

人文情報学科のための Java 入門教材の作  
成について

藤尾早紀

## 目 次

<b>1 卒業制作のテーマ</b>	<b>1</b>
1 テーマ	1
2 対象者	1
3 Java について	2
4 類似物との比較	3
<b>2 制作物について</b>	<b>4</b>
1 制作物の概要	4
2 教材の形態	4
3 デザインとレイアウト	6
4 実際に利用した個々の教材の評価	7
<b>3 制作過程</b>	<b>9</b>
1 制作する上で苦勞した点	9
2 内容の詳細	11
3 利用者のことを考えて工夫した点	15
<b>4 評価</b>	<b>17</b>
1 アンケート結果と考察	18
2 アンケート後の改良点	21
3 自己評価	22
<b>5 まとめ</b>	<b>23</b>
1 今後の課題	23
2 おわりに	25

## 1 卒業制作のテーマ

### (1) テーマ

私が所属している福田ゼミは、「人の役に立つものを作る」という事を基本テーマとしている。人の役に立つものを作るためには、それを利用する相手の立場に立って物事を考えていく必要がある。そのため、テーマを決める際にも、まずは対象者を絞ってからもの作りに取り掛かろうと考えた。相手の立場に立って考えると言っても、実際に私がその相手になれるわけではないので、結局は私の想像できる範囲に考えが偏ってしまうだろう。それならば、私が実際にそれを利用する相手の立場になった経験があり、その経験を生かしたものを作成すれば良いのではないかと考えた。そこで私は、自分と同じ人文情報学科の学生に向けて、Java の入門教材を作成することにした。

### (2) 対象者

テーマでは「人文情報学科のための」としたが、主になるのは福田ゼミ生である。本教材は、福田ゼミの必修授業である「プログラミング演習」を受講し、プログラミングの基礎を身につけた人を対象にしている。その理由は、教材を作成するにあたって、説明文の量や練習問題の難易度などを設定するとき、どうしても基準になるのが自分自身になってしまうからだ。そのため、「プログラミング演習」を受講していない人には、本教材は不向きなものとなるだろう。

Java のプログラミングについては基礎から解説するが、授業で作った事のある BMI 計算機や数当てゲームなどは、問題として出題したいと考えている。その際、Python<sup>(1)</sup>や PHP<sup>(2)</sup>によるプログラミングでこれらを経験していない人には、難問になってしまうだろう。私自身は

プログラミング演習で初めてプログラミングを学んだので、同じ授業を受けた人なら同等の理解力を持って本教材に取り組めるだろうと考えている。

### (3) Java について

入門教材として Java を選んだ理由は、Java にこれまで授業で学んできたプログラミングと異なるいくつかの特徴があるためである。Python や PHP と Java の一番の違いは、Java がオブジェクト指向プログラミング言語であるということだ。オブジェクト指向プログラミングとは、データとそれを操作する手続きをオブジェクトと呼ばれるひとまとまりの単位として一体化し、オブジェクトの組み合わせとしてプログラムを記述するプログラミング技法のことを言う（『IT 用語辞典』(<http://e-words.jp/w/OOP.html>) より引用)。現在ソフトウェアの開発において主流になっているのは、オブジェクト指向プログラミング言語であるとも言われており、Java を学ぶ事がすなわち現在主流のプログラミング手法を学ぶ事にも繋がると言えるだろう。人文情報学科の学生なら、これを学んでおいて損はないだろうと考え、本教材を作成することにした。

Java には用途に合わせた開発セットや API<sup>(3)</sup>が提供されているといった利点がある。図形などの描画も可能で、グラフィカルなプログラミングができるため、GUI<sup>(4)</sup>アプリケーションを作成することもできる。GUI も現在主流のインターフェースであるが、利用する事はあってもプログラムがどんな動きをしているかを想像しながら利用している学生はあまり居ないだろう。利用者と開発者の双方の立場になることで、学生たちの情報処理に対する理解をより深めていって欲しいと思

う。JDK<sup>(5)</sup>の最新バージョンは、JDK 7が2011年7月にリリースされたが、本格的な普及がまだ済んでいないため、本教材ではJDK 6を使用する。

Javaはプラットフォームを選ばず使う事が可能なので、MacやWindowsという事を気にせずにプログラムを書くことができる。福田ゼミの学生は主にMacOSXでプログラミングを行うが、家庭ではWindowsを利用している人が大半だろう。プラットフォームを選ばずに利用できるというJavaの利点を活かせば、家庭でも気軽にプログラミングが学習できる。授業以外でプログラミングをする機会があまり無いため、学生の中には長期休暇中にプログラミング離れしてしまう人も居るだろう。休暇中にこの教材を利用することで、休み明けの授業に復帰しやすくなる。また、教材に授業で学んだ内容を取り入れれば、復習に活用してもらうこともできるだろう。

#### (4) 類似物との比較

Webで検索すればJavaの入門サイトは多数あるものの、教材と呼べるような物は少なかった。参考にした個々の教材については、後の「実際に利用した個々の教材の評価」で具体的に検討する。

入門サイトの多くは、説明文とソースコードを記載してあるだけといった文字ばかりの構成で、画像付きで詳しく解説された物はあまり無い。サンプルプログラムはあっても、その実行結果などが確認できないものも多くある。また、練習問題が用意されておらず、利用者に自分で考えてもらうことがないため、ソースコードをその通りに読んだり実行するだけで終わってしまい、理解して応用に繋げることが難しいのだ。

これらは過去に教材を作成した人の卒業論文で、類似物の欠点として

よく挙げられていた事柄である。利用者としては、画像付きで詳しく説明してあるものの方が見やすく、理解もしやすいようだ。そのため、本教材でも文字と画像で各段階を詳しく解説し、より教材として利用しやすいものを作成したいと考えている。各章を順に進めていけば解くことのできる練習問題を間に用意することで、理解を深めてもらい、応用力も身につけてもらいたいと思う。

## 2 制作物について

### (1) 制作物の概要

本教材は、人文情報学科の学生が「プログラミング演習」を受講し、Python や PHP などプログラミングの基礎を学んだ人が、次のステップとして Java を学ぶ事を想定して作成する。Java の基礎を始めとし、授業で作る BMI 計算機や数当てゲームなどを、Java で再現できるように解説していく。この教材を利用することで、利用者は Java の基礎から GUI アプリケーションの作成までを学習することができる。実際の授業を受けるように、第 1 章から順に進めていき、全ての章が終われば、Java の基礎が身につけているという形を目指す。プログラミング演習で Java を扱う事は無いだろうが、休暇期間中の復習や課題として本教材に取り組んで欲しいと思う。制作過程や各章の内容については、後の「制作過程」で具体的に検討する。

### (2) 教材の形態

本教材は、オンライン教材として作成する。オンライン教材ならば、家庭の PC から閲覧可能であり、休暇中でも利用することができる。家庭で利用されている OS の多くが Windows であることを想定し、画面の

キャプチャー画像は Windows 視点のものをメインに扱う。しかし、制作過程の中でアンケートを実施する環境は MacOSX になるだろう。そのため、MacOSX の場合の方法についても、必要に応じて記述する。

近年の卒業制作物でオンライン教材を作成したものは、XHTML で作成されたものが多いが、本教材は HTML4.01 で作成する。確かに、XHTML には再利用しやすいといったメリットもある。しかし、World Wide Web Consortium (W3C) では XHTML の策定を打ち切り、今後は HTML5 の標準化に注力していくことを発表している。<sup>(6)</sup>つまりどちらの記述にしたとしても、いずれは HTML5 に移行していくことになるだろう。そのため、今回は資料も豊富にあり、多くのブラウザに対応している HTML4.01 を選んだ。

初めは本教材を電子書籍で作成したいと考え、XHTML で記述していた。2011 年 4 月に人文情報学科の学生に iPad が配布されてから、「DTP 演習」などの授業では実際に電子書籍を教材にした授業が既に行われている。そのため、学生にはこの新しい学びの形態に対応して貰いたかった。電子書籍はオフラインでの閲覧が可能のため、電車の中や自宅など、どこでも勉強してもらう事ができるなどの利点が多々ある。しかし、プログラミングは読むだけではなく、実際に自分の手を動かしながら記述しなければ理解する事が難しい。どこでも学ぶ事ができる利点があっても、読むだけではあまりプログラミングの勉強にはならないだろう。オンライン教材には、ファイルをダウンロードしたりソースコードをコピーペーストできるという利点もあるため、正しく動くプログラムを一度見てもらうことができる。家庭での学習のためならオンライン教材で十分で、むしろ利点の方が多く思えたため、今回の形態になった。

### (3) デザインとレイアウト

本教材のデザインには外部 CSS を使用する。CSS とは、Cascading Style Sheets の頭文字を取った略語であり、Web サイトのデザインに用いられる技術である。HTML ファイルから呼び出すだけで簡単に利用することができるため、CSS を外部ファイルにまとめておけば、CSS ファイルを書き換えるだけで、利用しているすべての HTML ファイルのデザインを一度に変更できる。また外部 CSS を用いることによりサイト全体の統一感も出すことができるなどの利点もある。

ページのレイアウトは、メニューエリアとメインエリアに分け、メニューをクリックすれば各章の中身がメインエリアに表示されるというフレームページのような見た目にしたい。しかし、フレームはブラウザによって対応しないことや、Web 検索の際にページの中身だけが表示されてしまうと言った問題が起きるため、一般的にあまり好まれず、W3C でも非推奨となっている。そのため、今回は<DIV>タグを使ってメニューエリアとメインエリアを分割する方法をとる。<DIV>タグはそれ自身は特に意味を持っていないが、<DIV>〜</DIV>で囲んだ範囲をひとかたまりとして、align 属性でコンテンツの位置を指定したり、スタイルシートを適用するのに用いる。まず<DIV>タグで分割したエリアに対して、スタイルシートの「position」プロパティに「fixed」を指定することで、絶対位置へ配置する。これによりメニューエリアを固定することができるため、擬似的にフレームページのような見た目を再現できる。同じ絶対位置の配置には「absolute」があるが、「fixed」はスクロールしても位置が固定されたままという特徴がある。しかし、「fixed」は IE6 には対応していないため、外部 CSS ファイルに IE6 の場合の対応を別途記述することで、この問題を解決する（図 1-1、1-2 参照）。



#### (4) 実際に利用した個々の教材の評価

教材を作成するにあたり、自身の Java の学習には書籍と入門サイトの両方を利用した。主に利用したのは、『よくわかる Java』(鳥海不二夫著、秀和システム、2007)だ。本書は Java の基礎から実践までを、イラストや画像で丁寧に説明した、初心者向けの解説書である。重要な項目がいくつかの章に分けて構成されており、章の終わりには練習問題があるため、教材としても利用できる。説明文の中で、重要単語は太字で強調したり、構文などはポイントとして枠で囲っていたりと、初心者にも理解してもらうための工夫が多く見られる。本書で学習できるのはコマンドプロンプトで表示されるプログラムばかりのため、「今後学ぶべきこと」として GUI アプリケーションの Swing<sup>(7)</sup>などが挙げられていた。

『Java 初心者入門講座』(<http://sunjava.seesaa.net/>)は、基礎から、Eclipse<sup>(8)</sup>を使った本格的なアプリケーション開発までを学習することができる。各ステップごとに画面のキャプチャー画像があり、ソースコードはそのままコピーペーストして利用する事が可能など、オンライン教材の利点を活かしている。しかし、初心者向けとは書いてあるが、説明が簡潔すぎていて、解説を読んでも理解できない事がまれにあり、分からない部分をその都度調べながら学習する必要があった。また、このサイトはブログだったため、タグなどを使ってポイントを強調することができておらず、重要な単語や構文などが分かり辛かった。この他にも、メニュー項目が長く、ページの内容が長い時にはわざわざ上までスクロールを戻す必要があったりと、利用する上で不便な点もあった。

GUI アプリケーションを作成するにあたり、基礎となる AWT<sup>(9)</sup>は『AWT 入門』(<http://wisdom.sakura.ne.jp/system/java/awt/>)で学習した。このサイトは、Web 上で動作する Java アプレットのサンプ

ルプログラムが中心で、実行結果をそのままオンライン上で確認することができるという利点があった。Java アプレットと Java アプリケーションではソースコードや実行方法などが異なるため少し分かり辛かったが、API に含まれる便利なクラスの用例を理解することができた。

より GUI アプリケーションについて学ぶため、『Eclipse 3.6 では始める Java プログラミング入門—Eclipse 3.6 Helios 対応』（掌田津耶乃著、秀和システム、2010）も参考にした。本書では、Eclipse を使って Swing や Java サブレット、携帯 Java プログラムの開発についてをメインに学習できる。Java サブレットと携帯 Java については本教材よりも複雑な話になるため、「Chapter2 Java プログラミングの基本を覚える」と「Chapter3 Swing による GUI プログラミング」を主に学習した。『よくわかる Java』ほど基礎について丁寧に解説していないため、Java の最初の入門書としては向かないかもしれないが、GUI アプリケーションの解説は画像付きで、イメージが掴みやすかった。2010 年に刊行されており、今回活用した資料の中ではもっとも新しいため、Java 7 (JDK 7) についても触れられていたが、本書でも初心者は JDK 6 を利用するよう薦めていた。

実際に自分が基礎から Java を学び、また色んな教材を参考にした事でそれぞれの教材の利点、問題点なども見えてきたように思う。分かりやすい教材のポイントは、説明文の中で重要語句や構文が強調されていること、画像やイラスト付きであること、実行例があることなどが挙げられるだろう。本教材においても、これらの利点を取り入れていきたいと思う。

### 3 制作過程

#### (1) 制作する上で苦労した点

ページのレイアウトやデザインなどは後から変更できるので、重要となる中身の作成から始めることにした。まず悩んだのが、章の構成である。難しくなるにつれて内容が増えていってしまうため、一つの章にどれだけの項目を割り当てるかが難しかった。そこで、教材を基礎編と GUI アプリケーション編の二つに区切り、それぞれの終わりに練習問題として出せるようなプログラムを先に作成する事にした。プログラムを先に作成しておくことで、どんな制御文を使うかや、必要な関数などが分かるため、そのプログラム作るために何を学習すれば良いかが把握できるのだ。

基礎編の練習問題には、プログラミング演習でも経験のある BMI 値を計算するプログラムを選んだ。これは身長と体重の値を元にしてプログラムが計算を行い、BMI 値と BMI 値の判定結果を返すという簡単なプログラムである。プログラミング演習を受講した人ならば、Python や PHP でこれを経験しているはずなので、どういったプログラムかも想像しやすく、練習問題としては最適だろうと思った。

当初予定していたのは、プログラムを実行した際、ユーザに身長と体重を尋ね、入力された値を元にして BMI 値を計算するというプログラムである (図 2 参照)。BMI 値を計算する部分は、Python と PHP で経験していたため、Java 言語になってもそこまで難しくはなかった。作成する上で難しかったのは、ユーザに身長と体重を尋ね、その値を受け取って変数に代入することだった。これまで私が学んできた Java の学習には、このようなプログラムが無かったため、その方法を調べることとなった。その結果、java.io パッケージの BufferedReader クラスを

利用して、入力した文字列を受け取るサンプルプログラムがあったので、それを参考にして作成した。まず `BufferedReader` クラスのインスタンスを作成し、`readLine()` メソッドで入力された文字列を読み取るという処理を行い、次に読み取った文字列を数値に変換して変数に代入することで、変数を計算に利用する事ができるという仕組みである（「PersonTest.java」参照）。

しかし、練習問題としてこれを作成して貰うのは難しいのでは無いかと思い、あらかじめソースコードに記述してある身長と体重を計算して表示するだけのプログラムに変更した。ユーザに身長と体重を入力して貰うプログラムは、練習問題とは別に作成の手順を示すことにした。今後の章に進んだ時、入力した値を受け取って利用するという機能が必要になるため、簡単なプログラムのうちにその仕組みを理解しておいて欲しいと考えたためである。

GUI アプリケーション編の練習問題として選んだのは、数当てゲームというプログラムである。これは指定した数（1～100など）の間で、コンピュータがランダムに決めた数字を答えにし、ユーザにヒントを元に答えを当ててもらおうという簡単なゲームだ（図 3-1、3-2 参照）。

作成する上で難しかったのは、ランダム変数を作成する部分と、フォーカスの設定である。ランダム変数の作成は、サンプルプログラムを参考にし、`java.util` パッケージの `Random` クラスにある、`nextInt()` メソッドを利用する事で作成する事ができた（「KazuateGame.java」参照）。このプログラムは、何度も数字を入力し、当たっているかどうかチェックする必要がある。しかし、数を入力した時、チェックする度に入力欄の文字列を `Backspace` キーを押して消し、また入力するという作業は少し面倒に感じた。入力欄上に焦点を当て、さらに選択状態になる設定

をしておけば、一回の Backspace キーで消すことや、上書き入力が可能になる。フォーカスの設定は、プログラムをテストしている時に気づいた事である。プログラミングは、ただ動くものを作るだけでなく、ユーザの事を考えて補助機能を加えることも必要であるということが分かった。

目標となるプログラムができたところで、プログラムを作成するために学習すべき内容を各章に分割した。これにより、章を順に進めれば Java の基礎が身につけられるように構成することができた。基礎編の後、Java アプレットではなく、GUI アプリケーション編を選んだことには理由がある。Java アプレットは Web 上で動作するという利点があるが、実行結果を確認するためには HTML でタグに埋め込みブラウザで実行するか、JDK 開発環境に付属しているアプレットビューワを利用しなければならない。しかし、HTML ファイルを作成するのは手間であるし、アプレットビューワを利用しようとするコマンドプロンプトでのコマンドも変わってきてしまう。今回は章の流れを意識した教材を作成したかったため、実行方法の同じ GUI アプリケーションを選んだ。

## (2) 内容の詳細

本教材は、はじめにトップページ (index.html) があり、メニューから各章に移動することができる。章は chapter1 から chapter4 までの基礎編と、chapter5 から chapter8 までの GUI アプリケーション編で構成される。chapter4 と chapter8 は、それまでの章を順に進めていれば解くことのできる問題形式になっており、模範解答の閲覧ができる。それぞれの詳しい内容は以下の通りである。

### (i) トップページ

トップページでは、本教材についての簡単な説明と注意書きを行っている（図 4 参照）。また Java でプログラミングを行うためには Java 開発環境の準備を行う必要があるが、MacOSX には JDK が元から備わっているため、改めてインストールや設定をしなくとも良い。JDK のインストール手順については、Web で検索した方が最新の環境でのインストール手順を知ることができるのでは無いかと思い、あえて本教材では解説しなかった。その代わりに、JDK6 のインストールについて解説してある『JavaDrive』（<http://www.javadrive.jp/>）へのリンクを貼り、参考にするよう薦めた。また、その他本教材を利用する上で参考になりそうなサイトや書籍などは、参考リンクや参考文献としてここで紹介した。

### (ii) chapter1 初めての Java プログラミング

chapter1 の最初では、トップページと同じ Java 開発環境の準備についての説明を行っている（図 5 参照）。これはトップページの注意書きを読まずに chapter1 に進んでしまった利用者に対する対策である。続いてフォルダの作成や Windows でのコマンドプロンプトの設定などがある。これはコマンドの入力方法などを統一することで、用意したキャプチャー画像と同じ状況にしてもらい、模範解答として参考にしてもらうためである。本章ではソースコードの中身に関する細かい説明は行わず、サンプルプログラムの記述と保存から、コンパイルして実行するまでの流れを説明している。これは多くの Java 教材でも最初に説明されているため、本教材でも同じ手順をとった。

### (iii) chapter2 Java プログラミングの基礎

chapter2 では、chapter1 で説明できなかったソースコードをもとに、クラスやメソッドなど Java の基礎についての解説を行った（図 6 参照）。『よくわかる Java』では、クラスとメソッドはそれぞれの章に分けて解説するほどの重要項目であり、オブジェクト指向プログラミングにおいては欠かせない内容である。しかし、chapter1 に比べて突然難しくなり、理解ができずにその後の学習も嫌になってしまうということ avoided かったため、本章では簡単な説明と定義方法のみを解説することにした。サンプルプログラムは、プログラムを実行すればデータが表示されるだけの簡単なものだが、クラス、メソッド、フィールドといった、Java の基礎が含まれている。クラスやメソッドなどの語句は今後の章でも頻出するため、その意味を簡単に理解してもらい、プログラミングの経験を積んでもらうのが本章の目的である。

### (iv) chapter3 配列、for 文、カプセル化、if 文

chapter3 は、chapter2 で作成したプログラムに追記していき、よりプログラムらしく改良しようというものである（図 7 参照）。配列や制御文は、本来ならばもう少し章に分けて丁寧に解説し、それぞれサンプルプログラムを用意すべきなのだろう。しかし、プログラミング演習を受講した学生ならば、これらは一通り経験しているはずなので、改めて解説せずとも理解できるのではないかと思った。そのため、本章は Java 言語でそれぞれの書き方を学んで貰うことを目的とし、サンプルプログラムもそれぞれの機能を追加していくという形になっている。

### (v) chapter4 問題：BMI 値を計算するプログラム

chapter4 は基礎編のまとめでもあり、初めて練習問題に取り組んで貰う（図 8 参照）。これまでの章は、この練習問題を解くために必要な

ことを中心に説明していたので、章を順に進めてきていけば、解くことができるようになっていく。しかし、まだ Java のプログラミングに慣れていない学生には、いくら授業での経験があっても難しいかもしれないと感じた。そのため、作るための手順をいくつかのステップに分け、解くためのヒントやポイントを記述した。クラスを自分で作り、BMI 値の計算と判定部分のメソッドを自分で考えて貰うことで、プログラムの作成に慣れて貰おうというのが本章の目的である。

#### (vi) chapter5 Java で GUI アプリケーションを作る

chapter5 からは GUI アプリケーション編になるため、本章の最初では GUI アプリケーションについて解説している (図 9 参照)。これまでのアプリケーションと GUI アプリケーションの違いを理解して貰うために、AWT でウィンドウを表示し、閉じるというだけの簡単なサンプルプログラムを用意した。

#### (vii) chapter6 画面のレイアウトとイベント処理

chapter6 では画面のレイアウトとイベント処理について解説している (図 10 参照)。chapter5 で作成したウィンドウに、ボタンなどを配置し、そのボタンをクリックすることにより発生するイベントの処理などを行うサンプルプログラムを用意した。

#### (viii) chapter7 BMI 計算アプリを作ろう

chapter7 では、chapter4 で作成したユーザに身長と体重を入力して貰い BMI 値を計算するアプリケーションを GUI で作成する方法を解説している (図 11 参照)。GUI アプリケーションになったとしても基本的な処理の流れは変わらないという事を理解して貰うため、同じアプリケーションを作成して貰うことにした。



### (ix) chapter8 問題：数当てゲームを作ろう

chapter8 では、本教材の総まとめとして練習問題に取り組んでもらう (図 12 参照)。ランダム変数の作成方法とフォーカスの設定はこれまでの章には無かったので、サンプルプログラムを用意するなどして解説し、数当てゲーム自体は教材の利用者に考えて作成して貰うという方法をとった。それぞれの章末には、毎回まとめとして重要語句を改めて挙げたり次章の予告を行っていたが、最終章では今後学習すべき事柄として Swing などを薦めた。

### (3) 利用者のことを考えて工夫した点

制作にあたり、利用者の事を考えて「見やすさ」「分かりやすさ」「使いやすさ」の三点に気をつけた。

「見やすさ」では、主に文字のサイズや配色などのデザインに気をつけた。説明に画像を多用するため、ページ全体はシンプルに作成することを心がけた。メニューエリアを含む両サイドの背景色は薄い黄色系にし、メインエリアの背景は白で作成した。この教材を利用する人ができる限りリラックスして勉強に集中できるようなものにできればと思い、落ち着いた配色を選んだ。背景色を薄い色に設定しておくことで、説明文などの邪魔にならず、強調したい時には赤などの目立つ色を利用すれば、重要部分であると気をつけて読んで貰うことができるだろう。書籍とオンライン教材の一番の違いは、長時間画面を見ていると目が疲れてしまう事である。そのため、説明文には強調色などを取り入れてなるべくメリハリを付けるようにした。

次に「分かりやすさ」では、自分が Java を学んだ時に難しかった点、苦労した点などを特に丁寧に解説するよう心がけた。しかし、あまり丁

寧に解説しすぎると説明文が長くなってしまうため、内容を簡潔にまとめ、尚且つ重要部分はしっかりと伝えられる文章を考えることが、教材を制作する上で最も難しかった。文字で伝えることの難しい実行結果などは、キャプチャー画像を活用した。自分が教材を利用していたときに感じたそれぞれの良い点を取り入れ、重要部分を太字で強調したり、構文などはポイントとして枠で囲うなどの工夫をした。また、実際にプログラミングをしながら学んでいけるような教材にしたかったため、それぞれの章にサンプルプログラムを用意した。実際にプログラムの動きを体感して貰うことで、より理解しやすくなり、プログラミングの経験を積むこともできる。

過去の卒業制作物でオンライン教材を作成したもののアンケートを見ると、練習問題の答えが欲しいという意見があったため、本教材の練習問題には全て模範解答を用意した。途中で諦めて全ての解答を見てしまうといったことがないように、途中段階の模範解答や実行例を確認できるようにしている（図 13-1、13-2 参照）。授業と違って途中で質問することができないので、できる限り自分で考えて解いて貰えるようにこのような方法をとった。過去のオンライン教材は、学習のためにソースコードを画像化し、コピーできないようにしているものもあった。しかし、初めてのプログラミングでは入力ミスによるエラーの発生も多くなるだろう。そういった場合に備え、ソースコードは基本的に画像で説明するが、必要に応じて完成したファイルのダウンロードもできるようにした。

「使いやすさ」は、オンライン教材ならではの注意点だった。『Java 初心者入門講座』を利用していた際、ページをスクロールするとメニューもスクロールされてしまい、戻るときにまた上へとスクロールしなければ

ばならない手間があった。本教材では、疑似フレームレイアウトによりメニューを固定し、メインエリアの右下にページの先頭へと戻るリンクを配置した。メニューを固定するには、画面内に収まるようにする必要があったため、各章のリンクをクリックした場合のみ、その章の中の見出しが表示されるようにした。それぞれの見出しに ID を付けているので、章全体をスクロールせずとも、必要な見出しに飛ぶことができる。別の章を学んでいて前の章を参照したい場合に、ピンポイントで見出しに飛べる機能があった方が良いのでは無いかと思い、これを用意した。

#### 4 評価

本教材が一通り完成した時点で、福田ゼミの 3 回生に 20 分程度本教材のモニターとして取り組んでもらい、終了後にアンケートを実施した。アンケートは見た目、内容、その他の 3 項目に分け、それぞれの項目をさらに細かく分類した。アンケートの詳細は以下の通りである。

##### 1. 見た目

- ・ デザイン
- ・ 文字や画像のサイズ

##### 2. 内容

- ・ チャプターの構成
- ・ 説明文の内容・量
- ・ 練習問題の難易度

##### 3. その他

- ・ この教材は初めて Java を学習する人にとって使いやすいと思うか
- ・ この教材を使って、Java について理解できそうだと思うか

- ・ その他何か思った事

回答者は5名で、見た目と内容については、問題点があったかそうでなかったか、その他は思う、思わないの二択に答えてもらい、問題点があった場合にはその理由を回答してもらった。

## (1) アンケート結果と考察

### (i) 見た目

デザインの項目では、5人中3人が問題点ありと回答した。その理由として、3人とも「トップページの参考リンクのリンク色が背景色とかぶって見辛い」と回答した。参考リンクのリンク色は、マウスカーソルを合わせるとピンク色に変化するが、そのままではオレンジ色をしていた。メインエリアの背景色は白だが、メニューなど両サイドのエリアが薄い黄色をしていたため、リンク色が同系色になってしまい、見づらく感じてしまったようだ。見やすさには気をつけていたつもりだったが、3人が解答しているということは私の配慮が足りなかったらしい。メインエリアの右下に配置しているページ先頭へのリンクも同じ配色だが、これはアンケートでも「右下の TOP は気にならない」と付け加えられていた。そのため、今回は参考リンクのリンク色のみ、元のオレンジ色から、背景色と差のあるピンク色に変更した。先頭へのリンクは、ページをスクロールして読む利用者の事を考えて配置したものだったが、利用者はメニューの方を主にクリックしているらしく、参考リンクほどの重要性が無かったため問題点として指摘されなかったのだろう。

他にも、「シンプルでわかりやすいが右側に空きがあるのが勿体無い」という回答があった。右側の空きは、あまりメインエリアを広くした場合、小さいモニターで見たときに文字や画像が切れてしまうなどの問題

が起きる可能性があるために空けていたものである。そのため、この部分は修正しなかった。

文字や画像のサイズでは、5人中1人が「文字サイズが全体的に小さいめなので、長時間見ているのが辛い」と回答した。修正を試みたが、文字サイズを大きくするとレイアウトが崩れてしまったり、小さいモニターでは説明文で画面が埋まり、文字の量が多いと感じるようになった。スタイルシートでは、「font-size」プロパティに「small」を指定していた。しかし、「small」はブラウザのフォントサイズに影響を受けるという特徴がある。そのため、「font-size」プロパティは「small」ではなく、「%」による指定に変更した。

## (ii) 内容

チャプターの構成は、5人全員が問題なしと回答した。本教材を作成する上で非常に悩んだ点なので、良い結果が得られて満足している。しかし、今回は5人という少ない人数に対してのアンケートなので、もう少し人数が増えたらこの結果も変わってくるだろう。

説明文の内容・量については、5人中2人が問題ありと回答した。その理由は「赤字のサイズをもう少し大きくしたほうが、重要だと意識できると思う」、「内容は分かりやすいが、少し文字の量が多くてつかれる」ということだった。赤字と言うのは、説明文の中で重要語句などの強調に使用していた色である。これはデザインにも含まれると思うが、説明文の問題でもあったので、この項目に分類した。指摘された通り、強調に使用している<em>、<strong>タグの文字サイズを修正した。

説明文は、自分自身が Java を学ぶ上で難しかった部分などを丁寧に説明するように心がけたため、その分文字の量が増えてしまい、疲れる

と指摘されてしまった。しかし、この回答者はその他の項目で「文字の量が多いと書いたが、それは少し気になるくらいで、説明も分かりやすくとても勉強しやすいと思う」とも回答していた。そのため、説明文の内容は、誤字脱字の修正のみ行った。

練習問題の難易度は、5人中1人が「もう少し練習問題の難易度を上げて良いのではないか」と回答した。各章を進めていけば誰にでも解答できる問題を目指したつもりだったが、そのせいで物足りなく感じてしまった人も居たようだ。Javaに興味を持ってもらうのも本教材の目的であるため、本教材の練習問題を物足りなく感じた人は、恐らく自分自身で他の教材を利用し、Javaについて学習して貰えるのではないかと思う。問題ありと答えたのが5人中1人という割合を考え、難易度には変更を加えなかった。

### (iii) その他

その他の項目では、この教材は初めて Java を学習する人にとって使いやすいか、この教材を使って Java について理解できそうだと思うかの二つの項目で、5人全員が思うと回答した。20分間のベータテストでは、この教材の全てを見る事はできなかつただろう。つまり、これらはサイトのデザインや章の構成などを見て感じた印象であるとも言える。この項目で5人全員が使いやすいそう、理解できそうだと思うと解答したということは、すなわち教材として取り組みやすいものが作れていると言えるのではないだろうか。その他の意見でも、「説明文に対して画像や色が多用されているためわかりやすかった」という回答があったため、工夫の成果が得られていると思う。

その他指摘された点は、「よく間違えそうなところをあらかじめ書いておくとスムーズに学習できそう」、「ある程度知識があれば分かりやす

くとても良いと思うが、全くの初心者の場合ファイル名等で大文字小文字が混ざりコンパイル時に詰まったりしそう」、「ソースファイルをコピーできる仕様になっているが、ソースコードをテキストボックス化するのも良いのではないか」などの意見があった。

これらの項目は、私自身ベータテストを行っている学生を後ろで見て、修正しなければならないと感じた点でもある。実際に学生達が教材を利用する様子を見て、学生がどういったところで間違いやすいかなどを再確認することができた。学生達がよく間違えたりミスをするのは、やはりソースコードを入力する部分である。例えば、小文字の「l」と大文字の「I」などは、フォントによっては判別がしづらい。ソースコードの画像を見て入力したとき、自分では正しく入力したつもりであるので、その間違いに気づきにくいのだ。ソースコードをテキストボックス化するのも良いのではないかという意見が出たのもそのためだろう。しかし、章が進むにつれてソースコードは長くなってしまいうため、画像とテキストボックスの両方を載せるとページ全体が長くなってしまいう。そのため、今回テキストボックス化は行わなかった。間違いやすい箇所については記述していたのだが、ソースコードの画像の後に書いていた。学生が間違いやすい箇所を確認してからソースコードの入力ができるように、説明文の内容は変えず、順序のみを変更した。

## (2) アンケート後の改良点

アンケートの結果をもとに、いくつかの箇所に変更を加えた。以下がその変更箇所である。

1. トップページにある参考リンクのリンク色を、オレンジからピンクに修正 (図 14 参照)。

2. 全体の font-size プロパティを、small から % 指定に変更。
3. <em>タグ、<strong>タグの文字サイズを修正。
4. 間違えそうな項目について、説明文の順序を変更。
5. 誤字脱字の修正。

アンケートの回答には無かったが、先生にプログラムを見てもらったところ、エンコーディングに問題があるのでは無いかという指摘を受けた。『JavaDrive』では、「保存する時の文字コードは Windows の場合であれば「Shift\_JIS」で結構です」と書かれていたため、本教材でも同様の説明を行った。コマンドプロンプトのデフォルトの文字エンコーディングは「Shift\_JIS」なので、Windows ならこの方法でも問題なかった。しかし、ターミナルの場合、デフォルトの文字エンコーディングが「UTF-8」のため、コマンドの入力などは同じでも、ターミナルの文字エンコーディングとソースコードの文字エンコーディングが一致しなかった場合、文字化けが発生してしまうのだ。この問題を解決するため、コンパイルの際にコマンドでエンコーディングを指定する方法も追記した（図 15 参照）。

### (3) 自己評価

今回 Java の入門教材を作成し、アンケートでも「この教材は初めて Java を学習する人にとって使いやすいか」、「この教材を使って Java について理解できそうだと思うか」の二つの項目で 5 人全員から良い評価を得られた。しかし、卒業制作という観点で自己評価した場合、満足のいく結果だったとは言えない。

最初に提出した題目は、「Java を使った服飾原型の製図アプリケーション作成について」という、今とは全く違ったものだった。私は高校



で家庭科を卒業しており、卒業後も趣味で服作りを続けていたため、卒業制作にもこの経験を活かしたいと考えていた。服飾原型は、人の肩幅や背丈などを測った数値を元にして計算を行い、決まった形で描かれる。そのため、はじめはこの服飾原型の製図をプログラミングで再現し、自分のように趣味で服を作る人の手助けになるようなアプリケーションを作成しようと考え、Java を学び始めたのだ。しかし、いくら授業でプログラミングの経験があるとはいえ、全く新しい言語を理解し、尚且つアプリケーションを開発するというのは、時間的にも私の力量にも無理があった。基礎を学習するだけで精一杯になり、GUI や描画といった製図アプリケーションの開発に必要な学習がほとんどできなかったのだ。

それならば、自分と同じようにプログラミング経験が多少ある人間が、これから Java を学ぶための教材のようなものがあれば良いのではないだろうかと思ったのが、今回のテーマに至った理由である。相手の立場になって考えることができるため、より良い教材が作成できるのではないかと考えた。アンケートでも分かりやすいという意見を貰うことができたので、結果として自らの経験を活かした教材が作成できたと思っている。

## 5 まとめ

### (1) 今後の課題

今後の課題としては、もう少し基礎としての内容を充実させる事などが挙げられるだろう。基礎編では、switch 文や while 文など、他の制御文についての説明も行いたかった。これらについて説明するためには、新しいサンプルプログラムを用意し、新しい章を作成する必要があるだ

ろう。しかし、あまり章の数が増えると、休暇中に利用して貰うのが難しくなってしまうかもしれない。手軽に Java の学習に取り組んでもらうというのも本教材の目的であり、なるべく章の数も少なくまとめたかったため、今回は練習問題を解くために必要なことだけを取り入れた。アンケートでは良い結果を得られたが、まだまだ Java の基礎を学習する上では不十分な教材であるとも言える。

本教材の GUI アプリケーションは、全て AWT で作成した。AWT は非常にシンプルで覚えやすいが、必要最低限の部品しか用意されていないため、あまり高度な GUI アプリケーションは作成できない。また、AWT は実際には OS によって異なる振る舞いをする難点がいくつか指摘されている。AWT と同じく、Java で GUI アプリケーションを作成できる Swing では AWT の難点の解消が図られており、AWT よりも実行速度が向上し、HTML に対応している他、OS による実行の様子の違いも抑えられている。

『Eclipse 3.6 では始める Java プログラミング入門—Eclipse 3.6 Helios 対応』では「AWT を覚えても、実際にそれを活用するシーンはあまりない」とはっきり書かれており、Swing は「Java におけるさまざまなプログラムの GUI の基本」であるとしている。しかし、Swing が AWT を継承しているため、Swing を学ぶ為には基礎として AWT を学んでおいた方が良いとする考え方もある。実際私も『Java 初心者入門講座』で AWT から Swing への書き換えについて学んだが、基礎程度ならそれほど書き方に違いは無いように感じた。そのため、簡単な AWT の方が GUI アプリケーションの基礎として入りやすいのでは無いかと思い、本教材にも取り入れた。本教材の内容を増やす機会があれば、Swing についても解説していきたい。

## (2) おわりに

今回、本教材の制作に取り組み、決して万人に通用する教材ではないが、福田ゼミの基本テーマである「人の役に立つものを作る」は達成できたのではないだろうか。当初のテーマと変わってしまったり、自身が Java を学習することに時間を割いてしまったりと、その制作過程は決して順調といえるものではなかつただろう。しかし、自分自身がそういった苦労を経験した事により、利用者の視点に立って考えることにも繋がったと言える。自分が学習する上で分からなかったことを丁寧に説明した事で、利用者にも分かりやすかったと感じ取ってもらうことができた。また、実際に利用してもらい、分かりやすかった、見やすかったという意見を貰えた事で、最終的には本教材を制作して良かったと思えた。人の役に立つものを作るためには、相手の立場になって考えることが大切であるという考えも間違っていなかつただろう。

Java のプログラムは、Java 仮想マシンが動くコンピュータならどんなプラットフォームでも実行することができ、デスクトップで動く Java アプリケーションや、Web 上で動くサーバーサイドプログラム、デジタル家電、携帯電話の i アプリなどさまざまな分野で利用されている。しかし Java の基本文法は同じなので、1つの分野で学んだことが無駄にならない。1つの言語で色んなことができることが、Java プログラミングの面白さであるとも言える。本教材を使い、利用者には Java がどういった言語なのかに触れ、Java プログラミングのもつ可能性というのを感じてもらいたい。そして、本教材をきっかけに Java に興味を持ち、Java プログラミングを本格的に学びたいと思って貰えたら幸いである。

## 注

- (1) Python—— プログラミング言語のひとつで、Perl などと同じくコンパイルを必要としないスクリプト言語に属する。  
(<http://www.python.org/>)
- (2) PHP—— 動的に Web ページを生成する Web サーバの拡張機能の一つ。正式名称「PHP: Hypertext Preprocessor」。  
(<http://www.php.net/>)
- (3) API—— あるプラットフォーム (OS やミドルウェア) 向けのソフトウェアを開発する際に使用できる命令や関数の集合のこと。  
『IT 用語辞典』 (<http://e-words.jp/w/API.html>)
- (4) GUI—— ユーザに対する情報の表示にグラフィックを多用し、大半の基礎的な操作をマウスなどのポインティングデバイスによって行うことができるユーザインターフェースのこと。  
『IT 用語辞典』 (<http://e-words.jp/w/GUI.html>)
- (5) JDK—— Java 言語でプログラミングを行う際に必要な最低限のソフトウェアのセット。Sun Microsystems 社が開発した。  
(<http://www.oracle.com/index.html>)
- (6) 『カレントアウェアネス・ポータル』  
2009「W3C、XHTML2 の策定を中止し、HTML5 の標準化に注力」。  
(<http://current.ndl.go.jp/node/13483>)
- (7) Swing—— Sun Microsystems 社のプログラミング言語 Java に標準で付属するグラフィック関連のクラスライブラリ。  
『IT 用語辞典』 (<http://e-words.jp/w/Swing.html>)
- (8) Eclipse—— フリーの Java 統合開発環境。現時点での最新版は、Eclipse 3.7.1。  
(<http://www.eclipse.org/>)

- (9) AWT—— Java でグラフィカルユーザーインターフェースを実装するために用いられるクラスライブラリの名称。Java 言語でボタンやメニューバー、チェックボックスなどを用いたアプリケーションを開発することができる。

『IT 用語辞典』 (<http://e-words.jp/w/AWT.html>)

## 文献表

鳥海不二夫

2007 『よくわかる Java』 秀和システム

掌田津耶乃

2010 『Eclipse 3.6 ではじめる Java プログラミング入門—Eclipse 3.6 Helios 対応』 秀和システム

堀江幸生

2003 『世界でいちばん簡単な Java の e 本—Java の基本と考え方がわかる本』 秀和システム

『Java 初心者入門講座』

<http://sunjava.seesaa.net/>

『Java AWT 入門』

<http://wisdom.sakura.ne.jp/system/java/awt/>

『JavaDrive』

<http://www.javadrive.jp/>