

工学系学科でのプログラミング入門教育—ドリトルを利用して—

兼宗 進

大阪電気通信大学医療福祉工学科

授業の位置付け

本学の医療福祉工学科では、医学、機械、電気、情報を中核として、医療職である臨床工学技士、医療福祉機器のエンジニアを育成する教育を行っている。特に医療機器を制御するために、プログラミングによる生体計測、計測制御は必須である。しかし、学生は「医療」や「福祉」を意識して入学するため、必ずしも「工学」の意識は高くなく、入学後にプログラミングを学習することに驚く学生も存在する。

プログラミング言語としては、組込み機器を考えるとC言語が中心となる。ただし、演習や卒業研究での自由度の高さを考えると、入出力の汎用性が高いLabVIEW^{☆1}などのビジュアル言語も並行して使われている。しかし、どちらのプログラミングも定着度が高くないことが問題となっていた。

そこで、2010年度から、中学校や高校から大学までで利用されている教育用言語の「ドリトル」^{1), ☆2}を初年度のプログラミング教育に利用することにした。

カリキュラムとしては、1年生で前期にドリトルによるプログラミング入門(今回紹介する授業)を行い、夏期集中講義でドリトルによるロボット制御と計測機器利用を体験した後、後期にC言語を学習する。2年生では、LabVIEWなどの実践的な計測制御学習を行う。

☆1 <http://www.ni.com/labview/ja/>

☆2 <http://dolittle.eplang.jp>

授業回	内 容
1	授業解説, プログラム体験
2, 3	反復, メソッド定義, ボタン
4	ペイントソフト作品作成
5~7	タイマーによるアニメーション作品製作
8, 9	条件分岐で性格判断, 乱数で占い
10~12	ゲームで衝突判定, 乱数, 得点計算
13~15	自由作品制作, 作品発表

表-1 プログラミング入門授業の構成例

初年度の前期で何を学んでもらうか

医療機器を扱う技術者になるためにソフトウェアの知識は欠かせないが、プログラミングに興味を持って入ってくる学生は稀である。

そこで、入学直後に行う1年生前期の授業では、「毎回の学習項目で達成感を持たせる」「自分のアイデアの入ったプログラムを作ることで楽しさを感じさせる」「もっと作りたい、と思っている状態で90分の授業を終える」ことを目標にした。

表-1に、半期のカリキュラム例を示す。学習する項目としては、「回数を指定したループ」「メソッド(関数)の定義」「GUIオブジェクトの利用」「タイマーによる間欠的なループ」「条件分岐」「論理式」「乱数」などがある。

時間	内 容
20 分	サンプルプログラムで条件分岐と論理式を説明
5 分	メッセージを変更しながらプログラムを理解
15 分	質問を 2 種類に増やし分岐をネストさせる
15 分	質問を 2 種類に増やし論理積で判定させる
25 分	今日の学習内容を利用した作品制作
10 分	課題提出, 授業アンケート

表-2 条件分岐を学習する回の流れ

ラーメンは好き? :		Yes :
判定		判定 麺類好きですね

1	Q1 = 選択メニュー! "ラーメンは好き?" "Yes" "No" 作る。
2	判定ボタン=ボタン! "判定" 作る 次の行。
3	判定ボタン: 動作=「
4	A1 = Q1 ! 何番目?。
5	「A1 == 2」!
6	なら「ラベル! "麺類好きですね" 作る」
7	そうでなければ「ラベル! "本当?" 作る」実行。
8	」。

図-1 プログラム例と実行画面 (性格判断のサンプル)

授業の実施例

実際の授業の例として, 8 時間目に実施した条件分岐の回を紹介する. 40 名ずつの 2 クラスで実施した. この授業では, 学習目標を条件分岐と論理積の理解とした. 題材は性格判断である. 表-2 に授業の流れを示す.

図-1 は, この回の授業で最初に示したサンプルプログラムである. 説明のために行頭に番号を置いた. 1 行目では 3 つの選択肢を持つ Q1 という名前の選択メニュー (プルダウン式に選択する GUI 部品) を生成し画面に置いている. 2 行目では判定を行うボタンを生成し画面に置いている. 3 行目ではボタンを押したときの動作を定義している. 動作の内容は 4 行目から 7 行目である.

判定ボタンが押されると, 4 行目で選択メニューで選ばれている項目が変数 A1 に代入される. 「ラーメンは好き?」「Yes」「No」はそれぞれ 1 から 3 の値になる. 5 行目から 7 行目は if, then, else に相当する条件分岐である. 5 行目では A1 の値を調べ, 値が 2 (「Yes」が選択された) の場合は 6 行目を, それ以外の場合は 7 行目を実行する.

授業では, 教員の画面を見せて, 1 行ずつ入力させて実行させながら説明を加えていく. たとえば, 図-1 のプログラムでは, 1 行目だけを入力して実行させると, 画面に選択メニューが表示される. 次に 2 行目を入力して実行させると, 実行ボタンが表

示されるが, 動作を定義していないため, 押しても何も起こらない. 次に 3 行目, 4 行目, 8 行目を入力したのち, 4 行目の変数 A1 の値を print 文に相当する命令で画面に表示させる. その後, 条件分岐の構文を説明し, 5 行目と 6 行目で if, then の動作を確認した後で, 仕上げとして 7 行目を入力して全体の動作を確認する.

その後, 質問と選択肢, 判定のメッセージを自由に変えさせて, 隣の同級生に画面を見せて使ってもらうよう指示する.

このように, 毎回の授業の中で「サンプルを説明しながら入力して理解し, 自分なりのアレンジを行うことで興味や理解を深める」体験を 2, 3 回繰り返す.

この授業では, 続いて「今は 1 個の質問の Yes/No だったが, 複数の質問を扱うにはどうすればよいか」を考えさせた後, if のネストの例を説明する. 実際にプログラムを示し, 入力して動作を確認させるが, 条件分岐を知ったばかりであるため, 十分に理解して使いこなせる学生はまだ多くない. そこで, ここでは「入れ子として書くこともできる」ことを体験する程度にとどめ, 先に進む.

次に扱うプログラム例を図-2 に示す. 説明では, 前の例と同様に, 段階的に入力しながら説明し, 7 行目を示し複数の条件文と論理積を含めて説明したところで時間をとって, 学生に 8 行目から 10 行目を考えて入力させる.

ラーメンは好き？ チャーハンが好き？ 判定	Yes Yes 判定 中華好きですね
---	--

1	Q1 =選択メニュー！ "ラーメンは好き？" "Yes" "No" 作る。
2	Q2 =選択メニュー！ "チャーハンが好き？" "Yes" "No" 作る。
3	判定ボタン=ボタン！ "判定" 作る 次の行。
4	判定ボタン：動作＝「
5	A1 = Q1 ！何番目？。
6	A2 = Q2 ！何番目？。
7	「全部！（A1 == 2）（A2 == 2）本当！」なら「ラベル！ "中華好きですね" 作る」実行。
8	「全部！（A1 == 2）（A2 == 3）本当！」なら「ラベル！ "麺類好きですね" 作る」実行。
9	「全部！（A1 == 3）（A2 == 2）本当！」なら「ラベル！ "ご飯好きですね" 作る」実行。
10	「全部！（A1 == 3）（A2 == 3）本当！」なら「ラベル！ "中華苦手ですか" 作る」実行。
11	」。

図-2 プログラム例と実行画面（性格判断の拡張サンプル）

全員が意味を理解したところで、この時間の自由作品に取り組みさせる。この回の課題は、図-2を次のように拡張するものである。

- 質問を3種類に増やす。判定は8パターンになる。（必須）
- 質問と判定の文を工夫する。
- 判定によって文字の色を変えたり、アニメーションが流れるなど独自の工夫を加える。

最後に授業支援システム（Moodle）にログインし、課題の提出とアンケートを入力させる。

授業で工夫した点

受講生に毎回の学習項目を理解させるために授業で心がけていることは、今回扱った性格判断のように簡潔だが目的を持って動くことで興味を持たせる題材を扱うことと、サンプルの理解とその改良を授業内で複数回行うことである。

与えられたサンプルを入力するだけでは、「頭を使わずに文字を入力するだけの行為」になってしまう。そこで、教員が1行ずつ入力して実行しながら開発していく過程を、学生にも同じように入力させながら体験させる形で進めている。

そして、小さなことでも構わないので、自分なり

の工夫をプログラムに入れられる体験を重視している。そして、それらの作品を同級生と見せ合うことを推奨している。これは、実際のソフトウェアは自分が使うだけでなく「他人に使ってもらうことができる」モデルを体験するためでもある。

プログラミングの授業で難しいことの1つは、受講生によって進度の差が大きいことである。詰まっている学生が多ければ、全員に共通のヒントを与え、少数の場合には授業アシスタント（TA）の学生に指示して対応してもらう。面白い工夫があれば、随時それを紹介する。授業の最後では、課題提出の際に数名の作品を紹介する。これらの工夫により、全員に（小さなものでもよいので）プログラム製作の達成感を与えつつ、よいプログラムを作った者にはそのことを伝えて正のフィードバックを作るように工夫している。

どのように評価するか

授業の評価については、毎回の授業の最後に、その日作成した作品プログラムを提出させることで行っている。評価項目は次のようなものである。

- その日の学習内容を使いこなしていること。
- 独自の工夫が含まれていること。

その日の学習内容が含まれていることは必須である。独自の工夫については、サンプルプログラムを入力しただけでは作品と認めず、サンプルとの差分で評価すると伝えている。

最後の3回の授業を使い、自由作品を作成した。受講者のほとんどはプログラミングの初心者であったが、全員がオリジナルの作品プログラムを作成することができた。作品には、季節に応じたアニメーション、乱数を利用した占い、迷路、シューティングゲームなどがあった。

プログラムを分析した結果、授業で扱った学習内容を活用し、独自の作品を作成できていることを確認した。

授業の結果

最終回に実施した、半期を通したアンケートを見ると、理解度、興味、難易度に関する4段階の選択結果は次のようになった。適度な難易度を保ちつつ、高い学習意欲を維持できたことが分かる。

- ・「よく / だいたい理解できた」が計 86.5%
- ・「とても / まあまあ楽しかった」が計 100%
- ・「とても / 少し難しかった」が計 83%

自由記述では、「楽しかった」「自分でもプログラムを作れることが分かった」「もっと作ってみたい」という感想が多く、次の学期に行われるC言語の授業や制御プログラミングの授業への期待が多く見られた。

実際に後期のC言語の授業では、例年より意欲を持って取り組んでくれたことが担当した講師から報告されている。定量的な評価は行っていないが、特にループと条件分岐についてはスムーズに学習できていたようである。

これらの結果から、プログラミングの基本的な考え方を習得させながら、プログラミングに対する興味を持たせ、C言語等の汎用的な言語の学習につなげることができたという意味で、初年度の入門教育としては意味があったと考えている。

今回は筆者が開発しているドリトル言語を用いた

授業を紹介した。ドリトルを用いる利点としては、「基本的に1行ずつ追加しながら説明できる」ことが対話的な授業に有効だった。また、英語が得意でない学生にとっては、命令等が日本語であることは敷居を下げていた可能性がある。構文が独自である点や、変数に型がない点などは、他の言語への発展を考えたときに好みに分かれるかもしれない。「オブジェクトを生成してメッセージを送る」モデルは、特に問題なく記述できていた。今回はCへの発展だったが、C++やJavaなどのオブジェクト指向言語への発展についても可能性を感じている。本誌で紹介されたドリトルとJavaを連携させる形の授業も興味深い²⁾。

入門の授業においては、言語の違いよりも、授業の進め方の影響のほうが大きいと感じている。授業の工夫をすることで、他の言語を用いた場合でも初心者プログラミングの興味を持たせる授業は可能と思われる。たとえば、Scratch³⁾などのビジュアル系言語や教育用言語⁴⁾、Processing⁵⁾、Ruby⁶⁾などのスクリプト系言語を利用することが考えられる。

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 第2版, イーテキスト研究所 (2011).
- 2) 久野 慎子: 一般学生向けのJava言語によるプログラミング入門, 情報処理, Vol.51, No.10, pp.1345-1346 (Oct. 2010).
- 3) 伊藤 一成: Scratchを用いた授業実践報告, 情報処理, Vol.52, No.1, pp.111-113 (Jan. 2011).
- 4) 兼宗 進, 阿部和広, 原田康徳: プログラミングが好きな言語環境, 情報処理, Vol.50, No.10, pp.986-995 (Oct. 2009).
- 5) 菊池 誠: Processingによるプログラミング教育, 情報処理, Vol.52, No.2, pp.213-215 (Feb. 2011).
- 6) 増原英彦: プログラミングを教える・プログラミングで教える, 情報処理, Vol.51, No.12, pp.1627-1629 (Dec. 2010).

(平成23年3月28日受付)

兼宗 進 (正会員) kanemune@acm.org

2004年筑波大学大学院ビジネス科学研究科博士課程修了。博士(システムズ・マネジメント)。企業勤務後、一橋大学准教授を経て2009年から大阪電気通信大学医療福祉工学部教授。プログラミング言語、情報科学教育に興味を持つ。