

iOS Core Data 徹底入門

補足

2014.02.03 初版

2014.08.14 初版

「プロトコル宣言の <NSObject> について」左欄にある本文参照ページが不適切だったのを修正。

プロトコル宣言の <NSObject> について

サンプル :6.2.3-moleBuster MAdminViewController.h では

```
@protocol MAdminViewControllerDelegate
```

となっていますが、これは著者の記述ミスです。本来は

```
@protocol MAdminViewControllerDelegate<NSObject>
```

とした方が適切です。

この <NSObject> という追加記述は MAdminViewControllerDelegate が NSObject プロトコルを採用している事をコンパイラに伝えるためのものです。追加記述がなくても、ビルド時や実行時に注意やエラーが出るわけではありませんが、プロトコル宣言では習慣づけておいた方がいいでしょう。

NSObject プロトコルの採用は、デリゲート側が実際に -adminViewControllerDidFinish: というメソッドを用意しているか調べる時に必要となるものです。

例えばサンプル :6.2.3-moleBuster の MAdminViewController では、次のように -done: メソッドを実装しています。

サンプル :6.2.3-moleBuster MAdminViewController.m の元々の実装

```
- (IBAction)done:(id)sender
{
    [self.delegate adminViewControllerDidFinish:self];
}
```

このメソッドで、デリゲート側 (self.delegate) が実際に -adminViewControllerDidFinish: というメソッドを実装しているかを実行時に調べたいなら、次のように NSObject プロトコルで宣言された -respondsToSelector: などを使うことになります。

```
- (IBAction)done:(id)sender
{
    if ([self.delegate respondsToSelector:@selector(adminViewControllerDidFinish:)]) {
        [self.delegate adminViewControllerDidFinish:self];
    }
}
```

この時、一般的に次のように宣言して用意する delegate が -respondsToSelector: というメソッドが見当たらないと Xcode に注意されないために、MAdminViewControllerDelegate

プロトコル宣言の段階で NSObject プロトコルを採用しておくわけです。

```
@interface MBAdminController : UIViewController
@property (weak, nonatomic) id <MBAdminControllerDelegate> delegate;
. . .
@end
```

497 ページ

日付別でセクション表示させた場合に、セクション内の生徒名が順番に並ばない理由

488 ページでもふれましたが、日付別でセクション表示をさせた場合には、セクション内の生徒名が順番に並びません。

日付セクション内の名前が順番に並ばない

+	日付順	名前順	Edit
20130101			
name-9 1組	[01/01]	94点	
name-9 1組	[01/01]	16点	
20130102			
name-0 1組	[01/02]	02点	
name-9 1組	[01/02]	39点	
name-8 3組	[01/02]	49点	
name-5 3組	[01/02]	45点	
name-3 1組	[01/02]	76点	
20130103			

この原因は NSFetchedResultsController に渡す NSFetchedRequest が保持する並べ替え用の NSSortDescriptor のうち、日付用の sortDateDescriptors が "date" 属性で指定されているためです。

```
NSSortDescriptor *sortDateDescriptors
= [[NSSortDescriptor alloc] initWithKey:@"date" ascending:YES];
```

このためセクション自体は "yearmonthday" 属性により、年月日で分類されるのですが、その中の成績は時分秒で並べられた上で、名前順で並べられる事になります。

1、時分秒も含めて並べられる

2、時分秒まで一致したものは名前でも並べられる

	2013/01/02 09:00:00	2013/01/02 09:00:00
	name-0	name-9
	2013/01/02 11:10:00	name-8
	2013/01/02 18:00:00	name-5
	2013/01/02 18:10:00	name-3

2013/01/02 のセクション

→

	日付順	名前順	Edit
20130101			
	name-9 1組 [01/01] 94点		
	name-9 1組 [01/01] 16点		
20130102			
	name-0 1組 [01/02] 02点		
	name-9 1組 [01/02] 39点		
	name-8 3組 [01/02] 49点		
	name-5 3組 [01/02] 45点		
	name-3 1組 [01/02] 76点		
20130103			

もし、セクション内を年月日のみ（時分秒を無視する）で並べた上で、名前順で並べたい時は "yearmonthday" 属性を日付並べ替え用の `sortDateDescriptors` に指定する必要があるのですが、この場合、`NSFetchedResultsController` の並べ替えに指定する `NSSortDescriptor` の属性名は `Record` エンティティの持つ属性に限られる事から、"yearmonthday" 属性を `EDRecord` のメソッドではなく、`Record` エンティティの一時的でない属性として用意しておくかなければいけません。

サンプル：11.1.6-fetchedresults の場合、記録はダミーデータなのでマイグレーションまでする必要はありませんが、少なくとも `Record` エンティティの属性として "yearmonthday" を用意して最初から作りなおす必要があります。

もちろん "date" に記録する日付を、時分秒に関しては 00:00:00 に固定する方法も考えられますが、それよりは "date" とは別に年月日のみを管理する "yearmonthday" 属性を用意する方がいいでしょう。

"date" と "yearmonthday" 属性の連動には、これまで学習した KVO の見張りや `NSManagedObject` 派生クラスでのアクセッサメソッドのオーバーライドなどが考えられますし、単純に `NSManagedObject` 利用側が、"date" 属性を変更するときには一緒に "yearmonthday" 属性も更新するとしてもいいでしょう。

サンプル：11.2.1-moleBuster に対しても同じ事がいえます。こちらはこれまでの記録を残したいならマイグレーションで対応するべきでしょう。

以上で補足説明を終わります。

お疲れさまでした。