

Swift言語を用いた 関数型プログラミングの 学習支援環境

宮城大学 事業構想学研究科 博士前期課程 情報デザイン領域

青木唯一

指導教員

須栗裕樹

目次

研究目的・背景

システム開発について

実験および評価

結論



研究背景


関数型言語とは...

「関数」を組み合わせるプログラミングを行う言語
(関数型プログラミングを行うに適した仕様の言語)

「関数」= 数学的な意味での関数
参照透過性があり、副作用がない

参照透過性	関数の値が関数外の文脈により変化しない
副作用	関数を値を求める過程で他の文脈に影響・変化を与える

関数型言語は(関数記述が正しければ)バグの発生が抑えられる優秀な言語



研究背景

実際に使われている事例

Twitter

LinkedIn

Manaba

SCAWAR

研究背景

しかし実際に関数型言語を知り、学ぶ人は少ない
プログラマーが利用している言語についての調査SocioPLT

調査結果

関数型言語になじみのあるプログラマーの割合	22%
C/C++ Java C#になじみのあるプログラマーの割合	94%
Perl Python Rubyになじみのあるプログラマーの割合	79%

教育機関に絞っても関数型言語を「学習済み」とした人は40%



研究背景

関数型言語の抱える問題

利用者が少ないため、開発環境も少ない




企業や独立開発者が開発言語に選択しない



利用者が増えない


といった悪循環が存在する

問題は「関数型プログラミングは学習する機会が少なく、学習する者も少ない」



研究目的

「関数型プログラミングを学習する機会が少なく、学習する者も少ない」を解決するための3つの点

1. 関数型言語に慣れていない人でも関数型プログラミングを学ぶことができる
 2. OSに依らず、容易に利用することができる
 3. 利用者が一人で学習を進めることができる
- 

研究目的

関数型言語に慣れていない人でも関数型プログラミングを学ぶことができる言語

Swift言語を用いた学習

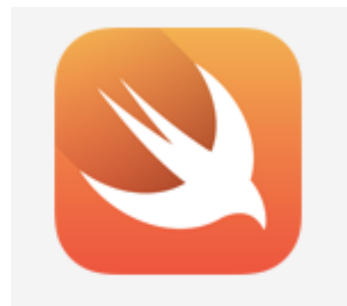
Haskellの仕様から影響を受けた言語

従来型の言語の仕様も残す

手続き型言語になれた人にとっては

純粋な関数型言語を用いて学ぶよりも

学びやすい言語



研究目的

OSに依らず容易に学習できるシステム

問題: SwiftはMac専用言語



Webアプリケーションとして設計



研究目的

利用者が一人で学習を進めることができる

Swift言語仕様の文法学習コンテンツ

+

関数型プログラミングの学習コンテンツ

先行研究 プログラミングの学習

河野敏行らによる「eラーニングによるプログラミング学習システムの開発」

ブラウザを用いたコードのエディタモード

講義者がエディタに入力する内容を配信できる講義配信モード

講師が学習者の入力した内容をモニター、対応できる個別対応モード

課題を提出する課題提出モード

といった機能をもった講義サポートシステム

先行研究 プログラミングの学習

先行研究では変数名やインデントなどの些細な違いに対応できない

本システムはコンパイラを通すため、学習者が自由にコーディングが可能となる

システム開発 概要

ログイン・ログアウト機能

学習テキスト表示機能

Swiftでのコーディング・実行結果表示機能

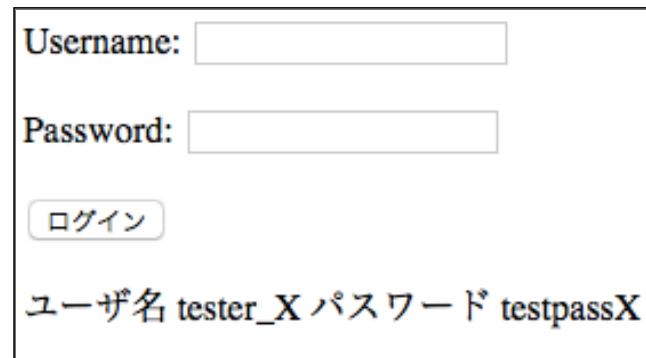
システム開発 システムの機能

ログイン機能

ログイン機能

セッションを生成する。

生成したセッションIDは他の機能で編集するファイルのファイル名管理に使用する。



Username:

Password:

ユーザ名 tester_X パスワード testpassX

生成したセッションはメイン画面のログアウトボタンを押すorブラウザを閉じることで消える。
＝システム利用終了

ログインでメイン画面へ

システム開発 システムの機能 メイン画面



システム開発 システムの機能

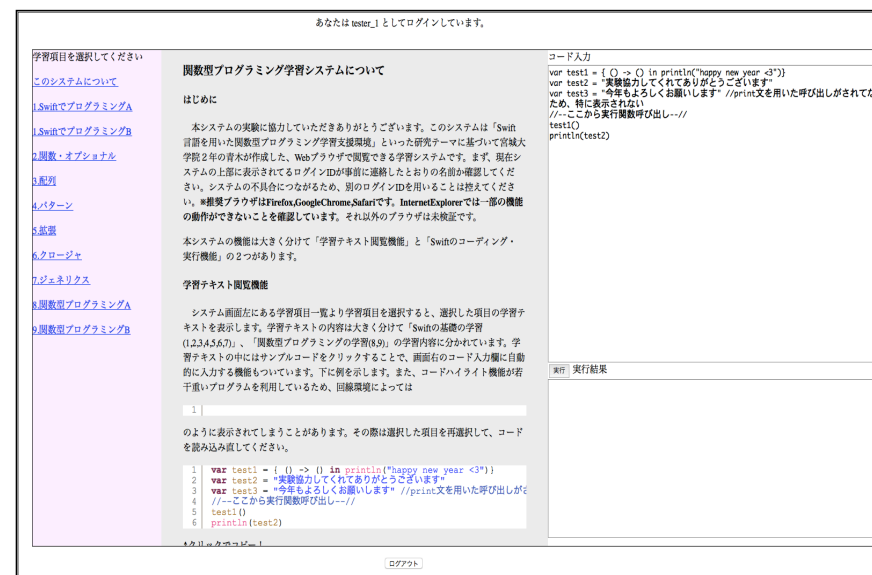
学習テキスト表示機能

学習テキスト表示機能

- (1) Swiftを使ったプログラミングの基礎学習(言語の文法の習得)
- (2) 高階関数を用いた関数型プログラミングの学習(関数型スタイルの習得)

学習項目を選択することで学習テキストを表示する

学習補助のため、コードを選択することで
コード入力テキストエリアにサンプルコードをコピー
実行結果の確認や書き換えが容易



システム開発 システムの機能 コーディング機能

○ Swiftでのコーディング・動作確認機能

コード入力テキストエリアでコーディング
実行ボタンで実行結果表示テキストエリアに
結果を表示

The screenshot shows a web application interface for Swift coding and execution. At the top, it says "あなたは tester_1 としてログインしています。" (You are logged in as tester_1).

学習項目を選択してください

[このシステムについて](#)

- 1. SwiftでプログラミングA
- 2. 関数・オブジェクト
- 3. 配列
- 4. パターン
- 5. 拡張
- 6. クロージャ
- 7. ジェネリクス
- 8. 関数型プログラミングA
- 9. 関数型プログラミングB

コーディング機能

※推奨ブラウザはFirefox, Google Chrome, Safariです。Internet Explorerでは一部の機能の動作ができないことを確認しています。それ以外のブラウザは未検証です。

本システムの機能は大きく分けて「学習テキスト閲覧機能」と「Swiftのコーディング・実行機能」の2つがあります。

学習テキスト閲覧機能

システム画面左にある学習項目一覧より学習項目を選択すると、選択した項目の学習テキストを表示します。学習テキストの内容は大きく分けて「Swiftの基礎の学習(1,2,3,4,5,6,7)」、「関数型プログラミングの学習(8,9)」の学習内容に分かれています。学習テキストの中にはサンプルコードをクリックすることで、画面右のコード入力欄に自動的に入力する機能もついています。下に例を示します。また、コードハイライト機能が若干重いプログラムを利用しているため、回線環境によっては

```
1
```

のように表示されてしまうことがあります。その際は選択した項目を再選択して、コードを読み込み直してください。

```
1 var test1 = { () -> () in println("happy new year <3") }
2 var test2 = "実験協力してくれてありがとうございます"
3 var test3 = "今年もよろしくお祈りします" //print文を用いた呼び出しがないため、特に表示されない
4 //---ここから実行関数呼び出し---//
5 test1()
6 println(test2)
```

↑クリックでコピー!

Swiftのコーディング・実行機能

学習テキストにあるサンプルコードを実行、または自分自身でコーディングを行うことができる機能です。「コード入力」テキストエリアにSwiftでのプログラミングを行い、書いたコードを「実行」ボタンで実行することができます。実行結果（エラー出力も含む）は画面右下の「実行結果」テキストエリアに表示されますが、画面出力を確認するためにはprint文を利用する必要があります。print文については「1.SwiftでプログラミングB」にて説明します。

コード入力

```
var test1 = { () -> () in println("happy new year <3") }
var test2 = "実験協力してくれてありがとうございます"
var test3 = "今年もよろしくお祈りします" //print文を用いた呼び出しがないため、特に表示されない
//---ここから実行関数呼び出し---//
test1()
println(test2)
```

実行 実行結果

```
happy new year <3
実験協力してくれてありがとうございます
```

ログアウト

システム開発 動作環境

ユーザ

- ・宮城大学イントラネット接続またはインターネット接続
- ・JavaScriptの動作が可能なブラウザ

サーバ

- ハードウェア Mac mini
- ソフトウェア OS X 10.10.4
 Server.app バージョン4.1.3
 Xcode バージョン6.4
 Swift バージョン1.2 (バージョン2から文法が大きく変わったため古いまま)
- ネットワーク 宮城大学のStudent VLANに有線で接続する. 学外からのインターネットアクセスはVPN

システム開発 実装

Webページ PHP, HTML/CSS, JavaScript

JavaScript jQueryおよびSyntaxHighlighterを利用



システム開発 実装 コード実行の流れ

ユーザが実行ボタンを選択する



コード入力領域のテキストエリアに入力された内容を
セッション名と関連付けられたSwiftファイルに上書きする



関連付けられたSwiftファイル名をSwiftコンパイラに渡すシェルコマンドをPHPにより実行する



実行した結果をテキストファイルに書き込む



実行結果が書き込まれたテキストファイルを読み込み、
内容を出力結果表示領域のテキストエリアに表示する

関数型学習のコンテンツと練習問題

関数型プログラミングの学習

S式のパーサを作成する学習コンテンツ

- ・アトム型とリスト型とを定義
- ・S式を定義
- ・基本型のパーサとパーサコンビネータを定義
- ・指定した文字をパースするパーサを定義(課題1)
- ・S式をパースするパーサを定義(課題2)

```
<Sexp> ::= <List> | <Atom>
<List> ::= '(' <Sexp>* ')'
<Atom> ::= <Symbol> | <Number>+
<Symbol> ::= '+' | '-' | '*' | '/'
<Number> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
```

評価実験 評価実験の目的

本学習システムを用いた学習方法がテキストのみを用いた学習より優れたものであることを実証実験で確かめる

テキストを用いた学習よりも有用である場合、学習環境として十分有用な方法である

評価実験 評価実験の方法

協力してくれるテストを

- ・本研究で実装したシステム上で学習するグループ
- ・本研究で利用する学習テキストのみを用いて学習するグループ

にわけて空いた時間に学習してもらう

今回はテスト8人 実験期間3日(3連休・・・テストの人たちごめんなさい)で実験
学年、学科、プログラミング能力が均等になるようにグループ分け

評価実験 評価実験の方法

実験後アンケート

学習テキスト内の課題プログラムの達成度

学習テキストの理解度をアンケート結果より考察する

評価実験 実験結果

アンケート結果

番号	設問	グループ A	グループ B
1	利用したシステムは？	Web ブラウザ	PDF テキストのみ
2	学習時間の平均	5 時間	4 時間
3	関数型も含めてプログラム経験あり	0 人	0 人
	関数型以外ならプログラム経験あり	3 人	3 人
	プログラミング経験無し	1 人	1 人
4	Swift でプログラムできるようになった	2 人	0 人
	Swift のプログラムが読めるようになった	2 人	3 人
	Swift がわからなかった	0 人	1 人
6	学習意欲がわいた	4 人	2 人
	湧かなかった	0 人	2 人

評価実験 実験結果

アンケート結果

関数型プログラミングについて

番号	設問	グループ A	グループ B
5	カーリー化を用いたコードが作れた	1人	0人
	カーリー化が理解できた	2人	0人
	カーリー化がわからなかった	1人	4人
7	パーサ問題 1問目と2問目ができた	1人	0人
	パーサ問題 1問目だけできた	2人	0人
	パーサ問題 できなかった	1人	4人

評価実験 考察

本システムは関数プログラミングを学習する効果はテキストよりも高い効果が得られる

特にテキストに現れるサンプルコードを即座に実行することが学習に大きな効果を与えることができた

学習テキストとコーディング機能を一体化させたシステムは学習に効果的である

評価実験 考察

Webブラウザを用いた学習は不特定多数の人が関数型プログラミングの学習を始めるのに有用な効果がある

学習意欲がテキストだけよりも高くなる効果もある

今後の人材育成に役に立てる可能性がある

ただし、より正確に測るため、実験規模を大きくする必要有

結論

関数型プログラミングは従来の言語にはない多くの利点を持つ

しかし利用者・学習者が少ないといった問題がある

従来型のプログラミングを行え、かつ純粋関数型言語Haskellから大きな影響を受けている言語Swiftを用いた。

学習支援環境の形態についてはWebブラウザを用いたシステムを提案

テキスト・実行環境の一体型システムでの学習はテキストのみを用いた学習よりも有用である

このようなプログラミング学習環境が一般的に普及すれば問題は解決されるだろう

ありがとうございました

