

補足資料：親クラスと継承

ここでは本書内では解説を割愛した「親クラス」について、もう少し詳しく解説します。本書を読み進めていくなかで、親クラスについてより詳しく知りたくなった際に読み進めてください。

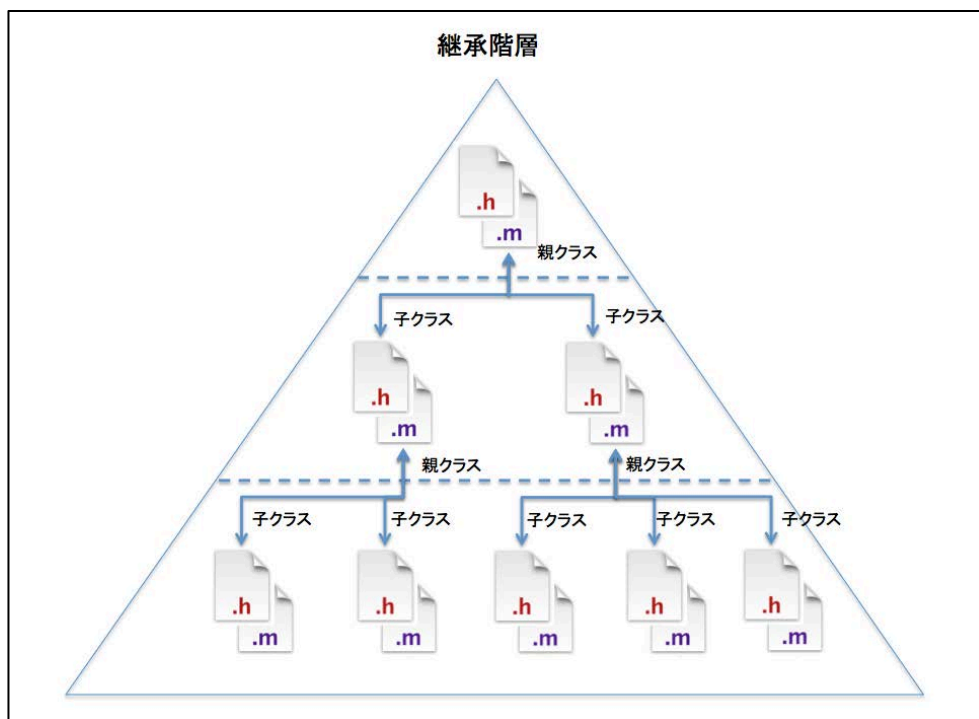
□ 継承階層

親クラスが何であるかを理解するためには『**継承階層 (inheritance hierarchy)**』という用語を知っておく必要があります。

まずは下図を見てください。親クラスと子クラスの配置によって定義されるクラスの階層です。すべてのクラスは親クラスを1つだけ持ちます(※)。またクラスは無数に子クラスを持つことが可能です。

(※) NSObject クラスというルートクラス（最上位のクラス）のみ親クラスを持ちません。

図：クラスの継承階層



この図はクラスの継承階層を表しています。ピラミッドのようにクラスが配置されています。この図では階層が3層に分かれています。頂点のクラスから下層に向かってクラスが枝分かれし、さらにその下の階層へクラスが枝分かれしています。このように複数のクラスによって構成される階層を『継承階層』といいます。

● 親クラスと子クラス

『**親クラス (superclass)**』とは、継承階層においてあるクラスの 1 つ上の階層にあるクラスです。Objective-C では、クラス作成時に必ず親クラスを指定しなければなりません。省略することはできません。

また『**子クラス (subclass)**』とは継承階層においてあるクラスの 1 つ下の階層にあるクラスです。新たに作成するクラスは、その親クラスから見ると「子クラス」になります。

□ 継承とは

継承とは、親クラスの特性（インスタンス変数とメソッド）を子クラスに渡すことのできる機能です。

親クラスを指定してクラスを作成すると、子クラス（これから私たちが作成するクラス）には、親クラスに定義されているインスタンス変数やメソッドが自動的に引き継がれます。そのため、子クラスではそれらを別途定義しなくても使用することができます。

● iPhone アプリにおける継承の具体例

下図をご覧ください。この図は iPhone アプリにおける継承の具体例です。上が写真アプリ、左下がメモアプリ、右下が設定アプリです。これらのアプリでは『**UIScrollView クラス**』という、iPhone 画面に収まりきらない大きなコンテンツを表示する場合に使用するクラスが使われています。

図：UIScrollView の継承階層



カメラで写真を取るとその写真は本来 iPhone 画面には収まりきらない大きさの解像度を持っています。デフォルトでは縮小されて画面に表示されていますが、ダブルタップなどをすると拡大することができ、拡大した画像はフリックなどで表示をスクロールしたりできます。写真アプリのこの画面は UIScrollView クラスによって実現されています。

この UIScrollView クラスを親クラスにすることで、その特性をそのまま引き継いだクラスを使っているのがメモアプリと設定アプリです。メモアプリは、UIScrollView クラスが持っているスクロール機能に加えて、画面にテキストを表示する機能を持っています。また、設定アプリは UIScrollView クラスが持っているスクロール機能に加えて、設定項目を一覧表示する機能を持っています。これらの機能は UIScrollView クラスを継承した(UIScrollView クラスを親クラスとした)子クラスが実現しています。

このように継承という機能を利用すると、すでにある機能を新たに作成することなく、利用することができます。

● 親クラスの指定方法

新しく MyString クラスを作成する際に、親クラスとして NSString クラスを指定する場合は以下のように入ります。

List : MyString クラスの作成 (親クラスは NSString クラス)

```
@interface MyString : NSString
```

上記の MyString クラスの場合は、親クラスに NSString クラスが指定されているので、NSString クラスに実装されている length メソッドを、MyString クラスでも使用できます。これをコードで見ると以下のように入ります。

List : MyString クラスの使用

```
MyString *string = @"こんにちは";  
int number = [string length];
```

このように、MyString クラスには length メソッドに関する記述はないのですが、NSString クラスのメソッドが継承されているため、そのメソッドを使用できるのです。NSString クラスの他のメソッドも同様に使用できます。これらのことから、通常は新規作成するクラスに関連するクラスを親クラスとして指定します (文字に関するクラスを作成する場合は NSString クラスを親クラスに設定するなど)。

□ NSObject クラスとは

本書の P.95 で作成した Animal クラスの親クラスとして指定した NSObject クラスは『**ルートクラス**』と呼ばれる、Objective-C における最上位のクラスです。もし特に決まった親クラスがない場合はこのクラスを親クラスにします。

なお、インスタンスの生成や初期化を行う際に使用する alloc メソッドや init メソッドは、NSObject クラスに実装されているメソッドです。

□ 既存クラスの親クラスの調べ方

NSObject を除くすべてのクラスには親クラスが設定されています (NSObject は最上位クラスなので親クラスはありません)。これは Objective-C にあらかじめ用意されているクラスの場合も同様です。

既存クラスの親クラスを調べるには、Xcode 上で [option] キーを押しながら、対象のクラス名の上にカーソルを移動します。すると、カーソルが「？」に変化します。

```
8
9 #import <Foundation/Foundation.h>
10
11 @interface MyClass : NSObject
12 @property NSString *myString;
13 @end
14
```

上図の状態でクラス名をダブルクリックするとオーガナイザウィンドウが開き、ドキュメンテーションが表示されます。これがクラスのリファレンスになります。ここには親クラスだけでなく、インスタンス変数やメソッドなども記載されています。親クラスの情報には『**inherits from**』（継承元）という項目に記載されています。下図を見ると、NSString の親クラスが NSObject であることがわかります。

The screenshot shows the Xcode Organizer - Documentation window. The left sidebar has a search bar with 'NSString' entered. Below the search bar are filters for 'Hits must', 'Languages', and 'Find in'. The main content area is titled 'NSString Class Reference'. A red box highlights the 'Inherits from' field, which shows 'NSObject'. Other fields include 'Conforms to' (listing NSObject, NSMutableCopying, etc.), 'Framework' (/System/Library/Frameworks/Foundation.framework), 'Availability' (Available in iOS 2.0 and later), 'Declared in' (listing NSString.h, NSURL.h, etc.), 'Companion guides' (String Programming Guide, Property List Programming Guide), and 'Related sample code' (AdvancedURLConnections, BonjourWeb, etc.).

Field	Value
Inherits from	NSObject
Conforms to	NSObject, NSMutableCopying, NSCopying, NSSecureCoding, NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in iOS 2.0 and later.
Declared in	NSLinguisticTagger.h, NSPathUtilities.h, NSString.h, NSURL.h
Companion guides	String Programming Guide, Property List Programming Guide
Related sample code	AdvancedURLConnections, BonjourWeb, CoreTextPageViewer, MVCNetworking, TableViewSuite