

Swift Programming Lecture 1

Backgrounds of Swift Language
Development Environments

Tatsuo Minohara



Swiftプログラミング言語の歴史

- ほぼ一番新しいプログラミング言語、1.0版は2014年にリリース
- 異常に使い勝手の悪かったObjective-C言語に代わるもの
- Mac OS X, iPad (iPad OS), iPhone(iOS)については、Swiftでアプリケーションを開発することが標準となる
- 言語仕様を過去に囚われず、次々に更新！開発する方は大迷惑！新版は最新のMac OS Xの版とXcodeの版を必要とする
 - Swift 2.0 2015年
 - Swift 3.0, 3.1 2017年
 - Swift 4.0, 4.1, 4.2 2017年～2018年
 - Swift 5.0～5.5 2019年～2021年
 - Swift 5.6～5.8 2022年～2023年



オブジェクト指向言語の系譜（初期）

- Simula...最初のオブジェクト指向言語と呼ばれる
- Smalltalk-80...Xerox Palo Alto研究所のAlto Workstationで稼働開始（1974年）、その後のオブジェクト指向プログラミング言語の始祖となる（Smalltalkで定義された用語が現在も使われている）
- C++...Bjarne Stroustrupが1980年代前半に策定、C言語に対して強引にオブジェクト指向を拡張したもの



オブジェクト指向言語の系譜（後期）

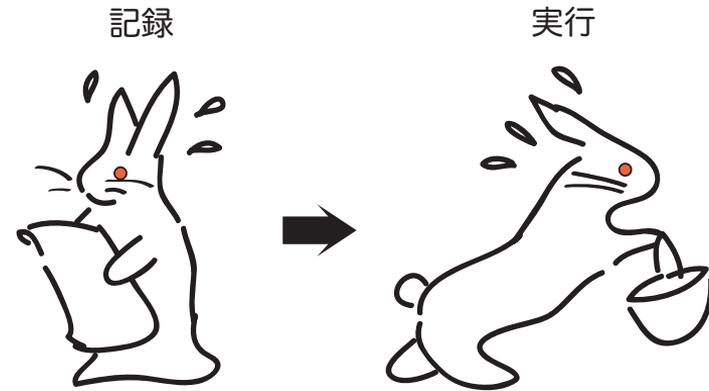
- Objective-C...1980年代後半NeXT社で開発、SmalltalkとC言語の悪いところ取りで、記号ばかりで非常に記述しにくい。Steve Jobsと共に、Apple社に異動してきて、Mac OS XのCocoaライブラリを形成
- Java...1990年代前半Sun Microsystems社で開発、C++の悪いところを改良して、言語自体はまともだが、Sunが吸収合併されてOracle社に移管されて誰にも使われなくなってきて、C++への回帰現象がでてくる。それでも、90年代設計の古さがあり、衰退は停まらず
- Python...2000年代に勃興、使いやすさ、記述のしやすさで、主流になりつつある



Swiftの言語に影響を与えた言語

- C, C++ → Java, JavaScript → Swift : 言語自体が似ている、C言語のくどさを引摺る
- Python → Swift : 版が改訂されるたびに、Python寄りになっている
- C# → Swift : C++, Javaと似ているが、C#の記法を一部導入
- Objective-C → Swift : Interface Builderの使いにくいところを未だに引摺っている

コンピュータの動き



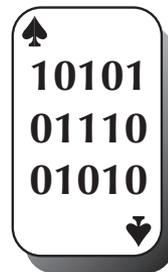
コンピュータはプログラムを一度メモリに記憶してから実行する。
⇒フォン・ノイマン式 (Von Neumann : プリンストン高等研究所)



プログラムのレベル



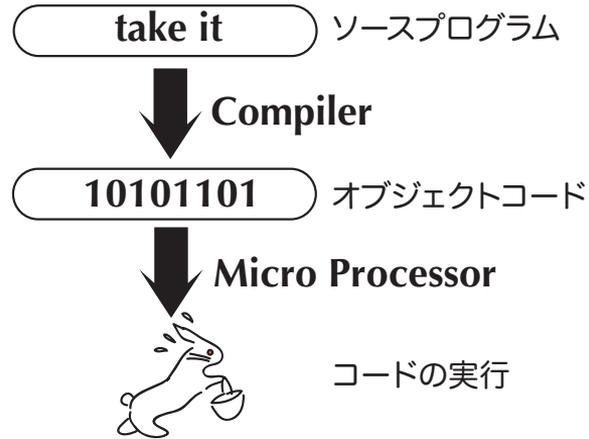
人間が記述する
プログラム



コンピュータ用の
プログラム

人間が記述するプログラムとコンピュータが実行できるプログラムの表現方式が異なる。

コンパイラ方式

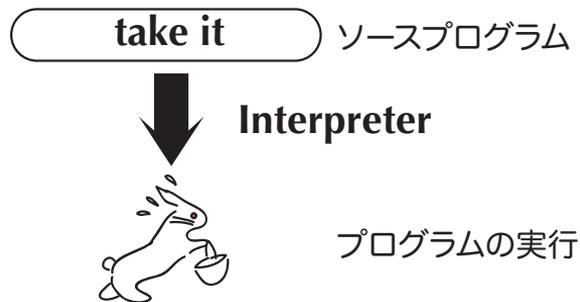


予めコンパイラでコンピュータレベルのプログラム
(マシン語のプログラム) に変換しておく。

⇒高速に実行できるが、各マシン (CPU) やOSごと
に変換しなければならない

例：C/C++, C#, Fortran, Swiftなど

インタプリタ方式



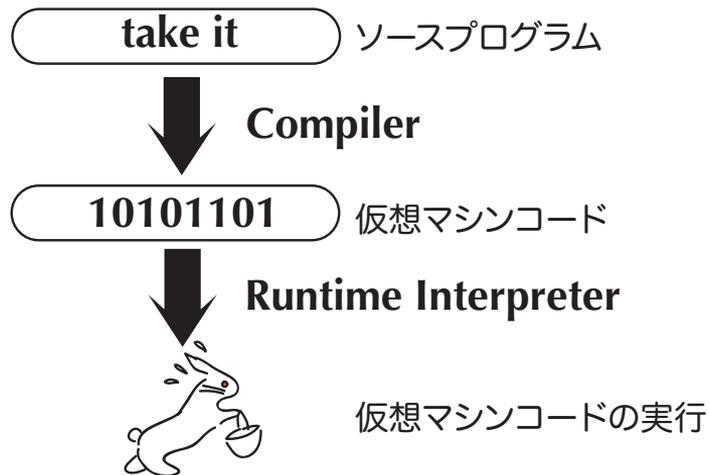
インタプリタが、いちいち解釈しながら実行する。インタプリタさえあればどこでも実行される。

最近では、REPLとも呼ばれる (Read-Eval-Print-Loop)

⇒実行が低速になる。

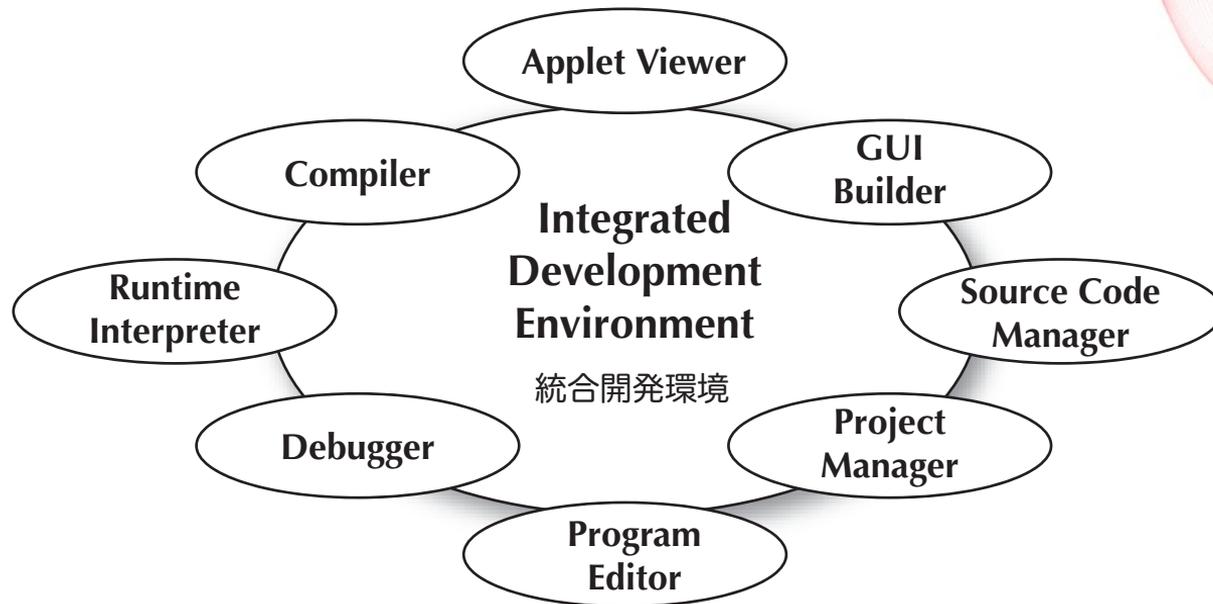
例：Python, Lua, Ruby, Perl, C-shell, Lisp, JavaScript, Swiftなど

中間（仮想）コード方式



両方式の中間的なもの。特定のマシン語のプログラムではなく、仮想マシンのプログラムに変換する
⇒ランタイム・インタープリタは、割と小さく高速に動くプログラムなので、そこそこの速度で実行可能
例：Java, Pascal, Smalltalkなど

統合的な開発環境





Mac OS Xでの開発アプリケーションのダウンロード

- App Store以外のアプリケーションをWebからダウンロードしてインストールする場合、インストールの制限が掛かっています
- ターミナルで、以下のコマンドを打って下さい。
- `sudo spctl --master-disable`
- システム環境設定 >> セキュリティとプライバシー
- 一般のタブで、ダウンロードしたアプリケーションの実行許可
- すべてのアプリケーションをOKにします（できたら）



拡張子を表示する、作業用のフォルダ

- Macintosh: Finderの環境設定
 - 詳細タブで「すべてのファイル名の拡張子を表示」
- デスクトップでFinderで、新規フォルダ作成
 - 「Swift2021」という名前で作成する

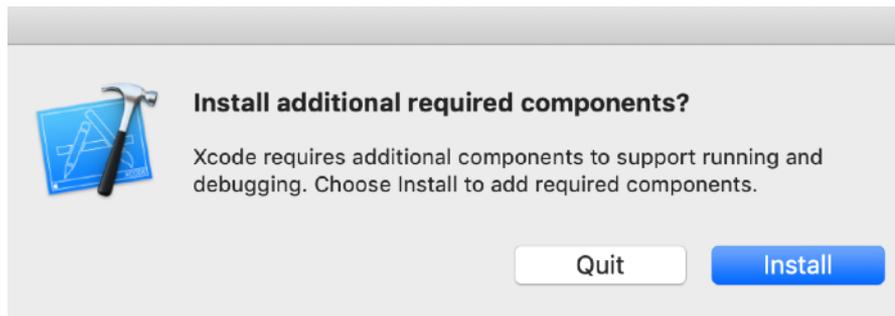


Xcodeのダウンロード

- Xcode統合開発環境の中に、Swiftのコンパイラ、インタープリタも入っている。
- App Storeから、Xcodeをダウンロードする。Xcode 13は、17GB以上もあるので、ダウンロードインストールに時間が掛かる。
- Xcode 13は、Mac OS Big surの上で稼働することを前提としている
- Mac OS Xで動くバージョン（Xcode 10/11/12など）についても、developer.apple.comのサイトからダウンロードできる。
- ただし、このサイトは、デベロッパ登録（無料）しなければならない。
- また、Swift言語自体は、オープンソース化されているので、swift.orgからもダウンロードできる
- <https://swift.org/download/#snapshots>

Xcodeのインストール

- Xcodeのインストールは、Xcode13.xipファイルをダウンロードした場合は、まず、ダブルクリックして展開する。
 - Xcode.appが34GBの大きさを、そのフォルダにできる
- Xcode.appをアプリケーションフォルダに移動する。App Storeからのダウンロードでは、ここまでは自動的にやってくれる。
- Xcode.appを起動すると、以下のような追加コンポーネントをインストールするか出てくる場合があるので、その場合は、それもインストールする





Swiftのインタプリタを動かす

- ターミナルで、Swiftのインタプリタを動かすためには、ライセンス認証が必要になるかも知れない。以下のコマンド実行しておく。
 - `sudo xcodebuild -license`
 - Spaceキーでスクロールして、最後に、`agree`を入力してReturnを押す
- `swift` (パスとしては、`/usr/bin/swift`) を入力する
 - `:help`で英語のヘルプが見える
 - `:exit`または`control+D`で終了



Swift開発環境（授業で扱うもの）

- Swiftインタプリタ
 - ターミナルから使う
- BBEdit（MacOSX用：フリー版：複数の言語に対応）
 - AppStoreからインストールする
 - 一行目に#!/usr/bin/swiftを書く必要がある
 - 実行を別ウィンドウ（ターミナル.app）で出すことができる
- Visual Studio Code（VSCode：フリー版：複数の言語対応）
 - Swift用の拡張設定をダウンロードする必要がある。



BBEditのダウンロード

- App Storeを立ち上げる
- 検索で、「BBEdit」を見つけて、ダウンロード
- Preferencesを選ぶ
- Editor Defaultsのパネルを選ぶ
- Default FontsのSelect...ボタンを押す
- メイリオ・Lucida Grandeで、24ptのフォントを選ぶ

BBEditのショートカットキーの設定

- 実行は、保存して、#!メニューのRunあるいは、Run in Terminalを選ぶと実行される
- Run in Terminalの場合は、ターミナルのアプリケーションが立ち上がってそこで実行される
- Run in Terminalを⌘Rで実行させる場合は、以下の設定を行なう
 - アプリケーションメニューのPreferencesを選んで設定のダイアログを出す
 - Menu & Short Cutキーを選ぶ
 - 左のリストで、#!メニューを選び、右のリストでRun in Terminalのショートカットキーのテキスト部分を選び、⌘Rを入力する



BBEditでSwift用の設定

- App StoreからBBEditをインストールしておく
- 起動したあと、Swiftの言語ハイライト（予約語などの色を変える）をするため、以下のサイトから、swift.plistをダウンロードする。
 - <https://github.com/curtclifton/bbedit-swift-clm>
- /Users/ログイン名/Library/Application Support/BBEdit/Language Modulesのフォルダに移動する
- ライブラリフォルダが見えない場合は、Shift + ⌘ + Gで以下のフォルダのパスを入れてFinderで開く
 - ~/Library/



SwiftのApple標準ライブラリ

- Core Foundation/Foundation ... 言語自体に関するもの
- AppKit ... アプリケーション作成のためのもの
- UIKit ... GUIのためのもの（これまでのもの）
- SwiftUI ... GUIのためのもの（5.1以降はこちらが主流）
- MapKit ... Appleの地図を利用した地図表示
- SceneKit ... 3次元CG（Colladaファイルしか読めない！）
- SpriteKit ... 2次元CG

- 標準が使いにくい場合は、RxSwift等の第三者（Third Party）ライブラリがあるので、いろいろ探すべし



Swiftのリファレンス

- iBook.app (ブック・アプリケーション)
 - The Swift Programming Language (Swift 5 edition)
 - App Development with Swift (iOS 11 edition)
- Xcode Help メニュー
 - Developer Documentation

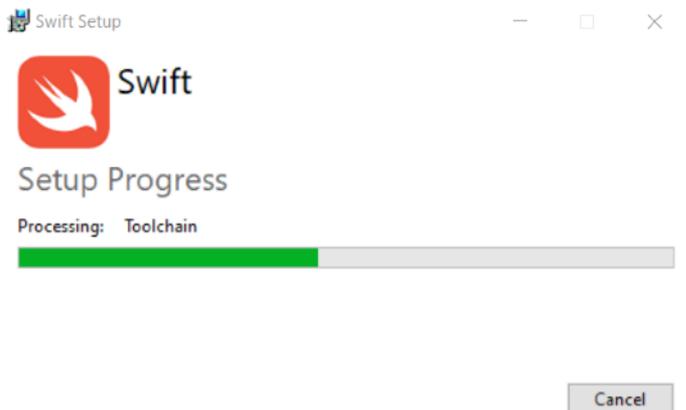


WindowsでのSwiftインストールについて(1)

- PowerShell上でwingetコマンドを使ってGitとPythonをインストールします。
 - `winget install Git.Git`
 - `winget install Python.Python.3.10` ← Pythonを事前にインストールしていない場合
- Pythonのpipコマンドを使って、sixライブラリをインストールします。
 - `pip install six`
- 自分用の備忘録：PowerShellの実行の仕方
 - Windowsキー + X → メニューから、PowerShell（ターミナルと表示されている）を実行する

WindowsでのSwiftインストールについて(2)

- SwiftのWindows版のInstaller（Stable版かDevelopment Snapshot版）をダウンロードして、インストールします。
 - <https://swift.org/download/>
- ちなみに、以下のフォルダにインストールされます。
 - %SystemDrive%\Library\Developer\Toolchains



ターミナルからのswift実行

- swift replあるいはswift
 - swiftのインタプリタが立ち上がります。
 - :quitで終了させることができます。
- swiftc ファイル名.swift
 - swiftのコンパイラが起動されます。
 - コンパイルされるとファイル名.exeという実行ファイルが生成されます。
 - ./ファイル名.exeで実行可能（Windowsの場合は、.\ファイル名.exe）

VSCodeでSwiftの機能拡張を追加

- VSCodeでSwiftを編集する場合、下記の機能拡張を追加すると、予約語などの色を自動的に変えてくれたり、補完機能を使うことができる。
- 追加機能：
 - ▶ Code completion
 - ▶ Jump to definition, peek definition, find all references, symbol search
 - ▶ Error annotations and apply suggestions from errors
 - ▶ Automatic generation of launch configurations for debugging with CodeLLDB
 - ▶ Automatic task creation
 - ▶ Package dependency view
 - ▶ Test Explorer view



Swift v1.2.1

Swift Server Work Group | 111,630 | ★★★★★ (19)

Swift Language Support for Visual Studio Code.

インストール ⚙️

VSCodeでの設定（macOSの場合）

- 構成の追加を行ない、launch.jsonを以下のように記述して、該当フォルダに保存するとswiftインタープリタを利用して、.swiftファイルを実行できる（デバッグなしで）。
- 記述例

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Swift Run",
      "type": "node",
      "request": "launch",
      "runtimeExecutable": "swift",
      "program": "${file}",
      "console": "integratedTerminal"
    }
  ]
}
```

VSCodeの設定 (Windowsの場合)

- Swiftのインタプリタがクラッシュする (5.8版でも) ために、一旦swiftcでコンパイルして、作成された実行ファイルを実行させるために、次のような構成ファイルを追加する必要がある。
- 構成ファイルの記述例 (launch.json) :

```
{ "version": "0.2.0",  
  "configurations": [  
    { "name": "Swift Build and Run",  
      "type": "node",  
      "request": "launch",  
      "runtimeExecutable": "cmd.exe",  
      "cwd": "${workspaceFolder}",  
      "args": [ "/c", "swiftc", "${file}", "-o",
```

```
    "${fileDirname}\\${  
fileBasenameNoExtension}.exe",  
    "&&",  
    "${fileDirname}\\${  
fileBasenameNoExtension}.exe" ],  
    "console":"integratedTerminal"  
  },  
  { "type": "lldb",  
    "request": "launch",  
    "name": "Debug",  
    "program": "$  
fileBasenameNoExtension}.exe",  
    "args": [],  
    "cwd": "${workspaceFolder}"  
  }  
  ] }
```

その他のSwiftプログラミング環境

- Web上でのプログラミング
 - ▶ <https://replit.com/languages/swift>
 - ▶ swift以外にも、Python, Java, C#, C/C++など様々なプログラミング言語をプログラミングすることができる
 - ▶ AIを利用したライブラリなどが入ったものもある。
- iPhone/iPad上でのプログラミング
 - ▶ Swifty Compiler
 - ▶ swiftの最新版に早めに対応している



Swifty Compiler EX

Hassan Taleb

iPad対応

「開発ツール」内45位

★★★★★ 5.0 / 3件の評価

無料・App内課金が有りません

[こちらで表示: Mac App Store](#)