

# 八戸工業大学工学部システム情報工学科における プログラミング入門教育の展開

小久保 温<sup>†</sup>

## Development of a Class on Introduction to Programming in the Department of System and Information Engineering, Hachinohe Institute of Technology

Atsushi KOKUBO<sup>†</sup>

### ABSTRACT

In the Department of System and Information Engineering, Faculty of Engineering, Hachinohe Institute of Technology, the programming environment was changed from command line tools for programming language C to Processing in the introductory programming class from the 2018 academic year. Then students began to understand the class better. In addition, the year 2020 saw an outbreak of COVID-19. Until the 2019 academic year, face-to-face classes were held in one classroom; from 2020, classes were divided into three classrooms; one classroom held face-to-face classes and real-time video feeds were provided to the other two classrooms. This change did not affect the students' comprehension.

**Key Words:** introduction to programming, programming language C, Processing, preventing COVID-19

**キーワード:** プログラミング入門教育, C言語, Processing, 新型コロナウイルス感染症対策

### 1. はじめに

八戸工業大学工学部システム情報工学科(以降、「システム情報工学科」)の1年生「プログラミング入門」の授業を、著者は2017年度から担当している。システム情報工学科は、コンピュータを学ぶ学科ではあるが、多様な入学者がいてその多くは、プログラミングを学ぶのは「プログラミング入門」の授業がはじめてである。

本論文では、2018年度から「プログラミング入

門」の教育内容を変更したことで学生の理解度が向上したことを論じる。また、2020年度は新型コロナウイルス感染症対策のため授業方法を変更したが、理解度に悪影響を与えている様子が見られなかったことなどを論じる。

また、2020年度の入学者からは、ノートパソコンを必携化している。これは新型コロナウイルス感染症の対策とは独立に検討され実施された施策である。後世への記録のため、これについても詳しく紹介する。

---

令和2年12月7日 受付

<sup>†</sup> 工学部システム情報工学科／大学院工学研究科電子電気・情報工学専攻・教授

## 2. プログラミング教育の難しさ

マーク・ガズダイアル「プログラミングを学ぶことはなぜそんなに困難か？」<sup>1)</sup>は、大学などにおけるプログラミング教育の包括的なレビューである。このレビューでは、タイトルの通りプログラミングを学ぶことは難しいことが示されている。ここではその内容を短く紹介する。

アメリカではプログラミングの入門科目は「CS1」(Computer Science 1)と呼ばれる。BennedsenとCaspersenの2007年の報告<sup>2)</sup>によると、学生の3割が「CS1」を落第あるいは履修を取り消している。その他にもプログラムのくりかえしを中断する問題<sup>3)</sup>、基本的な算術演算の問題<sup>4)</sup>といった、初歩的な問題でも正答率が低く、一部の学生は優秀な成績を修めるが、多くの学生はそうではないことが紹介されている。

しかし、学生は、場合分けが必要な複雑なコンサートのチケットの販売フローを考えることができるという研究<sup>5)</sup>もあり、考えたことをプログラムとして記述することが難しいらしいことが示唆される。

また、プログラミングで扱う対象をヴィジュアルなものにすると理解度が向上するという報告<sup>6,7,8,9)</sup>もある。

## 3. 「プログラミング入門」の変更

システム情報工学科では、2017年度までプログラミング言語C<sup>10)</sup>で、コマンドラインツールを用いて教育を行っていたが、先行研究のように学生の理解度は高くなかった。そこで、2018年度からProcessing<sup>11)</sup>を用いてヴィジュアルな対象を問題として扱うように変更したところ、学生の理解度が向上した。

以降、この変更の詳細を論じるが、2020年度にはノートパソコンの必携化、および新型コロナウイルス感染症対策により、教育環境が大きく変化した。教育環境の変化は、教育方法にも大きく影響したため、教育環境の変遷を先に説明し、次に授業内容の変更を論じる。

## 4. 「プログラミング入門」の教育環境の変遷

### 4.1 ノートパソコンの必携化

システム情報工学科では、コンピュータ室を用意してそれを利用して学生に教育を行ってきた。これは多くの小中学高校、大学などの教育機関で、コンピュータを用いた教育を行う際に行われてきた方法である。

システム情報工学科の場合、コンピュータのOSはMicrosoft Windowsで、ディレクトリ・サービスにMicrosoft Active Directoryを用いて利用者を認証・認可し、ネットワーク上に永続化記憶領域を用意してネットワーク・ドライブとして利用者に提供していた。

しかし、近年、ソフトウェアの中には、このような構成で動かなかったり、コンピュータの管理者権限がないと使えないものが増えてきた。また、コンピュータ室を用意しても、導入したコンピュータは2, 3年程度で古くなってしまふ。情報通信技術の世界では、短い期間で技術の進歩と利用形態や流行の変化が起こるため、設備は導入しただけではうまく運用することができず、導入の段階から更新に関しても意識しておく必要がある。

また、近年、スマートフォンの普及と高性能化・高額化により、個人や家庭でパソコンを所持しない学生も増えている。これは、これまでの学生がパソコンを用いて行っていたことをスマートフォンのみでやろうとする、あるいはパソコンでなければならないことを避けることにつながる可能性がある。しかし、システム情報工学科では情報技術者を育成することを目指しているし、そうでなくても仕事ではパソコンを使うシーンも多い。そのため、個人や家庭でパソコンを所持して接する時間を増やし、自分で管理することでスキルを向上させることが望ましいと考えられる。授業では、パソコンを利用して解くことが前提になっている課題なども課されるため、パソコンを所持しているかないかは、学習にも影響することが考えられる。

そこで、2020年度の新入生からノートパソコン

を必携化した。また、2019年度の入学者にも、なるべくノートパソコンを所持するように入学前から推奨していた。これに伴い、教室にはノートパソコンの電源を取れるように電源タップを用意した。また、Wi-Fiのアクセスポイントを増強し、多数の機器から無線LANを活用できるようにした。授業でもなるべくノートパソコンを利用するようにしている。

2019～2021年度の新入生に推奨したノートパソコンのスペックは以下である。

- OS: Windows10 64bit
- CPU: Intel Core i5相当以上
- メモリ: 8GB 以上
- ストレージ: SSD 240GB 以上
- ディスプレイ: フルHD(1920×1080)
- バッテリ駆動時間: カタログ値で8時間以上
- 通信機能: ワイヤレス(Wi-Fi準拠)IEEE 802.11 a/ac に対応
- カメラ: 内蔵, または別途外付けのものを用意

このスペックは、4年間使用できること、学内には電源を取りにくい環境もあることなどを考慮して決めた。ただし、高度な3D CGや機械学習などの大規模な計算で使うことまでは想定していない。それらの場合は、研究室でパソコンを用意するかクラウド環境を使用するだろうと考えた。

また、このうちカメラだけは2020年秋に追加した。近年のノートパソコンは、カメラを内蔵したものが多く、一部そうでないものもある。遠隔授業になった際にも対応できるようにカメラを明示することにした。

そして、このスペックに沿ったノートパソコンを青森県内の業者と連携して販売していただいている。価格は4年間の保障付きで税込12万円程度と16万円程度である。学生はこの業者による販売を利用してもいいし、しなくてもよいことになっている。

#### 4.2 新型コロナウイルス流行への対応

日本国内でも、2020年に新型コロナウイルスの流

行がはじまった。感染症の流行をおさえるため、教育環境にも変化が求められた。システム情報工学科では、2020年度から座席の間隔をあげ、教室の人口密度を下げることにした。

ノートパソコンの必携化のタイミングと一致したことが、この施策をすすめる上で有利に働いた。つまり、2019年度までコンピュータを使う授業は、学生を1室に集めて「密」な状態で実施していた。この場合は受講生全体の様子を見ながら授業をすすめられるというメリットがあった。2020年度からノートパソコンを持参してもらう形に移行したことで、コンピュータを設置した教室でなくても、コンピュータを利用する授業ができるようになった。そして、ある教室で教員が対面で授業を行いつつ、それをウェブ会議システムで他の教室に配信することで、複数の教室に分けて授業ができるようになった。また、自宅から受講することも可能となった。さらに配信を録画して提供することで、自宅の通信環境が整っていなかったり、体調不良などで授業の時間に受講できなかったとしても、後日、通信環境の整ったところで受講することもできるようになった。ただし、授業をする側からすると、配信や録画のみの場合、学生の様子を知ることが難しく、授業のペースを学生に合わせにくいというデメリットはある。

なお、授業における新型コロナウイルス感染症対策が必ずしもスムーズに進んだわけではない。2020年度は1年生はノートパソコンを必携、2年生は推奨、3年生以上には推奨も必携もしていない。これら学年によって事情が異なるため、「密」を避けた教室の確保と調整は複雑で難しくなり、当初予定していた時間割や教室の変更も必要となった。

なお、2020年度には新型コロナウイルス感染症対策に注意を払っていたであろう病院などでもクラスターが発生した。その原因の一つとして、業務で使用するタブレット端末やパソコンのキーボードを経由した感染が疑われている<sup>12,13)</sup>。ノートパソコンを必携化したことで、他人の触つ

たキーボードに触れなくて済むので、安全性も高まったと思われる。

また、学生は授業の資料が電子的に配布されている場合、それを手元のパソコンで見ることができる。これにより紙の資料を配布しなくても授業が可能となった。他人の触った紙の資料に触れないことは、感染症対策的には意味がある。とは言え、紙の資料と電子資料には一長一短がある。紙の資料を配布した方が、パソコンの画面が狭い場合は資料の一覧性が高い。また、書き込むこともできる。一方、電子資料の場合、資料を紛失する可能性が減り、拡大縮小が容易で、カラーのものはカラーのまま閲覧することができるというメリットがある。一方で媒体の形式やセキュリティの設定によっては書き込めない場合もある。

#### 4.3 「プログラミング入門」の教室と授業形態

2017～2019年度は、システム情報工学科1年生全員を1教室に集めて対面で授業をしていた。使用したのは八戸工業大学システム情報工学専門棟の1階のI104教室「サイバーラボラトリ」(図1)である。

2020年度は、新型コロナウイルス対策のために「密」を避け、3教室に分けて実施した。1教室は対面で行い、それを2教室にビデオ会議システム Google Meet<sup>14)</sup>で配信した。対面で行なったのがシステム情報工学専門棟の1階のI103教室「講義室」(図2)、配信したのが1階のI104教室、2階のI202教室「情報工学基礎実験室」(図3)である。

##### (1) I104教室

I104教室は階段状の教室で、プロジェクタのスクリーンは前方に1枚、座席は前向きに配置されている。机上にデスクトップパソコンが設置された教室で、2019年度までは92人、2020年度からは半分の46人まで受講可能な教室であった。そして、2019年度までは、コンピュータを利用し、1つの学年全員が同時に受講する授業で使用されてきた。

教室が大きいのでプロジェクタのスクリーン

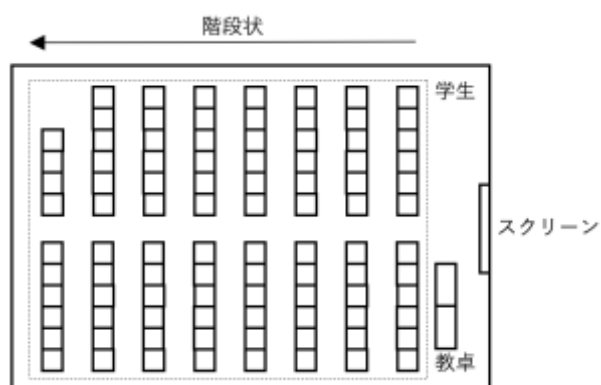


図1 I104教室: 収容人数92人

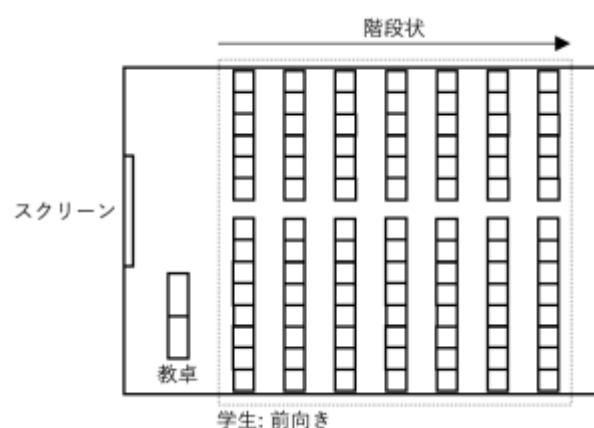


図2 I103教室: 収容人数98人

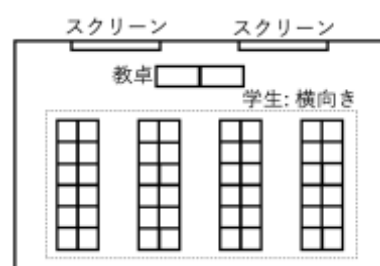


図3 I202教室: 収容人数48人

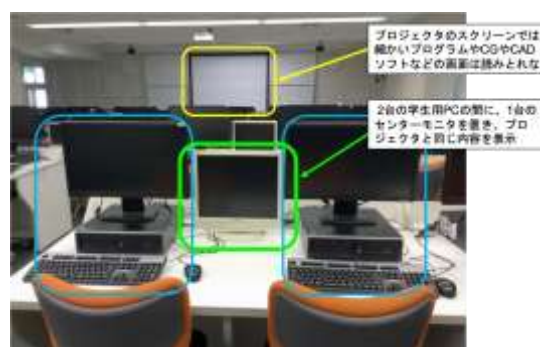


図4 I104教室のセンターモニター

は、教室の後ろの方からはっきり見えない。そこで、2席に対して1台のセンター・モニター(図4)を用意して、教卓のパソコンの画面を配信し、はっきり見えるようにしていた。

新型コロナウイルス対策として2020年度後期からはパソコンの台数を半分にし、広く空間をとった配置に変更されている。もはやセンターモニターと呼ぶことはふさわしくないが、プロジェクタの画面を手近に大きく見ることができるモニターが、1席に対して1台ずつ配置されている。

また、I104教室は階段状になっていて、学生の前にはモニタが設置されている。このため、対面の授業でも、モニタにさえぎられて学生から教員、教員から学生の顔はほとんど見えない。つまり配信でも対面でも受講生が見えるものはほぼ変わらない。そこで、この教室は他の教室から配信された授業の様子を受信して提示するのに使った。

## (2) I103教室

I103教室は、フラットな机の配置された階段状の教室である。収容人数は、2019年度までは98人、2020年度は49人である。プロジェクタのスクリーンは前方に1枚、座席は前向きに設置されている。教室は左右に広く、前後が近く、教員と学生相互に顔がよく見える。ノートパソコンを持ち込んで授業を受けるのに適した教室である。

2020年度の「プログラミング入門」の授業では、この教室で対面で授業を行い、他の教室に配信した。この教室にはパソコンが設置されていないので、プロジェクタで投影するには自分でノートパソコンを持ち込む必要がある。ウェブ会議システムで配信するとき、配信に使うのはマイクとカメラがしっかりしたパソコンであることが望ましい。筆者が授業で利用できるパソコンを考えたとき、ノートパソコンが最もこの条件に合っていた。パソコン教室に設置されているデスクトップパソコンには、カメラが付属していないし、内蔵のマイクの性能も心許ないからである。そこで、この教室にノートパソコンを持ち込んで授業し、それを他の教室に配信す

ることにしたというわけである。

## (3) I202教室

I202教室は、実験などでも利用してきたフラットな教室である。収容人数は2019年度まで48人で、2020年度からは24人である。プロジェクタのスクリーンは前方に2枚、座席は横向きに設置されている。教室はせまくスクリーンと学生の距離は近いが、横向きにながめる必要がある。机上にデスクトップパソコンが設置された教室で、プロジェクタの画面を手近に見ることができるセンター・モニターなどはない。2020年度はこの教室は配信を受信するのに利用した。

## 4.4 ティーチングアシスタントなど

八戸工業大学には、大学院生を雇用するティーチングアシスタント制度と学部生を雇用する特別指導補助学生制度があり、雇用した学生には授業のサポートをお願いできる。

特に初歩のプログラミングでは、パソコンの操作になれていなかったり、エラーに対応できない学生も多く、これらのサポート学生の助けは重要である。授業の担当教員がこのようなトラブルシューティングを行なうと、授業の進行がそこでストップしてしまうからである。

2019年度までは、70～90人程度の受講者に対し4、5人のサポート学生を雇用した。はじめて授業を担当した2017年度は、受講者がどこでトラブルを起こすか予測できなかったため、サポート学生に大きく頼ることとなった。2018年度からは、トラブルが発生しやすい部分と発生原因を把握できたため、トラブルを回避する指示ができるようになった。また2018年度からプログラミング環境にProcessingを用いるようになったことで、トラブルの発生も減った(5.1(2)参照)。

2020年度は、1教室で対面で授業を行って、これを他の2教室に配信した。配信では、サポート学生の存在が欠かせなかった。配信においては映像と音声すべてである。しかし、しばしば配信先に画面が適切に映らないことがある。典型的なのは、カメラがオフだったり、画面の共

有の失敗である。また、音声もミュートされていたり、マイクがオフだったり、ハウリングしたり、音量が大きかったり小さかったりする。もちろん、事前に準備してチェックしてから授業に臨むが、ある教室から配信したものを、別の教室で十分確認することは教員一人ではできない。また、教員は授業では授業することに注意力のほとんどを費やしてしまうため、コンピュータの操作ミスが起きやすく、適切に配信できないことがある。そこで、サポート学生には、配信先の各教室で、配信状況をモニタしてもらったり、質問への対応をしてもらったりした。サポート学生の助けがなければ、配信する授業で発生するトラブルに対応することは困難だった。

## 5.「プログラミング入門」の授業内容の変更

多くの学生は、「プログラミング入門」の授業で、はじめてプログラミングを学ぶ。そこで、この授業では、プログラミングの基本を学ぶことを目標にした。具体的には、文の記述方法などの文法、変数、データの記述方法であるリテラル、順次・分岐・繰り返しなどの構造化プログラミング、関数、配列を習得することを目指した。

そのときに、人間のワーキングメモリには容量と保持時間に限りがあり、ジョン・スウェラーの認知的負荷理論<sup>15)</sup>に見られるように、それを考慮して教育を設計する必要があるという視点から授業方法を検討した。そして、「プログラミング入門」の授業では、少しずつ学ぶことを提示することにした。たくさんの新しいことを提示すると、認知的負荷が高くなり、学習効果が低くなると思われるからである。

そのため、2018年度から使用するプログラミング環境を変更することにした。

### 5.1 プログラミング環境

#### (1) 2017年度: プログラミング言語C

2017年度までは、Windowsで、コマンドラインツールを用いて、プログラミング言語Cを教えていた。これは、2016年度まで教えていた前任者のやり方を踏襲したものである。具体的には、プログラムを記述するにはWindowsで動作するエディタTeraPad<sup>16)</sup>を用いていた。TeraPadは、日本で広く用いられていたエディタで、日本で利用されている文字コードや改行コードに対応している。また、さまざまなプログラミング言語に対応したモードを持っている。記述したプログラムはMicrosoft Visual Studio 2015の「開発者コマンドプロンプト for VS2015」を用いて、CとC++のコンパイラである「1c」コマンドを実行してコンパイルした。そして、生成された実行型ファイルを実行していた。

これはMS-DOSの時代から行われてきた方法を、時代に合わせてアップグレードしたものと言える。しかし、はじめてプログラミングを学ぶ学生にとって、エディタの使い方、コマンドプロンプトにおけるcdやmkdirなどのMS-DOSコマンドの使い方、コンパイルの実行、エラーメッセージの理解、コンパイルしてできた実行型ファイルの実行など、新しいたくさんを最初に学ばないと使えないという問題があった。実際、授業で説明し、くりかえし問題を解く中で練習してもらったが、数週間経っても習得できない学生がたくさん現れた。そこで、実技試験と称して、教員やサポート学生の目の前で、1行文字列を表示するだけのプログラムを記述してもらい、それをコンパイルして実行させた。実技試験も1回の実施では、全員合格せず数人が落第した。そして、再試験を行なって習得してもらった。これはプログラミングを行う環境が初学者向けではなかったことを意味する。

また、プログラミングをはじめて学ぶ学生にとって、プログラミング言語Cは難しいと思われた。2016年度までは、教科書に『入門C言語』<sup>17)</sup>を用いていた。この本は形式的にしっかりしているが、いきなり新しいことをたくさん提示しているため、認知的負荷が高いと思われた。そこで2017年度は少しずつ学習内容を提示していくスタ

イルで書かれた『やさしいC 第4版』<sup>18)</sup>を教科書に採用した。それでも理解度は低く、定期試験では89人中35人の43.8%が100点満点で60点以下であった。

プログラミング言語Cのプログラムを理解するには、前提知識が必要である。たとえば、プログラミング言語Cの開発者自身による有名な本『プログラミング言語C 第2版』<sup>10)</sup>に登場する最初のプログラムを現代風書き直したものが以下である。

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");

    return 0;
}
```

このプログラムを実行すると「hello, world」に続けて改行を出力して終了する。つまり、プログラムが意識的に実行したい内容は「printf」ではじまる1行に書かれた内容だけである。しかし、これを実行するために必要な記述を前後に5行も書かなければならない。つまり、暗黙に必要とされることを理解して記述する必要があるということである。これは初心者には大きな認知的負荷を与えることになる。

## (2) 2018年度以降: Processing

そこで、2018年度以降は、開発環境をProcessingに変更した。ProcessingはMITメディアラボで、ケイシー・リースとベン・フライが開発したプログラミング開発環境である。オブジェクト指向プログラミング言語Javaをベースにしている。プログラミング言語Cよりもモダンな仕様になっている。デザイナーやアーティスト向けのプログラミング環境で、ヴィジュアルな表現を簡単に記述することができる。また、プログラミング

初



図5 Processing統合開発環境

心者が学ぶのにも適している。たとえば、さきほど提示したプログラミング言語Cのプログラムの例と同等のものを記述すると以下ようになる。

```
println("hello, world");
```

プログラミング言語Cでは、実行したい内容の他にも数行書く必要があったが、Processingではシンプルに実行したい内容だけを書けばいい。また、プログラムの記述と実行も統合開発環境(図5)が提供されており、実行ボタンを押すだけで実行できる。エラーがあれば、必ずしも正しくはないが該当すると思われる箇所を示してくれる。

これらによりプログラミングの初学者の認知的負荷を下げる効果が期待できた。

## 5.2 取り上げた内容

2017年度のプログラミング言語Cでも、2018年度以降のProcessingでもプログラミングの入門という意味では共通であり、データの型、リテラル、制御構造、関数などといったものを授業では取り上げた。



ただし、若干内容に差異がある。プログラミング言語Cは標準入出力を利用して文字を読み書きするプログラム、Processingではウィンドウを表示してそこでインタラクティブでヴィジュアルな表現を行うプログラムを記述するのが普通だからである。

#### (1) 2017年度: プログラミング言語C

教科書には『やさしいC 第4版』<sup>18)</sup>を用い、次の内容を取りあげた。

- 標準入出力
- データ型
- リテラル
- 変数
- 制御構造
- 関数
- 変数のスコープ
- 配列
- 文字列

この他にとりあげるべき内容の候補として、ポインタ、構造体、ファイル入出力、プリプロセッサ、分割コンパイル、メモリ管理などがあるが、それは1年生後期の授業「プログラミング言語」で扱うことにした。

#### (2) 2018年度以降: Processing

教科書には、Processingの開発者が書いた『Processingをはじめよう 第2版』<sup>19)</sup>を用いた。そして次の内容を取りあげた。

- 描画
- データ型
- リテラル
- 変数
- 制御構造
- アニメーション
- イベント
- 関数
- 配列

この他にとりあげるべき内容の候補として、

座標変換、オブジェクト指向などがある。座標変換は数学の知識を必要とするためとりあげなかった。オブジェクト指向は、新しい概念がたくさん登場し、初学者には認知的負荷が高いため3年生の「情報工学応用実験Ⅱ」で扱うことにした。

### 6. 「プログラミング入門」の成績の変化

#### 6.1 定期試験の実施

「プログラミング入門」では、定期試験を毎年実施している。定期試験に先立って、授業では類似の問題を提示して演習し、解説を行なっている。試験は参考資料の持ち込み禁止で、90分間で行なっている。

#### 6.2 定期試験の内容

プログラミング言語CとProcessingではできることが異なり、授業内容も異なるので、それに対応した試験の内容も異なる。しかし、問うている内容とレベルは同じ程度にするつもりで作っている。もちろん、それは著者の思い込みである可能性は否定できない。それぞれの定期試験の内容は以下である。プログラミング言語Cでは標準入出力、Processingではヴィジュアル表現を扱っているところに違いがある。

#### (1) 2017年度: プログラミング言語C

1. 変数の宣言を記述
2. printf()とscanf()の書式文字列、四則演算を記述
3. if～else文を使ったプログラムの穴埋め
4. if文を使ったプログラムを記述
5. for文の繰り返しの記述を穴埋め
6. 配列の宣言と値を表示する穴埋め
7. 標準入出力の文字列を読み書きするプログラムの穴埋め
8. 関数の引数と戻り値を記述する穴埋め

#### (2) 2018～2020年度: Processing

1. 変数の宣言を記述



- 2.for文で繰り返した結果を記述
- 3.条件式を記述する
- 4.if～else文を使ったプログラムの穴埋め
- 5.長方形と円の描画
- 6.if文を使ったプログラムを記述
- 7.図形を移動させるアニメーションの穴埋め
- 8.関数を使う穴埋め
- 9.配列の宣言と値を表示する穴埋め

### 6.3 定期試験の結果

#### (1) 年度による平均の比較

プログラミング言語Cで授業を行なった2017年度は、定期試験の受験者が89人で、平均が100点満点換算で59.4点だった(図6)。Processingに変えてから2018年度は受験者72人で平均74.6点(図7)、2019年度は受験者78人で平均77.2点(図8)、2020年度は受験者80人で平均76.7点(図9)であった。

C言語で実施した2017年度の平均は、Processingで実施したいずれの年度とも有意水準95%で異なる。一方、Processingで実施した2018～2020年度の平均は有意水準95%で互いに異なるとは言えない。

つまり、C言語とProcessingでは平均は異なり、C言語の方が理解度が低い。Processingを用いた各年度の平均は異なるとは言えない。2020年度は授業を配信した教室もあったが、2019年度までと異なるとは言えなかった。

#### (2) 年度による不偏分散の比較

C言語で実施した2017年度の不偏分散828.9で標準偏差は28.6点だった(図6)。Processingで実施した2018年度が不偏分散380.8で標準偏差19.4点(図7)、2019年度が不偏分散309.8で標準偏差17.5点(図8)、2020年度不偏分散462.7で標準偏差21.4点(図9)である。

C言語で実施した2017年度の不偏分散は、Processingで実施した2018～2020年度のいずれの不偏分散とも有意水準95%で異なる。一方、Processingで実施した2018～2020年度の不偏分散は有意水準95%で互いに異なるとは言えない。

つまり、C言語とProcessingでは不偏分散は異な

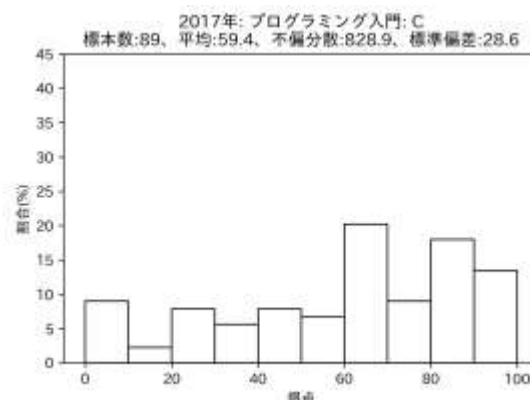


図6 2017年度の定期試験の成績分布

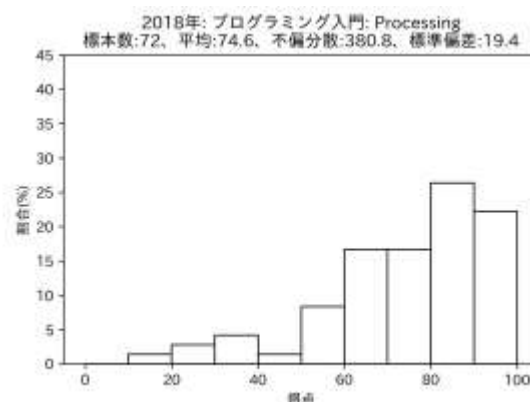


図7 2018年度の定期試験の成績分布

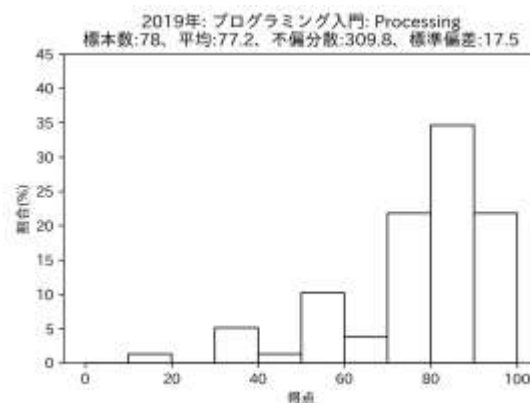


図8 2019年度の定期試験の成績分布

り、C言語の方が大きく、理解度のばらつきが大きい。Processingを用いた各年度の不偏分散は異なるとは言えない。2020年度は授業を配信した教室もあったが、2019年度までと異なるとは言えなかった。

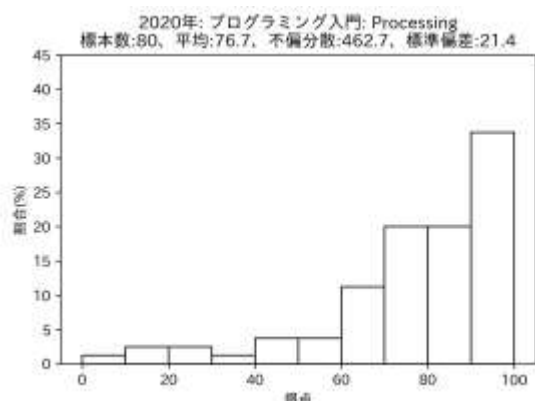


図9 2020年度の定期試験の成績分布

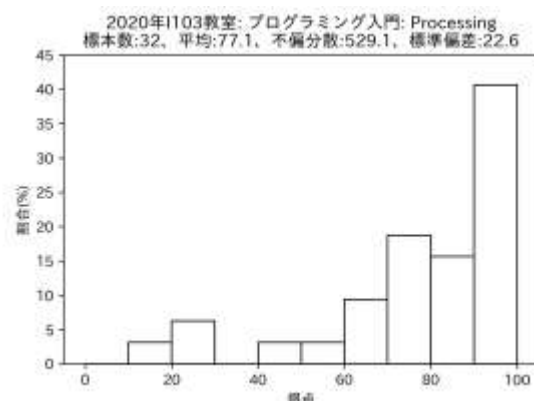


図10 2020年度I103教室の定期試験の成績分布

### (3) 対面と配信による平均と不偏分散の比較

2020年度は3教室に分けて実施したが、対面で授業を行なったI103は受験者が32人で平均77.1点(図10)、配信したI104は受験者34人で平均72.4点(図11)、I202が受験者19人で平均74.7点(図12)だった。これらは有意水準95%で互いに異なっているとは言えない。

また、I103の不偏分散は529.1で標準偏差22.6点(図10)、I104の不偏分散は599.7で標準偏差24.1点(図11)、I202の不偏分散は449.0で標準偏差20.6点(図12)だった。これらも有意水準95%で互いに異なっているとは言えない。つまり、対面の教室も配信した教室も理解度が異なっているとは言えなかった。

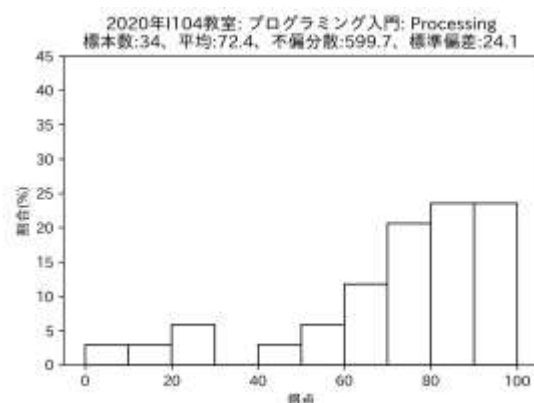


図11 2020年度I104教室の定期試験の成績分布

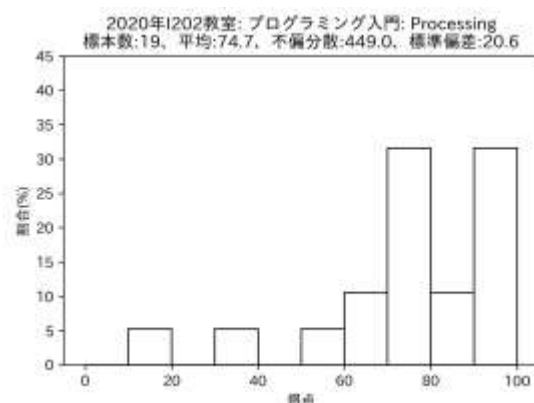


図12 2020年度I202教室の定期試験の成績分布

## 7.まとめ

八戸工業大学工学部システム情報工学科の「プログラミング入門」の授業は、2017年度までコマンドラインツールを用いてプログラミング言語Cを用いていた。2018年度からプログラム開発環境をProcessingに変更した。また、プログラミングの対象を標準入出力の文字列から、ビジュアルなものに変更した。これにより学生の成績の平均が向上し、不偏分散も小さくなった。

毎年、前の年よりも授業をよりよくしようと取り組んではいるが、Processingに変更してから3年間の成績は異なっているとは言えなかった。

また、2020年度は新型コロナウイルス感染症への対策で、「密」を避けるため、それまで1教室で実施していた授業を、3教室に分けた。そして1教室は対面で授業し、残り2教室に配信した。これらの教室の間で成績に違いがあるとは言えなかった。

それから、2020年度の入学者からノートパソコンを必携化した。これにより教室や授業の制約を受けにくくなった。折しも流行した新型コロナウイルス感染症への対応は容易になった。この変更により成績に違いがあったとは言えなかった。

## 参考文献

- 1) M. ガズダイアル, 「プログラミングを学ぶことはなぜそんなに困難か?」, 『Making Software: エビデンスが変えるソフトウェア開発』(Andy Oram, Greg Wilson 編, 久野禎子, 久野靖訳), オライリー・ジャパン, pp.107-120, 2011 年.
- 2) J. Bennedsen, M.E. Caspersen, "An investigation of potential success factors for an introductory model-driven programming course", Proceedings of the first international workshop on computing education research, pp.155–163, 2005.
- 3) E. Soloway, J. Bonar, et al, "Cognitive strategies and looping constructs: An empirical study", Communications of the ACM 26(11) pp.853–860, 1983.
- 4) M. McCracken, V. Almstrum, et al., "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students", SIGCSE Bulletin 33(4) pp.125–180, 2001.
- 5) G. Lewandowski, D.J. Bouvier, et al., "Commonsense computing (episode 3): Concurrency and concert tickets", Proceedings of the third international workshop on computing education research pp.133–144, 2007.
- 6) A. Forte, M. Guzdial, "Computers for Communication, Not Calculation: Media As a Motivation and Context for Learning", Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS '04) 4: pp.40096.1, 2004.
- 7) A.E. Tew, C. Fowler, et al. "Tracking an innovation in introductory CS education from a research university to a two-year college", Proceedings of the 36th SIGCSE technical symposium on computer science education pp.416–420, 2005.
- 8) R.H. Sloan, P. Troy, "CS 0.5: A better approach to introductory computer science for majors", Proceedings of the 39th SIGCSE technical symposium on computer science education pp.271–275, 2008.
- 9) B. Simon, P. Kinnunen, et al., "Experience Report: CS1 for Majors with Media Computation", Paper presented at ACM Innovation and Technology in Computer Science Education Conference, June 26–30, in Ankara, Turkey, 2010.
- 10) B.W. カーニハン, D.M. リッチー(石田晴久訳), 『プログラミング言語 C 第2版』, 共立出版, 1989 年.
- 11) B. フライ, C. リース(中西 泰人監訳, 安藤 幸央, 澤村 正樹, 杉本 達應訳), 『Processing』, ビー・エヌ・エヌ新社, 2015 年.
- 12) 「手すり・ドアノブ消毒は徹底したのに…大分の院内感染、盲点になった感染経路」, 読売新聞オンライン 2020 年 4 月 8 日, <https://www.yomiuri.co.jp/national/20200408-OYT1T50171/> (2020 年 12 月 7 日アクセス).
- 13) 「同じ端末のキーボード共有、クラスター原因か…ボタンに「高さ」あるタイプ」, 読売新聞オンライン 2020 年 12 月 7 日, <https://www.yomiuri.co.jp/national/20201207-OYT1T50081/> (2020 年 12 月 7 日アクセス).
- 14) Google, "Google Meet", <https://meet.google.com/>, (2020 年 12 月 7 日アクセス).
- 15) J. Sweller, John, "Cognitive Load During Problem Solving: Effects on Learning", Cognitive Science. 12 (2) pp.257–285, 1988.
- 16) 寺尾 進, 「TeraPad 公式サイト」, <https://tera-net.com/>, (2020 年 12 月 7 日アクセス).
- 17) 後藤 良和, 高田 大二, 中島寛和(寛 捷彦, 石田 晴久監修), 『入門 C 言語』, 実教出版, 2014 年.
- 18) 高橋 麻奈, 『やさしい C 第4版』, SB クリエイティブ, 2012 年.
- 19) C. リース, B. フライ(船田 巧訳), 『Processing をはじめよう 第2版』, オライリー・ジャパン, 2016 年.

## 要 旨

八戸工業大学工学部システム情報工学科では、2017 年度までプログラミング入門の授業においてコマンドラインツールを使ってプログラミング言語 C で教えていた。2018 年から Processing を使って教えるように変更したところ、学生の成績が向上した。

また、2020 年度は新型コロナウイルス感染症対策として、それまで 1 教室で実施していた授業を 3 教室に分けた。そして 1 教室は教員が対面で授業を実施し、残り 2 教室へはそれをビデオ会議システムで配信した。この変更により学生の成績に違いがあったとは言えないという結果が得られた。

**キーワード:** プログラミング入門教育, プログラミング言語C, Processing, 新型コロナウイルス感染症対策