

//-----/

Calc version 0.1

//-----//

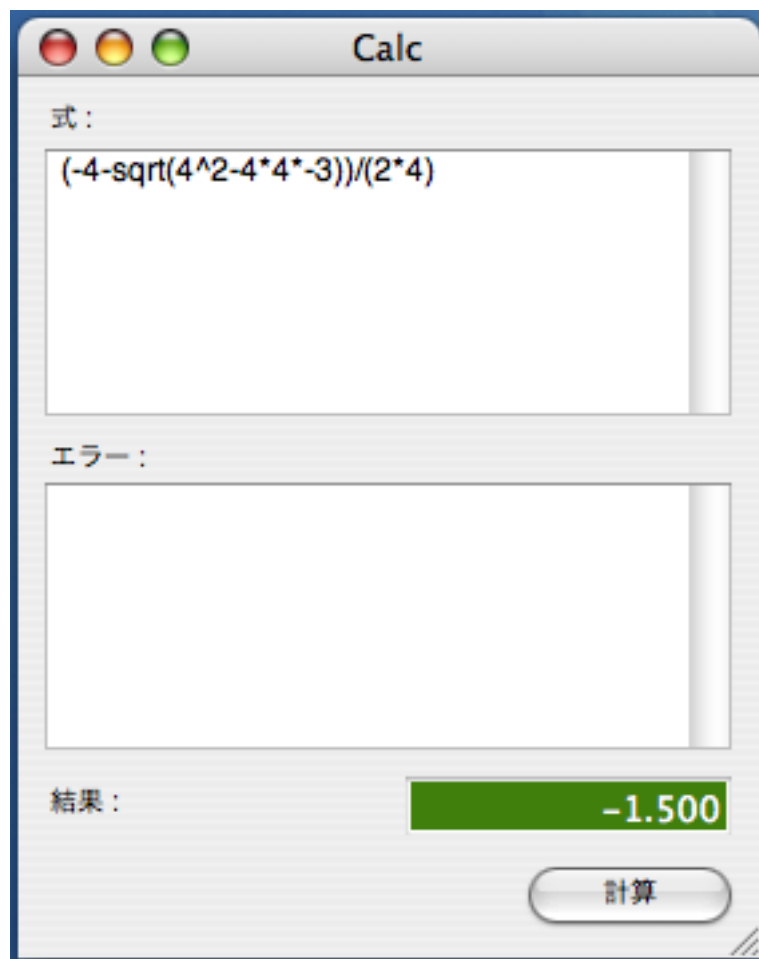


図 1) 実行例

【インストール】

Calc.zip を展開して作られる Calc フォルダを 'ホーム/Applications' に移動してください。
('ホーム' フォルダに 'Applications' フォルダがない場合は、新規に作成してください。)

【アンインストール】

'ホーム/Applications' の 'Calc' フォルダ、(中に含まれるすべてのファイル)
'ホーム/ライブラリ/Preferences' フォルダの jp.koyama.Calc.plist
を削除してください。

【説明】

図 1) のように上部にある式エリアに、計算したい式を入力してください。入力し終わったら、'計算' ボタンを押してください。
式エリアに間違いがある場合は中央部のエラーエリアに、エラーを表示します。エラーがなければ、結果が表示されます。

あくまでもプログラミングに挑戦したいために、取り組みました。重大なエラーがある可能性があるかもしれません。利用には十分注意してください。

定数、関数などを下表に記しておきます。内部では double 型として扱っています。

表 1) 定数, 関数

	例	結果
数字		
整数	10 + 10	20.000
小数	0.5 + 0.2	0.700
符号		
+	+5 + 5	10.000
-	- 5 + 5	0.000
定数		
pi	sin(pi)	0.000
e	ln(e)	1.000
加算 +	10 + 10.5	20.500
減算 -	10.5 - 0.5	10.000
乗算 *	2.5*3.4	8.500
除算 /	10/2	5.000
べき乗 ^	2^3	8.000
括弧 ()	(2+3) * 4	20.000
平方根	sqrt(2)	1.414
	sqrt(2^2 + 4^2)	4.472
三角関数		
sin	sin(45*pi/180)	0.707
cos	cos(45*pi/180)	0.707
tan	tan(45*pi/180)	1.000
対数関数		
自然対数 log	log(10)	1.000
常用対数 ln	ln(1)	0.000
指数関数		
exp	exp(0)	1.000

【著作権】

フリーウェア、含まれているソースの著作権は私に。変更して公開する場合は、私にメールを下さい。
私なりに参考文献 *1) のサンプルを理解してソースを書いているのですが、どうしても似てしまいます。そのため、その箇所の著作権は 参考文献 *1) にあります。問題があるかもしれません。

【保証】

無保証。このソフトを使用することで生じたいかなる損害も、作者は責任を負いません。

【動作環境】

動作確認は、Mac OS X バージョン 10.4.1 ビルド 8B15 でおこなっています。

【独り言】

2005/06/13(Mon)

書籍名 : スモールコンパイラの制作で学ぶプログラムのしくみ

出版社名 : 株式会社技術評論社

著者名 : 石田 綾 著

価格 : 1,974円

上記本と Web [<http://www.gihyo.co.jp/books/support.php/4-7741-2177-0>] の補足情報 (サンプルコード) を参考に Calc を作成しました。上記本でコンパイラをリンゴ農園に例えるより、普通に解説してもらったほうが良い感じがしました。ダウンロードしたサンプルのソースは大変勉強になりました。ただし、ダウンロードしたファイル source.zip を BOMArchiveHelper で解凍するとフォルダ名が文字化けします。StuffIt Expander で解凍してください。

解凍すると'ソース' フォルダに '2章 基本' と '3章 拡張' とフォルダができると思います。それぞれ Cell.java、CellG.java、CellK.java、CellV.java のソースの先頭 '`package cell;`' の部分をコメントアウト '`//package cell;`' します。

Xcode でプロジェクトを Cell という名前で新規作成し、先ほど変更を加えたソースをプロジェクトに追加します。メニュー 'Project/Edit Active Target 'Cell' ' を選択し、'Java Compiler Settings' の 'Source file encoding:' で 'Japanese (Mac OS)' を選択してください。

ソースファイルの指定は、左側の 'Executables/java' をインスペクタで開き、タブ 'Arguments' で設定してください。これでツールバーの Built and Go を選ぶとビルドと実行に成功します。

後、ターミナルで文字セットエンコーディングを設定して Cell.jar を実行するとエラーも日本語で表示されます。

文法

E expression

T term

F factor

$E ::= T \{ (- | +) T \}$

$T ::= F \{ (^ | * | /) F \}$

$F ::= ' (E) '$

$F ::= \text{Number}$

$F ::= (- | +) F$

$F ::= (\sin | \cos | \tan | \text{sqrt} | \log | \ln | \exp) ' (E) '$

うまく機能しているのには驚きました。こんな簡単に、字句解析、構文解析をおこなって計算機ができるとは思いませんでした。コンパイラ構成法の解説書は何冊も読みましたが、難しく理解できませんでした。この本の解説はもちろん参考になりましたが、サンプルのソースが一番理解に貢献しました。(サンプルソースは Java で書かれています。)それでも、まだまだですが！エラー処理はかなりいい加減です。あまり参考にならないと思います。

2005/06/17(Fri)

WWDC(世界開発者会議)の基調講演 [<http://stream.apple.akadns.net/>] で Apple が来年の6月までにインテル製マイクロプロセッサを使用したモデルのMacintosh の提供を開始し、さらに2007年末までにすべてのMacがインテル製マイクロプロセッサ搭載モデルに移行する計画を発表しましたね。[<http://www.apple.com/jp/news/2005/jun/07intel.html>]

私個人としては iMac G5 で後3年ぐらい買い替えないつもりでいたので微妙ですね。後、来年は Longhorn 搭載の Windows を購入しようと思っていたので悩みどころです。インテル製 iMac で CPU が 64 bit に対応していれば文句なしに買うのですが！ Windows についてはデュアルブートより、エミュレータで実用的な速度で動作するほうが良いですね！

私自身の Mac 利用目的はプログラミングなので、インテルに移行してもあまり問題がありませんね。

Calc は [Universal Binary \[参考文献 *3\]](#) になっています。うまく動作するかわかりませんが！

ターミナル上で file コマンドを実行すると下記のように Universal Binary の確認ができます。情報源は [OS Xハッキング! 第138回 Mac on Intelへの道のり - Mach-Oから探る](#) [<http://pcweb.mycom.co.jp/column/osx/138/>] です。

```
-----[ ターミナル ]-----
yoshiaki-koyama-no-imac-g5:~ yoshiaki$ file
/Users/yoshiaki/Programming/Developer/Calc/Sources/build/Deployment/Calc.app/Contents/MacOS/Calc
/Users/yoshiaki/Programming/Developer/Calc/Sources/build/Deployment/Calc.app/Contents/MacOS/Calc:
Mach-O fat file with 2 architectures
/Users/yoshiaki/Programming/Developer/Calc/Sources/build/Deployment/Calc.app/Contents/MacOS/Calc (for
architecture ppc):      Mach-O executable ppc
/Users/yoshiaki/Programming/Developer/Calc/Sources/build/Deployment/Calc.app/Contents/MacOS/Calc (for
architecture i386):     Mach-O executable i386
yoshiaki-koyama-no-imac-g5:~ yoshiaki$
-----
```

【開発環境】

Xcode 2.1
Component versions
Xcode IDE: 621.0
Xcode Core: 620.0
ToolSupport: 610.0
Interface Builder Version 2.5.1 (439)
The X Window System X11 1.1 - XFree86 4.4.0
gimp 2.2
EGWORD 14

【参考文献】

*1)
書籍名 : スモールコンパイラの制作で学ぶプログラムのしくみ
出版社名 : 株式会社技術評論社
著者名 : 石田 綾 著
価格 : 1,974円

補足情報 (サンプルコード) [<http://www.gihyo.co.jp/books/support.php/4-7741-2177-0>]

Calc ReadMe 0.1

*2)

[Apple Developer Documentation](#)

<http://developer.apple.com/documentation/>

[Cocoa Documentation](#)

<http://developer.apple.com/documentation/Cocoa/Cocoa.html>

[Cocoa API Reference](#)

http://developer.apple.com/referencelibrary/API_Fundamentals/Cocoa-api-date.html

[Application Kit Reference for Objective-C](#)

http://developer.apple.com/documentation/Cocoa/Reference/ApplicationKit/ObjC_classic/index.html

[Foundation Reference for Objective-C](#)

http://developer.apple.com/documentation/Cocoa/Reference/Foundation/ObjC_classic/index.html

*3)

[Universal Binary Programming Guidelines](#)

http://developer.apple.com/documentation/MacOSX/Conceptual/universal_binary/index.html

*4)

[OS X ハッキング!](#)

第138回 Mac on Intelへの道のり - Mach-Oから探る

<http://pcweb.mycom.co.jp/column/osx/138/>

【メール】

御意見、感想はこちらまで。

小山 嘉昭

KOYAMA Yoshiaki

koyama-@za2.so-net.ne.jp

yos_koyama@mac.com

Yoshiaki's HomePage

<http://koyama-cool.ne.jp/>

<http://www001.upp.so-net.ne.jp/koyama-/>