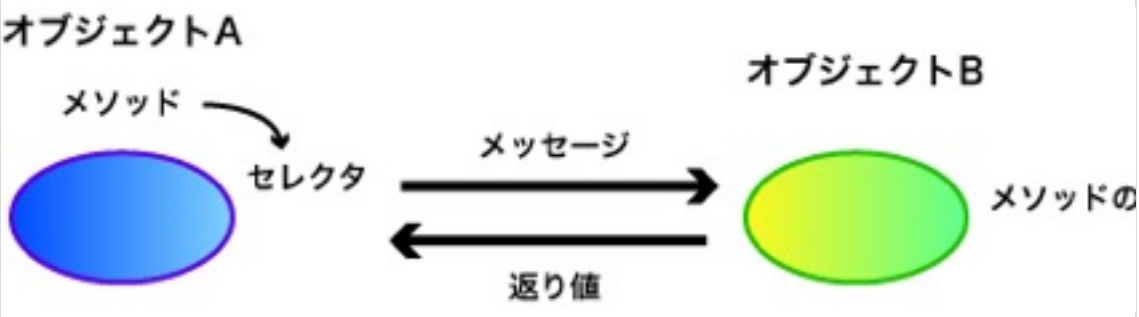
１０月２２日
まとめ：Objective-c でクラスの実体は C 言語の構造体。メソッドも構造体。これが Objective-c の動的に繋がる。

Objective-c：動的

①ポージング：既存のクラスを「乗っ取る」ことができる機能。
方法：poseAsClass:というメソッドで行う。このメソッドが呼ばれたクラスは、引数で渡されたクラスのように振る舞うことになる。

②メッセージ送信の仕組み



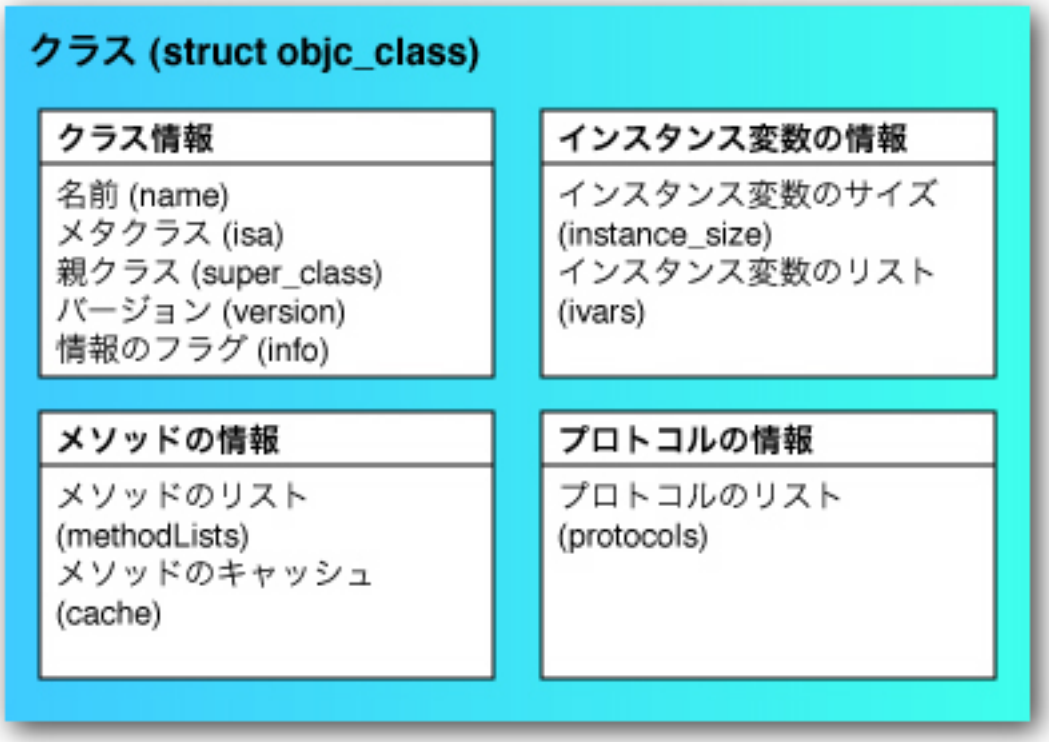
メッセージ送信

1. オブジェクト A がメソッドを指定
   まず、ソースコード上で呼び出すメソッドが指定される。
2. メソッドのためのセレクタを取得
   ソースコードをコンパイルすると、メソッドはセレクタでとして認識される。ここでは、セレクタはただの文字列である。ランタイムにこのコードが読み込まれると、一意のセレクタに変換される。
3. オブジェクト B にメッセージを送信
   メッセージ送信には、objc_msgSend というランタイム API を使う。この API に、ターゲットとなるオブジェクト B と、セレクタを渡す事になる。
4. セレクタからメソッドの実装を取得
   まず、オブジェクト B からクラス情報を取り出して、メソッドの実装がキャッシュされているかどうか調べる。キャッシュされていない場合は、

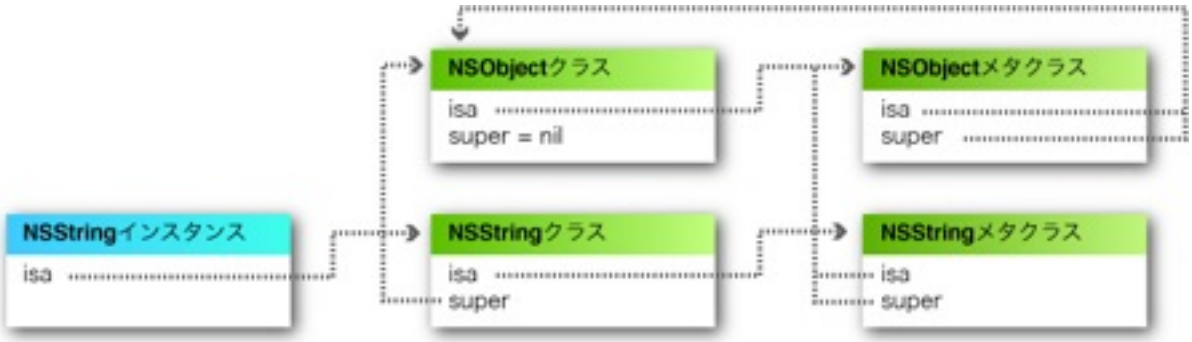
クラスが持つメソッド一覧から Method 構造体を探して、メソッドの実装を取り出す。

5. 実行
   関数ポインタである、メソッドの実装を呼び出す。このとき、第一引数としてオブジェクト、第二引数としてセレクタを渡す事になる。
6. 返り値を送信
   メソッドの実装の返り値を、objc_msgSend の返り値として返す。

③Objective-C のクラスの定義：構造体
以下の情報が含まれる。



④メタクラス：クラスのためのクラス。最終的には NSObject を指定される。



利点：クラスもオブジェクトとして見なせる。NSObject のインスタンスメソッドを使える。

⑤メソッド：

- メソッド　Method 型(セレクタ、引数、メソッドの実装のポインタ)
- セレクタ　SEL 型（objc_selector 構造体のポインタ型）
- メソッドの実装　IMP 型（関数のポインタ）

⑥Foundation と Core Foundation の関係：Foundation＝Core Foundation
ほぼ同じ。Core Foundation は C 言語で表現された Foundation。

⑦**Toll-free bridge**：Objective-c のオブジェクトは C 言語の構造体。変換にパフォーマンスの影響無し。

⑧クラスクラスタ：抽象クラス。Factory Method パターン＋Template Method パターン。