleta.h
3 // ClasseExemplo
4 //
5 // Created by agesandro scarpioni on 08/03/15.
6 // Copyright (c) 2015 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10 #import "AlimentoLiquidoeSolido.h"
11
12 @interface Atleta : NSObject <AlimentoLiquidoeSolido> {
13     NSString *nome;
14     int idade;
15 }
16
17
18 -(void) setName: (NSString *)_nome;
19 -(NSString *) getName;
20 -(void) setIdade: (int) _idade;
21 -(int) getIdade;
22
23 -(void) calcularImcComPeso:(float) peso eAltura:(float) altura;
```



# Protocolos

- Na classe Atleta.m vamos implementar os dois métodos do protocolo AlimentoLiquido e Solido.





# Protocolos

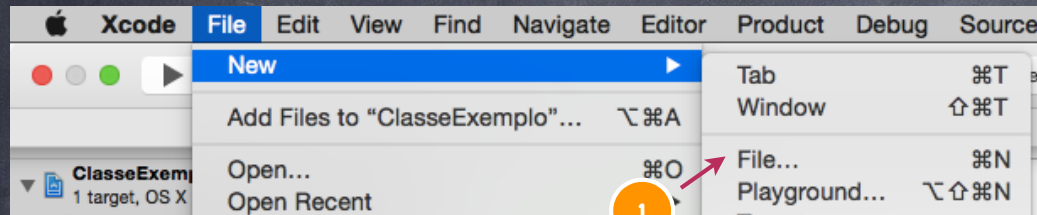
- Na main.m teste o protocolo chamando um dos métodos.

```
26 [a2 calcularImcComPeso:83.9 eAltura:1.87];
27 NSLog(@"%@", [a2 calcularSeuRendimentonaAguawithTempoemHoras:1.5 andMetros:8000]);
28
29
30 @try {
31     [a2 calcularImcComPeso:83.9 eAltura:2.20];
32 } @catch (NSEException *exception) {
33     NSLog(@"Erro: %@", [exception reason]);
34 } @finally {
35
36 }
37
38 [a2 beberIsotonico];
39
40 MassagemAtleta *m = [[MassagemAtleta alloc] init];
41 [m setAtleta:a]; //este método foi gerado automaticamente pelas notações @property e @synthesize
42 [m massagearAtleta];
43
44 return 0;
45 }
```

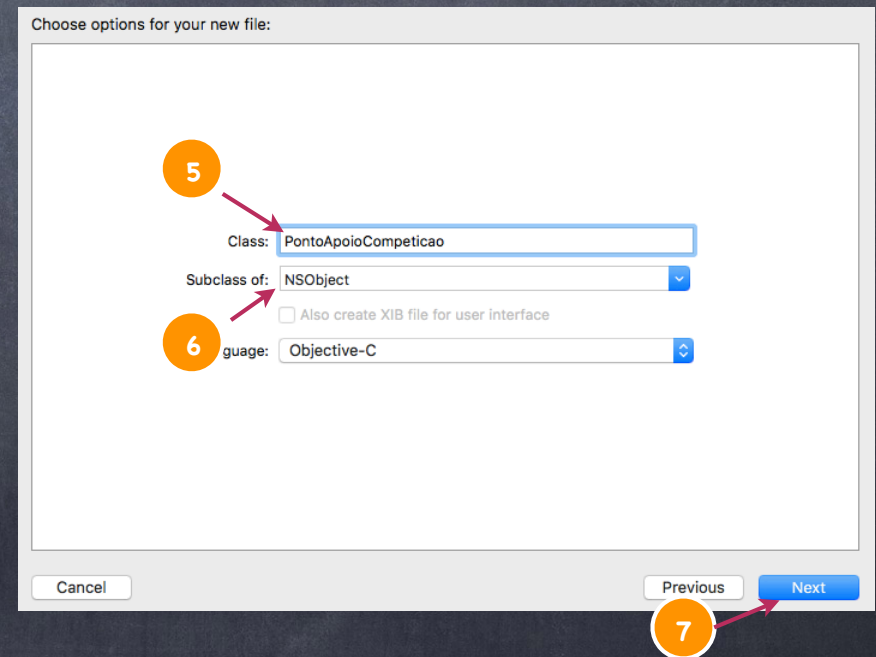
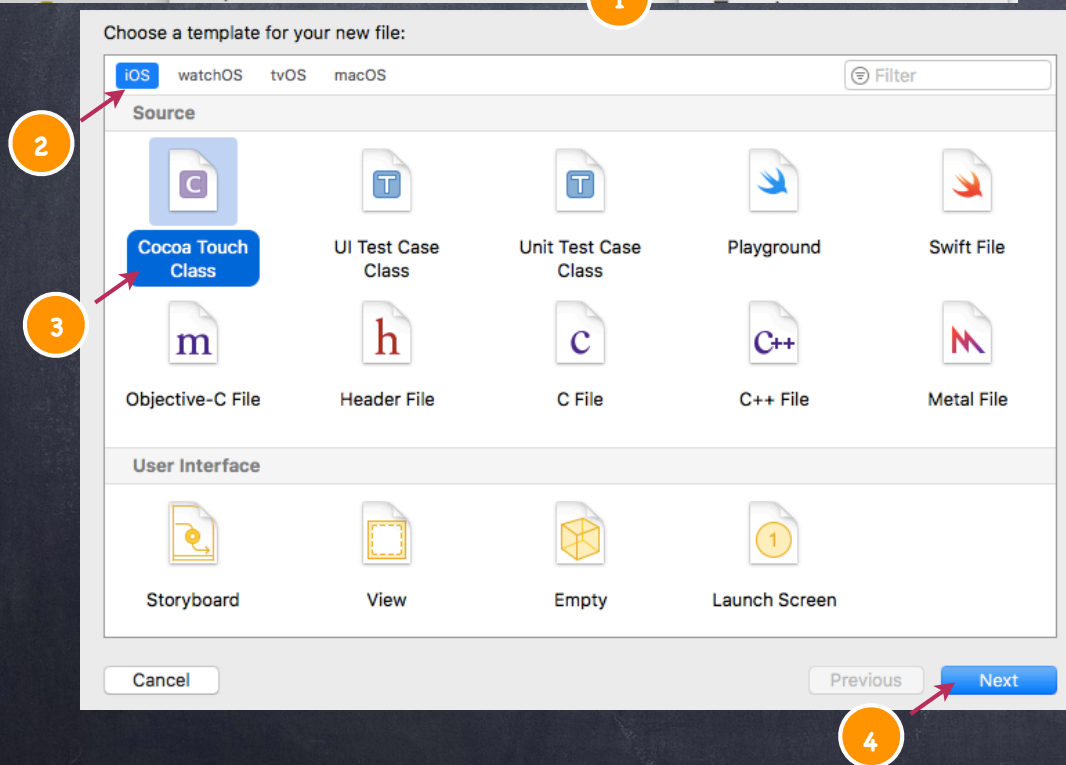


# Classes + POO

- Crie uma classe "PontoApoioCompeticao". Clique em File->New->File-> iOS -> Source -> Cocoa Touch Class -> Next.



- Nomeie a classe PontoApoioCompeticao como subclasse de NSObject



**Dica:** Esta classe será responsável por entregar um alimento do tipo líquido ou sólido para o atleta.



# Classes + POO

- O símbolo + indica que são métodos da classe, ou seja, métodos estáticos.
- Na classe PontoApoioCompeticao.h importe os dois protocolos (1) e declare os dois métodos estáticos (2).

```
1 //
2 // PontoApoioCompeticao.h
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/Foundation.h>
10 #import "AlimentoLiquido.h"
11 #import "AlimentoSolido.h"
12
13 @interface PontoApoioCompeticao : NSObject
14
15 +(void) entregarParaAtletaLiquidos:(NSObject <AlimentoLiquido> *) tipoLiquido;
16 +(void) entregarParaAtletaSolidos:(NSObject <AlimentoSolido> *) tipoSolido;
17
18 @end
19
```



# Classes + POO

- Na classe `PontoApoioCompeticao.m` implemente os dois métodos estáticos. Abaixo, a forma de declarar um parâmetro do tipo protocolo `NSObject <protocolo> *`

```
8
9  #import "PontoApoioCompeticao.h"
10
11 1 @implementation PontoApoioCompeticao
12
13  + (void) entregarParaAtletaLiquidos:(NSObject <AlimentoLiquido> *) tipoLiquido{
14      // delega para o método do protocolo
15      [tipoLiquido beberIsotonicos];
16  }
17
18  + (void) entregarParaAtletaSolidos:(NSObject <AlimentoSolido> *) tipoSolido{
19      // delega para o método do protocolo
20      [tipoSolido comerCarboidratos];
21  }
22  @end
23
```



# Classes + POO

- A classe chamada PontoApoioCompeticao vai receber atletas com necessidades de alimentação dos tipos líquidos (isotônicos, vitaminas, proteínas, etc) ou sólidos (barras de cereais, barras de proteínas, carboidratos, etc) ou até mesmo dos dois tipos, lembre-se que apenas implementamos isotônico e carboidrato.
- A implementação dos métodos de PontoApoioCompeticao recebe objetos do tipo AlimentoSolido ou AlimentoLiquido (que são os protocolos) e apenas delega a chamada para os métodos comerCarboidrato e bebeIsotonico.



# Classes + POO

- No main.m importe a classe pontoApoioCompeticao (1), e chame o método estático entregarParaAtletaSolidos(2).

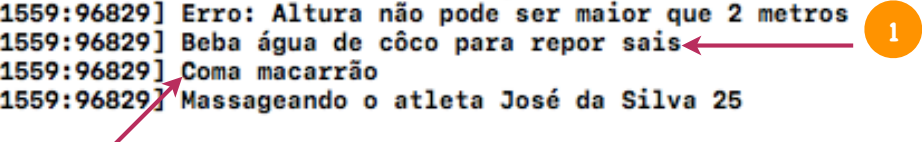
```
1 //
2 // main.m
3 // ClasseExemplo
4 //
5 // Created by Agesandro Scarpioni on 05/02/17.
6 // Copyright © 2017 Agesandro Scarpioni. All rights reserved.
7 //
8
9 #import <Foundation/Foundation.h>
10 #import "Atleta.h"
11 #import "MensagemAtleta.h"
12 #import "ImcException.h"
13 #import "PontoApoioCompeticao.h" ← 1
14
15 int main(int argc, const char * argv[]) {
16     @autoreleasepool {
17
18         Atleta *a = [[Atleta alloc] init];
19         [a setName:@"José da Silva"];
20         [a setIdade:25];
21         NSLog(@"Iron Man %@ %d anos", [a getName], [a getIdade]);
22         [a calcularImcComPeso:83.9 eAltura:1.87];
23         NSLog(@"%@", [a calcularSeuRendimentonaAguawithTempoemHoras:1.5 andMetros:8000]);
24
25         Atleta *a2 = [[Atleta alloc] initWithName:@"Maria Mendes" andIdade:18];
26         NSLog(@"Iron Man %@ %d anos", [a2 getName], [a2 getIdade]);
27         [a2 calcularImcComPeso:83.9 eAltura:1.87];
28         NSLog(@"%@", [a2 calcularSeuRendimentonaAguawithTempoemHoras:1.5 andMetros:8000]);
29
30
31         @try {
32             [a2 calcularImcComPeso:83.9 eAltura:2.20];
33         } @catch (NSEException *exception) {
34             NSLog(@"Erro: %@", [exception reason]);
35         } @finally {
36
37         }
38
39         [a2 beberIsotonico];
40
41         [PontoApoioCompeticao entregarParaAtletaSolidos:a2]; ← 2
42
43         MensagemAtleta *m = [[MensagemAtleta alloc] init];
44         [m setAtleta:a]; //este método foi gerado automaticamente pelas notações @property e @synthesize
45         [m messageAtleta];
46     }
47     return 0;
48 }
```



# Classes + POO

- Veja o resultado dos dois últimos métodos.

```
2017-02-12 11:47:50.848868 ClasseExemplo[1559:96829] Iron Man José da Silva 25 anos
2017-02-12 11:47:50.849120 ClasseExemplo[1559:96829] O imc de José da Silva é 23.99
2017-02-12 11:47:50.849219 ClasseExemplo[1559:96829] O meu rendimento na água é 5333.33 metros por hora
2017-02-12 11:47:50.849240 ClasseExemplo[1559:96829] Iron Man Maria Mendes 18 anos
2017-02-12 11:47:50.849254 ClasseExemplo[1559:96829] O imc de Maria Mendes é 23.99
2017-02-12 11:47:50.849278 ClasseExemplo[1559:96829] O meu rendimento na água é 5333.33 metros por hora
2017-02-12 11:47:50.849940 ClasseExemplo[1559:96829] Erro: Altura não pode ser maior que 2 metros
2017-02-12 11:47:50.849982 ClasseExemplo[1559:96829] Beba água de côco para repor sais
2017-02-12 11:47:50.850008 ClasseExemplo[1559:96829] Coma macarrão
2017-02-12 11:47:50.850034 ClasseExemplo[1559:96829] Massageando o atleta José da Silva 25
Program ended with exit code: 0
```





# Classes + POO

- Exercícios práticos com MVC no próximo conjunto de slides.