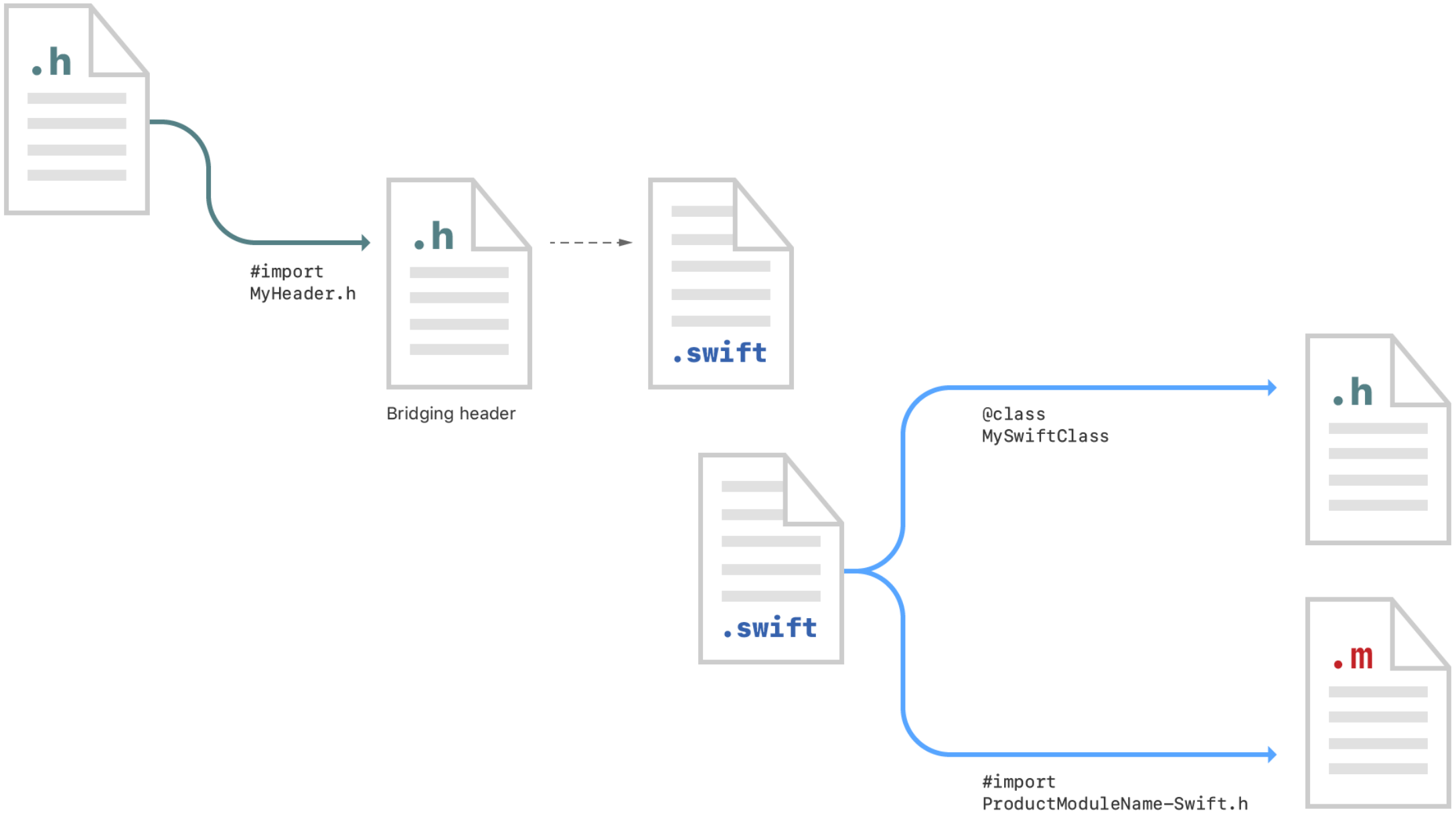
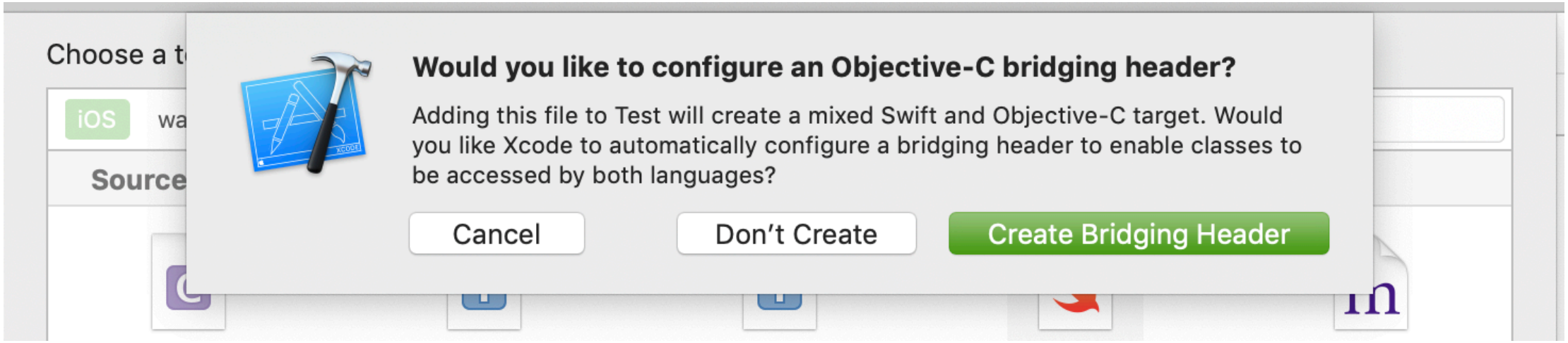
Swift and Objective-C


.h
#import
MyHeader.h
.h
Bridging header
.swift
.swift
@class
MySwiftClass
#import
ProductModuleName-Swift.h
.h
.m

Bridging-header file



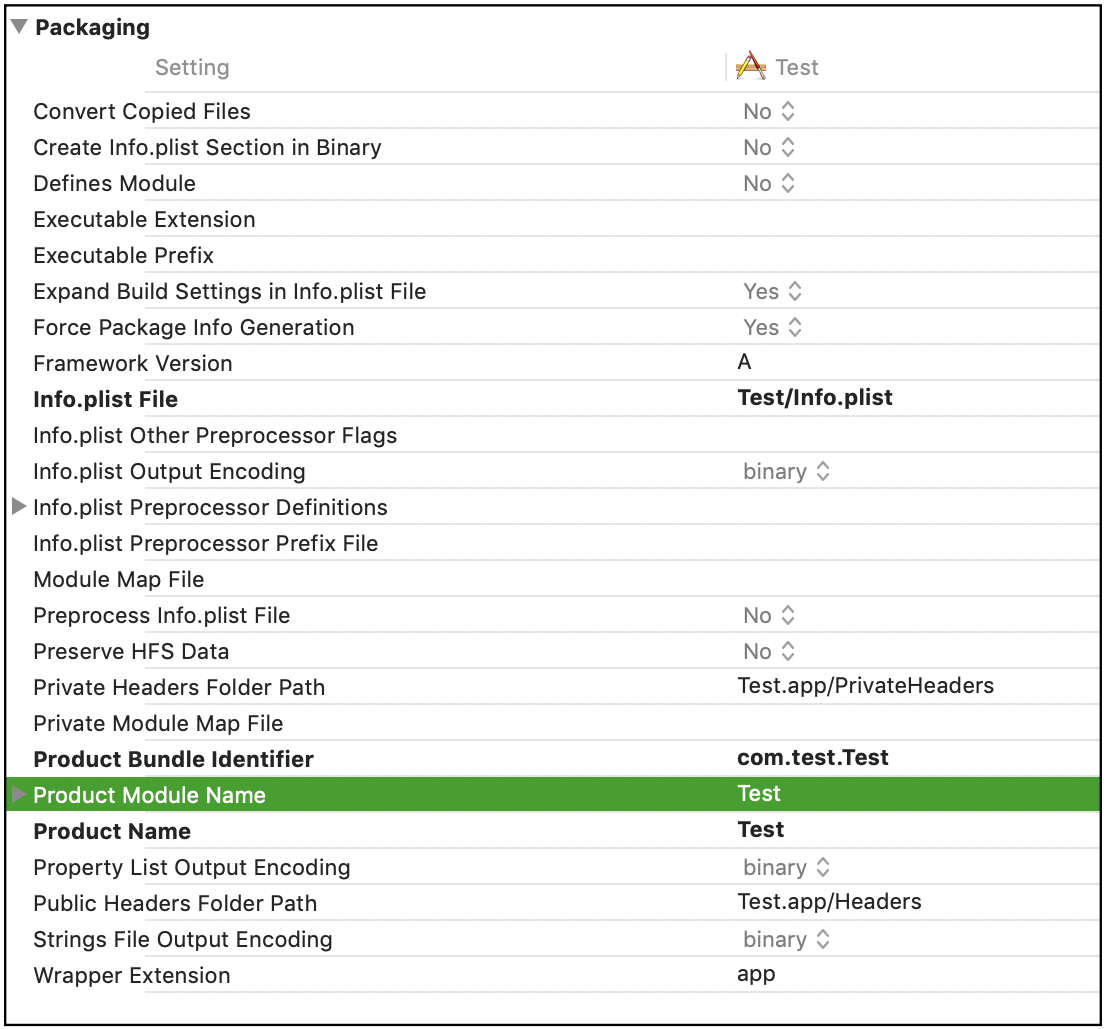

```
1  //
2  //  Use this file to import your target's public headers that you would like to expose to Swift.
3  //
4
5
```

Objective-C to Swift

```objc
//
//  Use this file to import your target's public headers that you would like to expose to Swift.
//

#import "AppDelegate.h"
#import "ViewController.h"
```

```swift
//
//  MyClass.swift
//  Test
//
//  Created by Aleksandr Sychev on 10/03/2019.
//  Copyright © 2019 Aleksandr Sychev. All rights reserved.
//

import Foundation

class Swifty : NSObject {
    let viewController = ViewController()
}
```

Swift to Objective-C



Swift to Objective-C

```objc
//
//  ViewController.m
//  Test
//
//  Created by Aleksandr Sychev on 10/03/2019.
//  Copyright © 2019 Aleksandr Sychev. All rights reserved.
//

#import "ViewController.h"

#import "Test-Swift.h"

@interface ViewController ()

- (Swifty *)mySwiftObject;

@end
```

Swift to Objective-C

```objc
//
//  AppDelegate.h
//  Test
//
//  Created by Aleksandr Sychev on 10/03/2019.
//  Copyright © 2019 Aleksandr Sychev. All rights reserved.
//

#import <UIKit/UIKit.h>

@class Swifty;

@interface AppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) UIWindow *window;

@end
```

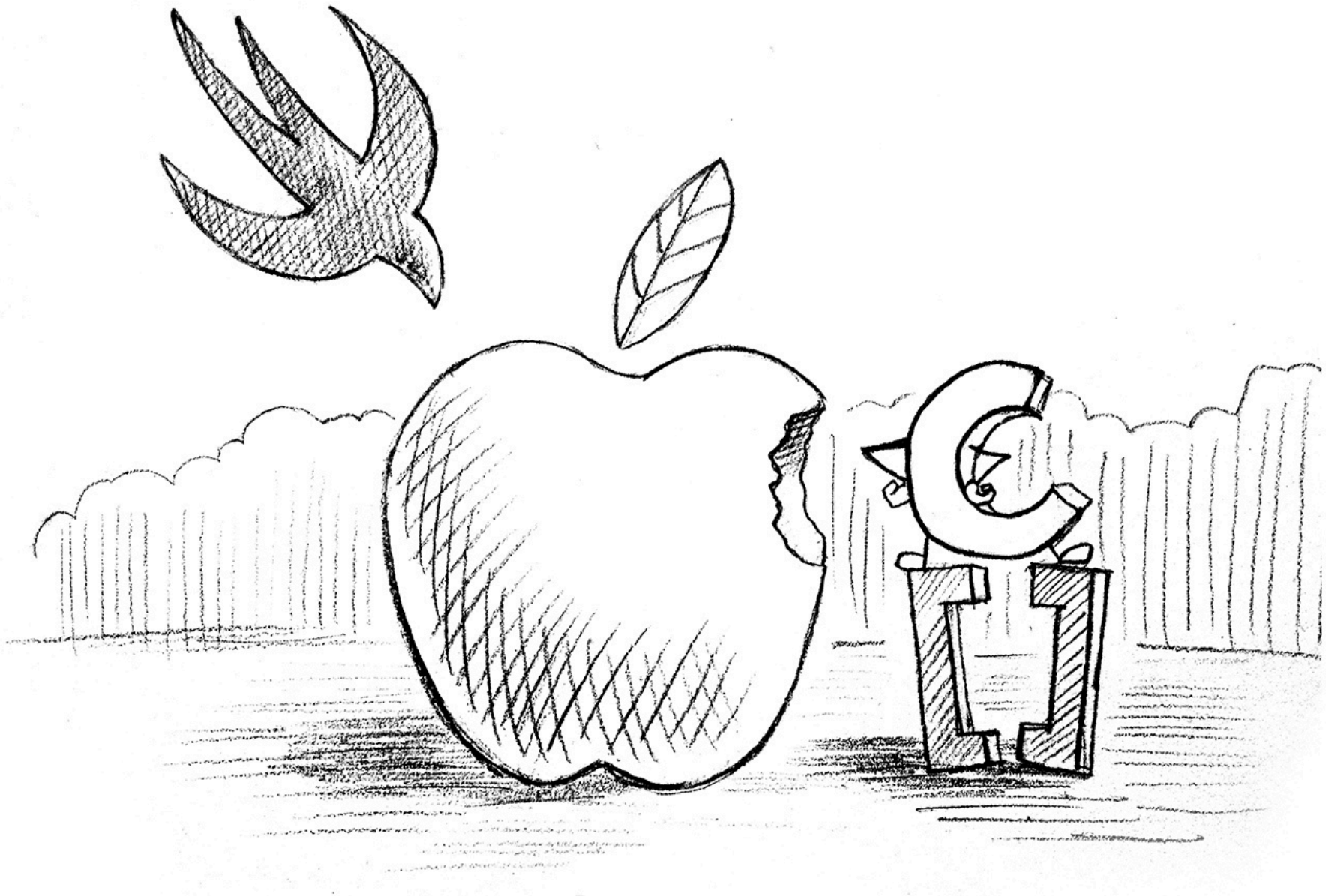
Swift to Objective-C

> *A Swift class must be a descendant of an Objective-C class to be accessible and usable in Objective-C.*

Swift to Objective-C

```
185  SWIFT_CLASS("_TtC4Test19SwiftyForObjectiveC")
186  @interface SwiftyForObjectiveC : NSObject
187  - (NSInteger)sum SWIFT_WARN_UNUSED_RESULT;
188  - (nonnull instancetype)init OBJC_DESIGNATED_INITIALIZER;
189  @end
```

Пример



Objective-C Types in Swift

Objective-C Type	Swift Equivalent
`BOOL`	`Bool`
`NSInteger`	`Int`
`SEL`	`Selector`
`id`	`AnyObject!`
`Class`	`AnyClass!`
`NSString *`	`String!`
`NSArray *`	`AnyObject[]!`

Objective-C Nullability

```objc
@interface MyList : NSObject
- (MyListItem *)itemWithName:(NSString *)name;
- (NSString *)nameForItem:(MyListItem *)item;
@property (copy) NSArray<MyListItem *> *allItems;
@end
```

```swift
class MyList: NSObject {
    func item(withName name: String!) -> MyListItem!
    func name(for item: MyListItem!) -> String!
    var allItems: [MyListItem]!
}
```

Objective-C Nullability

- **Nonnullable** — as nonoptionals
- **Nullable** — as optionals
- Without a nullability or with a **null_resettable** — as implicitly unwrapped optionals

Objective-C Nullability

```objc
@interface MyList : NSObject
- (nullable MyListItem *)itemWithName:(nonnull NSString *)name;
- (nullable NSString *)nameForItem:(nonnull MyListItem *)item;
@property (copy, nonnull) NSArray<MyListItem *> *allItems;
@end
```

```swift
class MyList: NSObject {
    func item(withName name: String) -> MyListItem?
    func name(for item: MyListItem) -> String?
    var allItems: [MyListItem]
}
```

Objective-C Nullability

_Nullable id * _Nonnul

Objective-C Nullability

```
NS_ASSUME_NONNULL_BEGIN

@interface MyList : NSObject
- (nullable MyListItem *)itemWithName:(NSString *)name;
- (nullable NSString *)nameForItem:(MyListItem *)item;
@property (copy) NSArray<MyListItem *> *allItems;
@end

NS_ASSUME_NONNULL_END
```

Read

- https://medium.com/ios-os-x-development/swift-and-objective-c-interoperability-2add8e6d6887
- https://developer.apple.com/swift/blog/?id=25
- https://developer.apple.com/documentation/swift/objective-c_and_c_code_customization
- https://developer.apple.com/documentation/swift/imported_c_and_objective-c_apis
- https://www.hackingwithswift.com/articles/114/objective-c-to-swift-conversion-cheat-sheet
- https://youtu.be/WL9i7tBf3vk