

## Javaによる自己表現能力の育成

齋藤 俊則

tsaito@crew.sfc.keio.ac.jp

慶應義塾大学大学院政策・メディア研究科

中鉢 欣秀

yc@crew.sfc.keio.ac.jp

慶應義塾大学大学院政策・メディア研究科

脇 佳代子

t95009kw@sfc.keio.ac.jp

慶應義塾大学環境情報学部

大岩 元

ohiwa@sfc.keio.ac.jp

慶應義塾大学環境情報学部

### 抄録

我々はJava言語を用いた初学者向けプログラミング講座のカリキュラムを設計した。このカリキュラムは実際のソフトウェアの開発工程に沿って設計されている。それによって学習者は学習する技術や概念が現実的にどのような場面で有効になるのかを理解・納得しながら学習を進めることができる。またそれらをコンピュータ上での問題解決のための表現技法としてとらえながら学習を進めることができる。このカリキュラムは慶應義塾大学大学院政策・メディア研究科修士課程で現在実施されている。

## A Programming Course of Java Language to Bring Up Ability of Self Expression

### Abstract

We developed a course of programming using Java language for beginners. This course has been developed in accordance with the process of software development. Therefore it enables the learners to realize the effectiveness of techniques and concepts they learn in the real occasion of software development. At the same time, the learners can learn them as a methodology of expressing themselves in a problem-solving activity using computers. This course is now put into practice in the Graduate School of Media and Governance of Keio University.

### 1. はじめに

初学者向けのプログラミング教育の多くはボトムアップ方式で行われる。ここでいうボトムアップ方式とは「変数への代入」、「場合分けの仕方」等プログラミングを行う際に頻出する技術や概念を学習項目として並置し、比較的単純と思われるものから順番に一つずつ学習させてゆくやり方

である。このやり方の長所は、単純なものからより複雑なものへと学習項目を積み上げてゆくことで比較的無理なく効率的にプログラミング技術を学ばせることができる点にある。一方短所として、学習者の立場からは学習過程の全体像が見渡しにくいために、現在学んでいる技術・概念が最終的にどのようにプログラミングの中で役立つものなのかがわかりにくいという点があげられる。この方式はあらかじめ技術の習得という特化された目的意識をもつ学習者に対しては有効であると考えられるが、より一般的に非技術系専攻者にたいする情報教育としてのプログラミング教育を考えた場合には学習者のモチベーションの低下などの問題が考えられる。

我々は慶應義塾大学大学院政策・メディア研究科の初学者向けプログラミング講座カリキュラムを設計した。この講座ではカリキュラムにある程度トップダウンな観点を導入することを試みた。そして学習内容を「プログラミング技術の習得」としてよりはむしろ「コンピュータ上での自己表現能力の育成」としてとらえられるように工夫した。具体的には最終的な到達目標として、習得させたい技術・概念をすべて含んだ一つの課題プログラムをあらかじめ用意し学習者にたいして提示する。全カリキュラムはこのプログラムを完成させる過程の追体験として設計されている。それぞれの技術や概念はプログラム完成の過程で必要となったときにその都度学習させる。このようにカリキュラムを設計することで学習者はいつでも学習内容を学習過程全体の中で位置づけることが可能となる。また学習する技術や概念が現実的にどのような場面で有効となるのかを理解・納得したうえで、それらをコンピュータ上での問題解決のための表現技法として学ぶことが可能となる。

## 2. 講座の位置づけ

当研究科修士課程では、主にコンピュータ技術の活用技法を教育する「マルチメディア表現法」という講座が設置されている。これは研究活動を行う際に分野を問わず必要となるであろう技法の習得を支援する「技法科目」のうちの一つである。殆どの修士課程在籍者はいくつかの「技法科目」のうち最低1つを選択履修する。我々が設計したカリキュラムはこの「マルチメディア表現法」の中で1998年度春期から実施されている。

この講座は当初、初学者及び技術系を志望しない学生の履修を想定して計画された。しかしながら実際の履修者の中にはプログラミング経験者が多く含まれていた。そこで講座の中で自己申告により初学者と経験者とを分けて、初学者に対してのみこのカリキュラムを実施することにした。また慶應義塾大学湘南藤沢キャンパス（以下SFCと略）ではコンピュータの標準環境がUNIXベースであるため、外部からの入学者でUNIX未経験者に対しては講座の時間外にUNIXの基本的な利用法を中心とするリテラシー講座を行うことにした。結果として「マルチメディア表現法」は事実上SFCの修士課程における「情報教育講座」となっている。

## 3. 講座の概要

この講座の最終的な教育目標は、コンピュータにさせたい仕事をプログラミング言語で適切に表現できるようにすることである。一般的にコンピュータ技術者以外の人間はコンピュータを道具として使えばよいといわれる。ただし特定のワープロや表計算のソフトの使い方を覚えても、それらの知識はバージョンアップ等によってすぐに陳腐化してしまう。教育の一環としてコンピュータを扱う場合には、できるだけ普遍性を持つ応用可能な能力を身につけさせる方が望ましい。そのような観点からコンピュータを道具として利用する際に必要となる能力を考えるならば、自分の要求をコンピュータに理解できるように整理・定型化して表現する能力の育成がもっとも必要であるといえる。これは人間のあいまいな要求をそのまま受け付けられないコンピュータを主体的に活用するために欠かせない能力である。そしてそのような能力を育成するためには、コンピュータの動作を原理的に学習する必要がある。このような学習を可能にするために、プログラミングはもっともよい教材であると我々は考えている。

この講座の具体的な教育課題は、クライアント・サーバモデルに則った簡単な情報システムを構築することである。今回はユーザーに対してネットワークを介して任意の授業の時限、教室番号等の情報を提供する「時間割表示システム」の完成を目標に据えた。このような課題を選んだ理由は、今後の社会におけるコンピュータの利用形態の中心がスタンドアローンからネットワークユースへ移行してゆくことを鑑みた際に、ネットワーク利用を前提としたコンピュータ上での情報のやりとりの原理を利用者の側も学ぶ必要があると判断したためである。今後は技術者以外の人間にもいかにコンピュータを活用するかのアイデアを持つことが求められると予想される。そのようなアイデアを持つためには、ネットワークを利用した情報通信の仕組みを含めてコンピュータの動作原理を理解しておく必要があると考えられる。

講座で使用する言語はJava言語である。あえてJava言語を選んだ理由は、第一にネットワークプログラミングが他の言語と比べ非常に容易である点と、第二に手続き型言語と比べてオブジェクト指向であるJava言語はシステムの分析からトップダウンにカリキュラムを進めやすいという点がある。なお、しばしばJava言語の初心者向けテキストはアプレットの作成が中心であることが多いが、アプレットがブラウザと連携して動作する機構は初心者にとって非常に難解である。その機構をブラックボックスにしてアプレットの作成技術だけを表面的に学ぶことはこの講座の教育目標に反することになる。またアプリケーションの作成を通してJava言語プログラミングの基礎を学ぶことで、アプレットの作成を自習するための素養を得ることは十分である。したがってこの講座ではアプレットを題材にするのではなくJava言語によるアプリケーションの作成の基本のみを扱うことにした。

この講座のカリキュラムは次の通りである。毎週90分で全12週である。

- 第1回： 情報を共有する仕組みとしてのクライアント・サーバモデルの理解
- 第2回： 「時間割表示システム」全体像の俯瞰、そこに実現される機能の理解
- 第3回： 時間割を構成するデータの定義、Java言語による定義の書式
- 第4回： 授業データを表示するプログラムの作成
- 第5回： 授業データをキーボードから入力させる仕組みの実現
- 第6回： 授業データをファイルに保存させる仕組みと、ファイルから読み込む仕組みの実現
- 第7回： 一日分の授業データを扱う仕組みの実現
- 第8回： 一日分の授業データをファイルに保存させる仕組みと、ファイルから読み込む仕組みの実現
- 第9回： 曜日を手がかりに表示すべき時間割を選び表示するプログラムの作成（ローカル版「時間割表示システム」）
- 第10回： 「クライアント・サーバ」という考え方と、それを実現する簡単なプログラムの作成
- 第11回： 「時間割表示システム」サーバプログラムの作成
- 第12回： 「時間割表示システム」クライアントプログラムの作成、クライアント・サーバ版「時間割表示システム」の完成

この講座は全体的にはソフトウェア工学における開発工程（分析→設計→作成→評価）に沿って進められる。すなわち第1、2回目で課題となる「時間割表示システム」の仕様を「分析」し、システムにどのような機能が求められているのかをあらかじめ把握させる。第3回以降は毎時間ごとに前週に作成したプログラムを部分的に拡張しながら進める。各回ごとに前回のプログラムを「評価」し、拡張する部分を含めてその日の目標となるプログラムを「設計」し、実際に「作成」を行う。その際主に「設計」の部分は担当者が講義の中で理解させ、「作成」の部分は各履修者に課題として実習させる。このプロセスを繰り返すことで、最終的に第2回で明らかにしたシステムの仕様を満たすプログラムを完成させる。

できる限りトップダウンに開発工程に沿ったカリキュラムを実現するため、全12回で必要となるサンプル・プログラムや履修者に要求すべき課題の内容、そして各回の学習事項の中心となるキーワード群は事前に決定した。カリキュラムを実施する中で調整すべき点は当然でてくると考えられるが、そのような場合でも元になるカリキュラムの設計が明確であれば履修者の混乱も最小限ですむ。それは履修者のモチベーションを考えると非常に重要なことである。

以下は毎回の授業内容の概要と、キーワード群、そしてその回に履修者に科す課題である。これらの情報は履修者たちに授業シラバスとして配布した内容と同じものである。

## 第1回

情報を共有する仕組みとしてのクライアント・サーバモデルの理解

概要：

社会生活をおこなってゆく上では、様々な局面で他者との情報の共有が不可欠となる。コンピュータ・メディアにおいては不特定多数の人々における情報の共有を、「クライアント・サーバモデル」とよばれる仕組みを導入することで、より効果的・効率的に実現した。World Wide Webはこの仕組みを導入した、最も身近な例の一つである。そこで今回は、World Wide Webの例をとりながら、「クライアント・サーバモデル」という概念がどのようなものであるのか、そしてそれが情報の共有に対してどのような点で優れているのかを学んでゆく。

キーワード：

情報の共有、クライアント・サーバモデル、World Wide Web

課題：

キャンパス・ライフにおいて学生の間で共有すべき情報としてどのようなものがあげられるかを考察せよ。

## 第2回

「時間割表示システム」全体像の俯瞰、そこに実現される機能の理解

概要：

この講座では、必要なときにその日の時間割を提示してくれる「時間割表示システム」の完成を目標にしながら、プログラミングの基礎とクライアント・サーバモデルの理解を深めてゆく。まずはその目標となる「時間割表示システム」の完成例を実際にユーザーの立場から利用することで、それがどのように機能するシステムであるのかを知る。つぎに、その機能がどのような仕組みで実現されているのか、その全体像を解説する。そして最後に、ここで当面目標にすべき「時間割表示システム」に求められる機能の概要を整理・確認する。

キーワード：

時間割システム、時間割、データ、機能、仕様の定義、設計

課題：

この「時間割システム」において扱われるべき「データ」にはどのようなものがあるのか。また、どのような「機能」が実現されるべきなのか。その2点を整理せよ。

## 第3回

時間割を構成するデータの定義、Java言語による定義の書式

概要：

プログラミングとは、「このように機能せよ」といったコンピュータに対する命令をプログラミング言語で記述することである。この講座ではJava言語と呼ばれるプログラミング言語を用いてこれをおこなう。プログラミング言語にはそれぞれ決まった書式があり、プログラムを作成するにはかならずそれに則る必要がある。したがって、まずはJava言語における最低限踏まえないといけない書式を知ることからプログラムの作成を開始する。具体的には、時間割を構成するデータとしては何が必要かを定義し、そこで定義された項目（例えば「授業名」とか）をプログラムとして記述するやりかたを学習する。

キーワード：

クラス、クラスの属性、属性の型

課題：

時間割を構成する各項目を正しく記述したファイルJugyou.javaを作成せよ。

## 第4回

授業データを表示するプログラムの作成

概要：

前回定義したデータの各項目には、実際には具体的な数字や文字が与えられることが想定されている。それらの数字や文字はコンピュータが実行する何らかの機能の中で利用されることになる。ただし、前回作成したJugyou.javaというファイルには、コンピュータにどのように機能してほしいか、という部分の指示はまだなにも書かれていない。コンピュータを自分が望むように機能させるためには、その機能をあらかじめ適切にプログラム中に表現しておかなければならない。今回は、具体的な数字や文字の与えかたと、実行してほしい機能の表現の仕方を学習する。具体的には、授業のデータを実際にコンピュータに記憶させ、それを画面に表示する、というプログラムを作成する。

キーワード：

mainメソッド、クラスとオブジェクト、属性への値の代入、標準出力

課題：

前回作成したJugyou.javaを、時限、科目名、教室番号を順に画面に表示するプログラムに書き換えよ。

## 第5回

授業データをキーボードから入力させる仕組みの実現

概要：

今回は標準出力を利用してデータの画面への表示をおこなったが、その際データはあらかじめプログラム中に書き込んでおいた。今回は、それぞれのデータをその都度ユーザーがキーボードから入力するようにプログラムを改作する。そのために必要な「データの入出力」という考え方と、それを実現するために必要な仕組みを中心に学習する。また、改作の結果複雑で見通しの悪くなったmainメソッド内部を分かりやすくするために、mainメソッドを記述する前にあらかじめ各機能を「メソッド」として定義しておくやり方も学習する。

キーワード：

データの入出力、標準入力、メソッド、例外、文字列から数字への変換

課題：

キーボードから入力された時限、科目名、教室番号を画面に表示するプログラムを作成せよ。その際、データの入力の部分と画面へ表示させる部分はメソッドとしてあらかじめ定義せよ。

## 第6回

授業データをファイルに保存させる仕組みと、ファイルから読み込む仕組みの実現

概要：

今まで試作してきたプログラムでは、授業データはユーザー自身がキーボードから入力するようになっていた。これに対して我々が目標とする「時間割表示システム」とは、時間割を知らないユーザーに対して情報をあたえるものでなければならない。そのためには、授業データをあらかじめコンピュータに記憶させておく必要がある。そのやり方としてプログラムの書かれたファイルにデータを直接書き込んでおく方法も考えられるが、時間割を書き換える場合などを考えるといろいろなデメリットがあげられる。そこで今回は「データの入出力」を発展させたよりスマートな解決法として、授業データだけをプログラムとは別のファイルに保存し、プログラムが機能する上で必要になったときにそのファイルのデータを利用するやり方を学習する。

キーワード：

データのファイルからの読み込み、ファイルへの保存、ストリーム

課題：

Jugyou.javaを改良して授業データをファイルに保存・読み込みする機能をあたえよ。

## 第7回

一日分の授業データを扱う仕組みの実現

概要：

これまでは1回分の授業に関するデータをあつかうプログラムを作成してきたが、「時間割」にするためには1日におこなわれるすべての授業についての情報を扱う必要がある。そのためには、一日分の授業データを扱えるように、データの扱い方や表示の仕方をかえる必要が出てくる。具体的には、複数のデータを一括して扱う仕組みの導入と、同じ命令を指定する回数だけ繰り返しおこなう仕組みの導入が考えられる。そこで今回はこれらの仕組みを導入して、一日分の授業データをファイルに保存し、保存された全てのデータを表示するようにプログラムを改良する。

キーワード：

ベクター、繰り返し文

課題：

Jugyou.javaを一日分の授業データを扱えるように改良せよ。

## 第8回

一日分の授業データをファイルに保存させる仕組みと、ファイルから読み込む仕組みの実現

概要：

これまで学習してきたとおり、プログラミングにおいては様々なデータをいくつかの「型」に分類し、それぞれを区別して扱う習慣がある。しかしながら、データの入出力に関しJava言語では、全ての型のデータを一括して扱うための方法が用意されている。我々はこの講座において主に文字列をあらわす「String型」を中心にデータの入出力を扱ってきたが、この方法を学習しておくことでその他の型のデータを扱う必要性が生じた際にも対処が容易になり、またプログラム自体も簡潔にすることができる。

キーワード：

オブジェクトの入出力、シリアライズ

課題：

Jugyou.javaをオブジェクトの入出力を前提とするプログラムへと改良せよ

## 第9回

曜日を手がかりに表示すべき時間割を選び表示するプログラムの作成  
(ローカル版「時間割表示システム」)

概要：

今回は、これまでの学習を総括する意味で、単独のコンピュータ上で動く(つまりは、クライアント・サーバモデルを用いない)「時間割表示システム」を作成する。月曜日から金曜日までの5つ時間割を用意し、ユーザーが選択した曜日の時間割を画面に表示するという機能を実現する。今回の新たな学習事項としては一点だけ、複数用意された命令の中から設定された条件に従ってひとつのものを選択し実行する、という仕組みの実現があげられる。

キーワード：

選択文

課題：

月曜日から金曜日までの時間割の登録されたファイルを用意し、Jugyou.javaをユーザーが入力した曜日の時間割を表示するプログラムへと改良せよ。

## 第10回

「クライアント・サーバ」という考え方と、それを実現する簡単なプログラムの作成

概要：

この講座の最終目標である「時間割表示システム」を完成するためには、前回作成したローカル版のシステムをクライアント・サーバを前提とするもの書き換えなければならない。そのためには、クライアント・サーバを実現するコンピュータ・ネットワークに関連するいくつかの概念を理解し、プログラムとして表現する方法を身につける必要がある。そこで今回は、一旦Jugyou.javaのプログラムを離れ、非常に単純な例題プログラムから、ネットワークを利用するプログラミングに必要な最小限の知識・概念を学習する。

キーワード：

ネットワーク、ソケット、ポート、ポート番号

課題：

キーボードから読み込んだ文字列を送信し、サーバから受信した文字列を表示するクライアントのプログラムと、クライアントから受信した文字列をそのままクライアントに送り返すサーバのプログラムを作成せよ。

## 第11回

「時間割表示システム」サーバプログラムの作成

概要：

これまでに作成してきたJugyou.javaを利用して時間割のデータの通信を行う、サーバおよびクライアントのプログラムを作成する。今回はまず、「時間割表示システム」においてサーバ、クライアントがそれぞれどのような機能を持つ必要があるのかを再確認する。そして、前回作成した単純なサーバのプログラムをもとに、必要な変更や機能の追加を行って「時間割表示システム」のサ

一バを完成させる。

キーワード：

特になし

課題：

「時間割表示システム」のサーバ用プログラムを作成せよ。

## 第12回

「時間割表示システム」クライアントプログラムの作成、

クライアント・サーバ版「時間割表示システム」の完成

概要：

前回に引き続き、今回は「時間割表示システム」のクライアント側のプログラムを第11回の例題を参考にしながら作成する。そして、作成した全てのプログラムがひとつの「時間割表示システム」として機能するかどうかを最終確認し、問題がなければシステム完成とする。

キーワード：

特になし

課題：

クライアント用プログラムを作成し、「時間割表示システム」を完成させよ。

## まとめ

コンピュータ上での自己表現能力の育成を主眼とするプログラミング講座のカリキュラムを設計した。その際、特に非技術系専攻の初学者への配慮から、できる限りトップダウンな観点からのカリキュラム設計を試みた。その結果Java言語を用いたごく小規模のネットワークシステムの開発を題材とする、ソフトウェア工学の開発工程に沿った授業内容となった。

今後の課題は以下の通りである。

- 1) 今期の履修者の講座に対する反応、要求を分析する。それに基づきカリキュラムを洗練する。
- 2) 目標のプログラムを洗練する。履修者にとってより興味をそそる題材がまだ他にもあると思われる。
- 3) 学習事項の順序を洗練する。プログラムがより自然に拡張されてゆくような順序がある可能性が残されていると思われる。
- 4) カリキュラム全体をよりソフトウェアの開発工程に近いものにする。もう少しシステムの「分析」や「評価」の部分に時間をかけられると理想的である。
- 5) カリキュラムをテキストの形にまとめる。その過程で内容を洗練する。