

現場のための Swift4 Swift4.1+Xcode9.3 対応 <正誤表>

●146 ページ 「ライフサイクルメソッドで呼ばれるメソッド」1 行目

【誤】 viewDidLoad: インスタンス化された直後

【正】 viewDidLoad: **ViewController が自身の View をメモリ上にロードした直後**

●157 ページ 本文 1~4 行目

【誤】

これは「print(array[4])」の箇所エラーが発生します。理由は array が格納している数に対して、それ以上の箇所へのアクセスを行おうとしているためです。あるはずのない箇所=nil へのアクセスを行おうとしているため、クラッシュしてしまいます。

【正】

これは「print(array[4])」の箇所エラーが発生します。理由は array が格納している数に対して、それ以上の箇所へのアクセスを**行った**ためです。**配列のインデックスの範囲外にアクセスしようとした場合、実行時にクラッシュして**しまいます。

●179~180 ページ 本文 8 行目以降、コード「役割は関数だが、変数として利用したい場合の一例」の最後まで

【誤】

また、クロージャーの一番の使いどころは、関数としての役割を持ちながらも変数として扱いたい場合です。

▼役割は関数だが、変数として利用したい場合の一例[sample-6-8-5.playground]

……ここからコード……

```
struct MyStruct {
    var valueOne: Int = 50
    var valueTwo: Int = 55

    // 2 つの変数の和が 100 を超えているか判定
    var isOverTotalHundred: Bool {
        return (valueOne + valueTwo) > 100
    }
}

let myStruct = MyStruct()
if myStruct.isOverTotalHundred {
    print(myStruct.valueOne)
} else {
    print(myStruct.valueTwo)
}
// 50(myStruct.valueOne の値を表示)
……ここまでコード……
```

本来であれば関数として定義しますが、変数として関数の振る舞いを行います。また、変数には値を保持する必要はありません。関数を意識せず変数のように扱えることがポイントです。

【正】

上記内容を削除

理由:

該当箇所はクロージャーについて解説している内容を記載していましたが、サンプルコードの使用法ならびに、それに付随する解説はクロージャーではなく「計算型プロパティ(computed property)」の使用例、解説となります。深くお詫び申し上げます。

計算型プロパティは一見変数のように見えますが、中身はメソッドのように特定の処理を行うものです。クラス、構造体、列挙型で定義することが可能です。また、計算型プロパティは特定の値を保持することができないという特徴もあります。

●214 ページ 「イベント通知の身近な例」2 行目

【誤】 **NSNotificationCenter**

【正】 NotificationCenter

※Swift 3.0 より Foundation Framework に含まれる一部のクラス、プロトコルから接頭辞の NS が削除されました。Swift では NotificationCenter を使用します。

●215 ページ 1 行目

【誤】 `NotificationCenter`

【正】 `NotificationCenter`

●215 ページ 図 7-2-1 の表題及び図の内部

【誤】 `NotificationCenter`

【正】 `NotificationCenter`

※Swift 3.0 より Foundation Framework に含まれる一部のクラス、プロトコルから接頭辞の `NS` が削除されました。Swift では `NotificationCenter` を使用します。

●223 ページ ▼ `SwitchCell` クラスファイルにデリゲートメソッドのプロトコルを追加 3 行目

【誤】 `protocol SwitchCellDelegate {`

【正】 `protocol SwitchCellDelegate: AnyObject {`

※「AnyObject」の箇所は「class」と書くこともできます。Xcode で「class」の定義にジャンプすると「AnyObject」の定義が表示されるようになっています。

●224 ページ ▼ `SwitchCell` クラスファイルにデリゲートメソッドのプロトコルを追加 3 行目

【誤】 `var delegate: SwitchCellDelegate? = nil // 変数を追加`

【正】 `weak var delegate: SwitchCellDelegate? = nil // 変数を追加`

※循環参照が発生し、メモリリークとなる恐れがあるため。

●229 ページ ▼ `NSNotificationCenter` クラスを利用した通知の一例 コードのタイトル

【誤】 `NotificationCenter`

【正】 `NotificationCenter`

※Swift 3.0 より Foundation Framework に含まれる一部のクラス、プロトコルから接頭辞の `NS` が削除されました。Swift では `NotificationCenter` を使用します。

●229 ページ ▼ `NSNotificationCenter` クラスを利用した通知の一例 1 行目

【誤】 `NSNotification.Name`

【正】 `Notification.Name`

●232 ページ ▼ `FirstViewController` クラス内で行う通知元の設定 6~7 行目

【誤】 `NSNotification.Name`

【正】 `Notification.Name`

●233 ページ ▼ `SecondViewController` クラス内で行う通知元の設定 下から 2~3 行目

【誤】 `NSNotification.Name`

【正】 `Notification.Name`

●233 ページ ▼ `FirstViewController` クラス内で行う通知先の設定 下から 3~4 行目

【誤】 `NSNotification.Name`

【正】 `Notification.Name`

●234 ページ ▼ `SecondViewController` クラス内で行う通知先の設定 下から 3~4 行目

【誤】 `NSNotification.Name`

【正】 `Notification.Name`

●485 ページ ▼ iOS のバージョンの分岐処理 3 行目

【誤】 `self.navigationItem.title = "記事一覧"`

【正】 `self.navigationItem.title = "最新記事"`

●517 ページ 5 行目、及び下から 2 行目

【誤】 `Provisioning Files`

【正】 `Provisioning Profile`

●518 ページ 「Provisioning Files」作成の流れ タイトル及び下から 2 行目

【誤】 `Provisioning Files`

【正】 `Provisioning Profile`

●518 ページ 本文 2、3、5 行目

【誤】 Provisioning Files

【正】 Provisioning **Profile**

<本書サポートサイト>

<http://www.shuwasystem.co.jp/support/7980html/5442.html>

<秀和システム>

<http://www.shuwasystem.co.jp/>

※本書の正誤情報・補足情報につきましては、以下 URL でも公開しています。

<http://takashings.hatenablog.com/entry/swift4-seigohyo>