

# 一般学生向けの Java 言語による プログラミング入門

久野 禎子

(早稲田大学メディアネットワークセンター)



## ✓ 科目の概要

早稲田大学の文系寄り学部が集まる早稲田・戸山キャンパスで学部1年から院生までを対象とする、メディアネットワークセンター(MNC)設置科目「プログラミング初歩」について報告する。この半年間の科目は、「興味がある」という動機から、単位には縛られず自発的にプログラミングを学ぶ履修生が大半を占めるという意味では、教えやすい。他方この科目は、全学共通副専攻制度に基づきMNCが設置するソフトウェア学・統計学副専攻コースの選択科目でもあり、これらの副専攻を修めようとするプログラミング初学者には残りの単位を積み重ねる土台となる。また、SEなどの職種に就職が決まり、卒業間際にプログラミングを学んでおこうとする者も少なくない。「プログラミングが初体験」という以外はバラバラな学生たちを相手にしてきて大事だと思うのは、プログラミングに興味を持ち自主的に学びに来た人たちを(1)苦しめないこと、(2)つまらないものだと失望させないこと、の2つである。人生においてプログラミングを学ぶのは、この科目の半年間だけになる人も多いはずだ。半年間プログラミングを楽しみ、講義終了後もプログラムを作り続けたいと思える土台を作れば理想的だが、そうでなくてもプログラミング体験が将来の仕事や生活に何らかの形で役立ってくれることを願っている。(1)(2)とも「苦しめないで面白いプログラムが作れるようになるか」という話なので、楽しく授業を進める工夫が不可欠である。

授業はPC室で毎回実習をしてもらいながら進める。1クラス最大40名の登録であり、今年は13クラスが開講された。使用言語は担当の教員により異な

り、筆者はJava言語を使用した。Javaは初学者にとっての「おまじない」が多いため(1)の点では最悪だが、学んだ結果役に立つものが作れる点と、グラフィックスの例題を豊富に用いて理解の助けにしたいという理由で選択している。加えて、多くの学生の作ってみたいものがゲームであり、アプレットで動かせばWebページ作りも楽しくなる、という旨みもある。

## ✓ Java 言語でプログラミングの何を学んでもらうか

筆者自身がプログラミングに入門した頃は、まだ手順を書き下すだけ、繰り返しと分岐、そして配列を学んだら、あとはがんばって書くという時代だった。今でも繰り返しと分岐でアルゴリズムを実現するのがプログラミングの本質であることは変わらないが、その周辺はかなり様変わりして、オブジェクト指向が採り入れられて部品化も進んだ。今では、オブジェクト指向のクラス概念を理解せずに使いこなせるプログラミング言語はないし、API文書を利用せずに作れるプログラムもない。Javaを選んだ以上、Javaのクラスについて、インタフェースクラスや継承まで含めて理解すること、APIの使い方がひと通り分かることの2つは、具体的な履修項目として欠かせない。クラス概念を理解すれば、自分でプログラムを作るとき、手順よりもオブジェクトに注目したプログラム設計ができるようになる。それをしながら、従来どおり問題解決のさまざまなアルゴリズムを実現できる力を付ける必要がある。

## ✓工夫1: グラフィクス例題の多用

筆者はグラフィクスの例題を多く用いるが、これは文系の人でも自分が書いたプログラムで何が起きているか分かりやすいためである。以前は簡単な数値計算で繰り返しと分岐の構文を学んでからグラフィクスの例題に移り、図形をクラスにして描き、その図形をアニメーションに使うという流れで授業を行っていた。しかし退屈な数値計算は最小限に抑えても失望を買ってしまうので、今は最初のプログラムからグラフィクスの例題とし、最初からAPI文書でGraphicsクラスの使い方を調べつつ、円の描画を繰り返しと分岐で行わせ、配列の説明も済ませた後、円をクラスにして描くプログラムに移る。図-1は円を一定間隔で並べながら、ある条件のものは位置や色を変えるという練習問題の画面であり、繰り返しと分岐の演習になっている。

## ✓工夫2: dolittle 言語の併用

工夫1の障害は、Javaのグラフィクスプログラムが長く、初心者には説明の難しい部分が多かった。長々しいプログラムはそれだけタイプミスも増え、コンパイラに引っ掛かり、プログラムが動かないという不愉快な体験を強いる。デバッグに慣れるのも大事だが、簡単なプログラムさえ自分でちょっと書いてみたらいきなり動かない、というのでは意欲を削がれる。例題プログラムをそのまま打っても動くまでに大変な思いをするようでは、例題で説明したい事柄の本質まで考える余裕がなくなり、落ちこぼれていく原因になる。Javaのそうした学びにくさを補うものとして、筆者は豊富なグラフィクス機能を持つ教育用言語dolittle<sup>1)</sup>との併用学習を試みている。dolittleでは日本語を用いた記述が可能であり、タートルグラフィクスによって描いた図形をそのままオブジェクトとして操作できる。昨年は最初にdolittleに4週間かけ、かなり複雑なゲームの設計・実装ができるまで学んでからJavaの学習に移った。dolittleではあまりエラーに出会わずどんどんプログラムが動かせるので「気軽に自分で考えてプログラムを書くことを楽しめる」というの

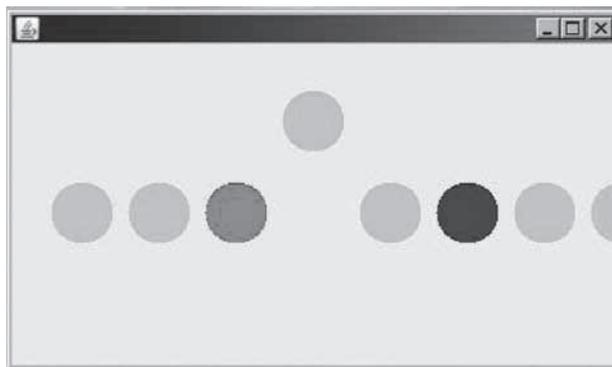


図-1 繰り返しと分岐で円を描画

は画期的な成果だった。しかしそのぶん、Javaに移ってからの窮屈さに耐えられない人が出てしまい、dolittleとJavaで学ぶことの繋がりも分かりにくかったようだ。

今年はその反省から、学習のメインはJavaの習得に置きつつ、毎回の授業で取り扱う事柄を説明する際にdolittleプログラムでエッセンスの部分を見せ、同じことをJavaで書くようになるよ、というやり方で進めている。オブジェクト定義もJavaはクラス方式、dolittleはプロトタイプ方式なので、その違いについても説明をし、比較しながら進めた。両者の違いは説明だけでの理解は難しいが、使えば自然と分かるというのは発見だった。プログラミング言語は2つ以上のものを学ぶと楽に学べるというのは真実かもしれない、もしそうなら、わずか15回の授業でもそれをやる価値はあるのだろう。

筆者は目標(1)より(2)に重きを置いたためか、毎回単位をとるのはクラスの半数ほどである。言い訳ではあるが、この授業でプログラミングの楽しさに目覚める人がいる一方で、スキルとしてプログラミングを身につけることが自分には向いていないと気付くのも大事なのではと思う。

### 参考文献

1) 兼宗 進, 久野 靖: ドリトルで学ぶプログラミング, イーテキスト研究所出版(2008).

(平成22年7月31日受付)

久野 禎子 titkuno@yahoo.co.jp

1984年広島大学総合科学部卒業, 1992年東京工業大学より理学博士。1994年より早稲田大学MNC非常勤講師として大学生の情報リテラシー教育にかかわる。ほかに東京工業大学, 慶應義塾大学非常勤講師。