「利子を得る」関数の陰仕様と、ニュートン法による陽仕様

佐原伸
日本フィッツ株式会社
情報技術研究所
TEL：03-3623-4683
shin.sahara@jfits.co.jp

2005 年 5 月 30 日

概 要

「利子を得る」関数の陰仕様と、ニュートン法による陽仕様。

1 はじめに

ある年数で資金を何倍かにしたいとき、その利子は幾らであるかを求める関数の仕様と、そのニュートン法による解。関数型プログラミング技法を使った、再利用できる関数を含んだ、汎用的な仕様の例。この例題は、Real.vpp などの SSLib のソースから、ニュートン法関連だけを抽出したものである。

1.0.1 責任

「利子を得る」関数の陰仕様と、ニュートン法による陽仕様。

1.0.2 注釈

「導関数」や「ニュートン法で方程式を解く」操作は、関数を引数として渡す高階関数機能を利用している。

class $Newton$

values

1.0 $Rcsid$ = "$Id : Newton.vpp, v 1.8 2005/06/15 08 : 01 : 00 shin Exp $";

2.0 誤差 $= 0$;

3.0 変分 $= 0.00001$

functions

public static

4.0 $= : \mathbb{R} \to \mathbb{R} \to \mathbb{B}$

.1 $= (r1)(r2) \triangleq$

.2 abs $(r1 - r2) <$ 誤差;

5.0 利子を得る陰仕様-数学版 : $\mathbb{Q} \times \mathbb{Z} \to \mathbb{Q}$

.1 利子を得る陰仕様-数学版 (倍数, 年) $\triangleq$

.2 is not yet specified

.3 pre 倍数 $> 1 \wedge$ 年 > 0

.4 post 倍数 $> 1 \wedge$ 年 $> 0 \wedge$

.5 $\exists!$ 利子 : $\mathbb{Q} \cdot$

.6 let 元利合計 = 元利合計倍数を得る (利子, 年) in

.7 倍数 = 元利合計 $\wedge$ $RESULT$ = 利子 ;

6.0 利子を得る陰仕様-計算機版 : $\mathbb{R} \times \mathbb{Z} \to \mathbb{R}$

.1 利子を得る陰仕様-計算機版 (倍数, 年数) $\triangleq$

.2 is not yet specified

.3 pre 倍数 $> 1 \wedge$ 年数 > 0

.4 post 倍数 $> 1 \wedge$ 年数 $> 0 \wedge$

.5 $RESULT \in$ {利子 | 利子 : $\mathbb{R} \cdot$

.6 let 元利合計 = 元利合計倍数を得る (利子, 年数) in

.7 = (倍数) (元利合計)} ;

利子を得る陽仕様（ニュートン法を利用、関数プログラミング使用）

public static

7.0 利子を得る : $\mathbb{R} \times \mathbb{Z} \to \mathbb{R}$

.1 利子を得る (倍数, 年数) $\triangleq$

.2 let $f = \lambda$ 利子 : $\mathbb{R} \cdot$ 倍数 $-$ 元利合計倍数を得る (利子, 年数) in

.3 ニュートン法で方程式を解く (f) (0);

public static

8.0 元利合計倍数を得る : $\mathbb{R} \times \mathbb{Z} \to \mathbb{R}$

.1 元利合計倍数を得る (利子, 年) $\triangleq$

.2 $(1 +$ 利子$) \uparrow$ 年

.3 pre 利子 $\geq 0 \wedge$ 年 > 0 ;

平方根を求める陰仕様 _ 数学版

public static

9.0 root 陰仕様数学版 : $\mathbb{Q} \to \mathbb{Q}$

.1 root 陰仕様数学版 $(x) \triangleq$

.2 is not yet specified

.3 pre $x \geq 0$

.4 post $\exists!$ 平方根 : $\mathbb{Q} \cdot$ 平方根 $\uparrow 2 = x \wedge$ 平方根 $= \mathit{RESULT}$;

平方根を求める陰仕様 _ 計算機版

public static

10.0 root 陰仕様 : $\mathbb{R} \to \mathbb{R}$

.1 root 陰仕様 $(x) \triangleq$

.2 is not yet specified

.3 pre $x \geq 0$

.4 post $\mathit{RESULT} \in \{$平方根 $|$ 平方根 : $\mathbb{R} \cdot = ($平方根 $\uparrow 2)\,(x)\}$;

平方根を求める陽仕様版

public static

11.0 $\mathit{root} : \mathbb{R} \to \mathbb{R}$

.1 $\mathit{root}\,(x) \triangleq$

.2 let $f = \lambda\, y : \mathbb{R} \cdot y \uparrow 2 - x$ in

.3 ニュートン法で方程式を解く $(f)\,(x)$

.4 pre $x \geq 0$;

方程式の解法のニュートン法

public static

12.0 ニュートン法で方程式を解く : $(\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to \mathbb{R}$

.1 ニュートン法で方程式を解く $(f)(x) \triangleq$

.2 let 終了条件 $= \lambda\, y : \mathbb{R} \cdot$ abs $(f\,(y)) <$ 誤差,

.3 次の近似 $= \lambda\, y : \mathbb{R} \cdot y - (f\,(y)/$導関数 $(f)\,(y))$ in

.4 $\mathit{Funtil}[\mathbb{R}]$ (終了条件) (次の近似) (x);

public static

13.0 導関数 : $(\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to \mathbb{R}$

.1 導関数 $(f)(x) \triangleq$

.2 $(f\,(x +$ 変分$) - f\,(x))/$変分;

ある条件 p が真になるまで、初期値 x に関数 f を繰り返し適用する。

public static

14.0 Funtil[@型] : (@型 $\to \mathbb{B}) \to ($@型 $\to$ @型$) \to$ @型 $\to$ @型

.1 $\mathit{Funtil}\,(p)(f)(x) \triangleq$

.2 if $p\,(x)$

.3 then x

.4 else Funtil[@型] $(p)\,(f)\,(f\,(x))$

end *Newton*

Test Suite : vdm.tc

Class : Newton

Name	**#Calls**	**Coverage**
Newton‘=	0	0%
Newton‘導関数	5	$\surd$
Newton‘root	0	0%
Newton‘利子を得る	1	$\surd$

Name	**#Calls**	**Coverage**
Newton‘Funtil	6	$\surd$
Newton‘root 陰仕様	0	0%
Newton‘元利合計倍数を得る	21	46%
Newton‘root 陰仕様数学版	0	0%
Newton‘利子を得る陰仕様-数学版	0	0%
Newton‘ニュートン法で方程式を解く	1	$\surd$
Newton‘利子を得る陰仕様-計算機版	0	0%
Total Coverage		**39%**

class $NewtonT$ is subclass of $TestDriver$

functions

$tests : () \to TestCase^*$

$tests\,() \triangleq$

[

new $NewtonT01\,()$]

end $NewtonT$

class $NewtonT01$ is subclass of $TestCase$

values

誤差 $= 10 \uparrow -10$

operations

protected

$test : () \xrightarrow{o} \mathbb{B}$

$test\,() \triangleq$

let $r =$ new $Newton\,()$ in

(dcl $w : \mathbb{R}$;

return (r.利子を得る $(2, 10) - 0.071773 <$ 誤差)

);

protected

準備する : $() \xrightarrow{o} ()$

準備する $() \triangleq$

テスト名 := "$NewtonT01 : \backslash t$「利子を得る」を検査する";

protected

後始末する : $() \xrightarrow{o} ()$

後始末する $() \triangleq$

return

end $NewtonT01$

参考文献

[1] Cliff Jones： Systematic Software Development using VDM, Prentice Hall International(1990)

[2] CSK： The VDM++ Language V6.8.5, CSK(2005)

[3] 佐原伸：事務システムにおける形式仕様適用例, ソフトウェア技術者協会, ソフトウェア・シンポジウム (2001)

[4] 佐原伸：大規模事務処理システムにおける形式手法の適用経験, ソフトウェア技術者協会, ソフトウェア・シンポジウム (2003)

[5] 佐原伸：VDM++基本ライブラリの作成, ソフトウェア技術者協会, ソフトウェア・シンポジウム (2004)

[6] ジョン・フィッツジェラルド，ピーター・ゴーム・ラーセン著, 荒木啓二郎，張漢明，荻野隆彦，佐原伸，染谷誠 訳：ソフトウェア開発のモデル化技法, 岩波書店 (2003)