

Processingにおけるプログラミング初学者向け外部入力機能の開発と評価

三浦 兼¹ 植野 俊治¹ 西田 誠幸¹

概要：Processing は、グラフィックスを容易に操作可能なメディアアート向けの言語である。Java の構文を基盤とした比較的簡潔なプログラムでの図形描画が可能であり、プログラミング導入教育で利用されている。しかし、if 文の構文を有効に利用した問題が提示しづらい。そこで本稿では、if 文の必要性の理解に外部入力機能の利用が有効であると仮定し、既存の外部入力機能に加えて Processing 初学者向けの外部入力機能を提案する。また、既存の外部入力機能と提案機能を用いて比較実験を行った結果を報告する。その結果、if 文の理解度に差は見られなかったが、比較実験の終わりに実施した確認テストにおいて提案機能を使用したグループの正答率が僅かに高い結果となった。また、提案機能を用いたグループとコマンドライン引数を用いたグループ両方において、一部の参加者に if 文の必要性の理解度の向上が見られた。

キーワード：Processing, 外部入力機能, if 文

1. はじめに

Processing は、グラフィックスを容易に操作可能なメディアアート向けの言語である。Java の構文を基盤としているが比較的簡潔なプログラムによって図形を描画することができる。また、実行操作のほとんどが GUI 上で完結しており、プログラミング導入教育で使用されることがある。

しかし、Processing による導入教育では if 文の必要性が分かりづらいという問題が存在する。これは、提示される問題やサンプルプログラムに if 文や変数を必要としないものが多く、分岐処理を有効に使用できていないためである。この問題に対して、プログラムの実行中に入力した値を if 文の条件式内で用いるようなプログラムを提示できれば実行後に動作の変化を与えることができ、初学者にとって if 文の必要性が理解しやすくなるのではないかと考えた。

そこで、本稿では Processing のプログラムに対して GUI で外部入力を行える機能を提案する。この提案機能と、既存の CUI による入力機能を比較する実験を行うことで、初学者に対する導入教育という観点で if 文の必要性の理解に優劣が生じるかを明らかにする。

2. Processing における if 文の学習

2 章では、外部入力機能を使用せずに if 文を学習する際

の問題について具体的なプログラムを例として説明する。次に示す 2 つのサンプルプログラムはどちらも if 文の構文を学習することが可能だが、if 文の必要性が分かりづらいプログラムの一例である。

2.1 if 文の条件式に外部入力値を使用しないプログラム

図 1 は Processing 学習ノート [1] で例題として掲載されているソースコードとその実行例である。変数 i の値が 1 以上であれば背景を赤く塗りつぶすプログラムだが、変数の値がソースコードに埋め込まれている。このことによりプログラムの実行結果が予測可能で、if 文を使用しなくても同じ画面が表示可能であるため初学者が混乱する可能性がある。

2.2 if 文と for 文を組み合わせたプログラム

図 2(a) のソースコードは東京工芸大学の 2019 年度のメディアプログラミング演習 I で講義資料に掲載された、if 文を扱ったプログラムの例である [2]。このプログラムでは横一列に円を等間隔で描画し、画面の半分を過ぎると円の色が赤くなるという処理を行っている。for 文の繰り返しの中で変化する値を条件式に使用しており if 文の使用方法として有効な例の一つだが、if 文を使用しないプログラムでも同様の表示結果を得ることが可能である。図 2(b) のソースコードは if 文を使用せずに同様の図形表示を行うように編集したプログラムである。左右どちらのプログラム

¹ 拓殖大学工学部情報工学科
Takushoku University, Tokyo 193-0985, Japan
snishita@cs.takushoku-u.ac.jp

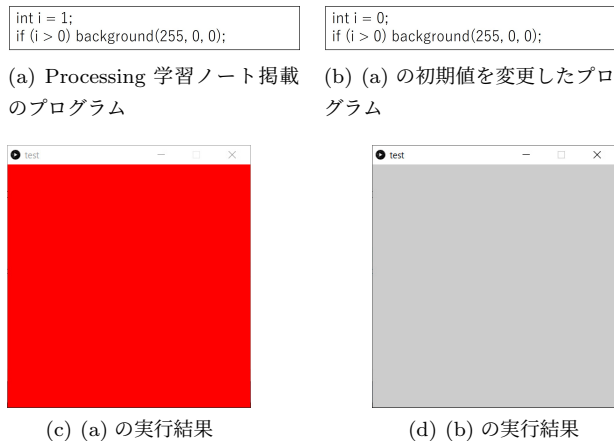


図 1 Processing における if 文の基本的な使用例

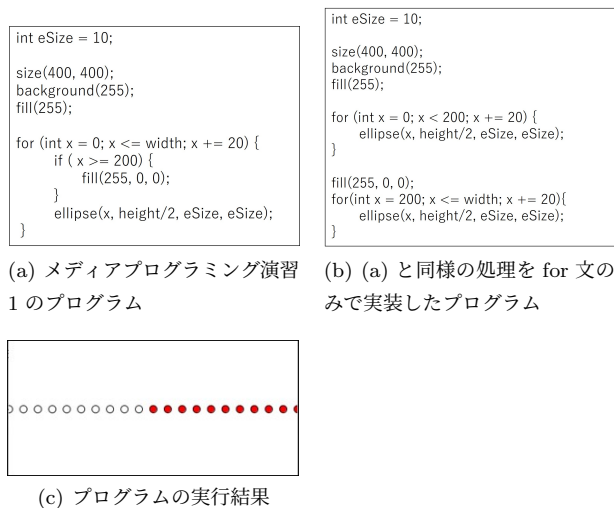


図 2 if 文と for 文を組み合わせた使用例

も実行すると同じ実行結果が得られるため、if 文を使用する必要性が伝わりづらい。

3. 関連研究

Processing と類似した図形描画機能を取り扱うプログラミング導入授業において if 文の理解度に関する問題が見られた例として、米屋らによる研究 [3] が挙げられる。この文献では Ruby 言語とお絵かき教材を使用した授業における効果的な学習方法の提案を目的として、学生が解いた課題の答案やアンケートから各学習項目ごとに理解度の調査を行っている。その結果として、アンケートの有効回答者の半数が分岐について理解できていなかったことが報告されている。これに対して米屋らは、分岐を使用する演習が絵に直接結びつくものになっていないとして、先にループを学習した上でその中に分岐構造を取り入れる演習を実施することで、学生に分岐の活用と理解を促すことができると考察した。しかし、この研究では外部入力機能について言及していない。

本研究では、米屋らによって報告された if 文に対する理解度の低さの原因がその必要性の理解不足によるものだと仮定し、上記研究では触れられなかった外部入力機能の利用によって if 文の理解度向上をねらう。

4. Processing における値の入力方法

2章で述べた外部入力機能を使用しない場合の問題点と、3章で紹介した外部入力機能を使用していない授業での学習理解度の調査結果から、if 文の必要性を伝えるためには実行時に取得した値によって動作が切り替わるプログラムを例示することが重要であると考えた。Processing において外部から入力を与える既存の手段や拡張機能としては、下記の 3つが挙げられる。

4.1 コマンドライン引数

コマンドライン引数を使用することにより、プログラム実行時にターミナルから値を複数入力し指定の配列から使用することができる。

コマンドライン引数を用いたプログラム例とその実行結果をプログラム 1、図 3 に示す。プログラム 1 では、6 行目でコマンドライン引数として入力した文字列を int 型に変換して変数 n に代入し、8 行目で n の値を if 文の条件判断に使用して円か四角形のどちらかを表示する。値の入力のため、コマンドプロンプトからの実行時にコマンドライン引数を指定している (図 3 (a))。プログラム 1 の 6 行目では入力値の変換と変数への代入を行っているが、配列 args に対する値の格納についてはプログラム内で明示されていない。このことから、プログラム実行時に入力した数値が if 文の動作にどのように対応しているのかが初学者には分かりづらいと考えられる。また、使用にあたって配列の知識が求められることに加えて、配列 args の宣言がプログラム内で明示されていないことから、コマンドライン引数の利用に不慣れな学習者が混乱する恐れがある。

4.2 イベント機能の利用

イベント機能を使用することでマウス座標やキーボードの押下情報などからプログラム実行中に値を取得することが可能である。しかし、学習者には事前にイベント駆動方式について理解することが求められる。

プログラム 2 にマウスイベントを使用したプログラムの例を示す。このプログラムではユーザーがマウスボタンをクリックした際に int 型の変数 x に座標を格納し、クリックされた位置が画面の右半分であれば円を表示するプログラムである。図 4 に実行画面の様子を示す。初めは (a) のように画面に何も表示しておらず、画面の右側をクリックすると (b) のように円を表示している。

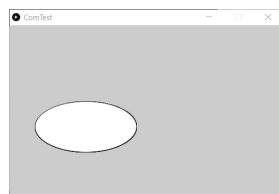
プログラム 2 では、Processing の基本的なメソッド 2 つに加えてイベントハンドラの記述が追加されている。こう

プログラム 1 コマンドライン引数を使用したプログラム

```
ComTest.pde
1 void setup(){
2   size(540, 340);
3 }
4
5 void draw(){
6   int n = Integer.parseInt(args[0]);
7
8   if(n<50){
9     ellipse(150, 200, 200, 100);
10  }
11  else{
12    rect(150, 200, 200, 100);
13  }
14 }
```

```
application.windows64
windows64>ComTest.exe 30
```

(a) 30 を入力してプログラム 1 を実行するコマンド



(b) (a) のコマンドの実行結果

```
application.windows64
windows64>ComTest.exe 100
```

(c) 100 を入力してプログラム 1 を実行するコマンド



(d) (c) のコマンドの実行結果

図 3 コマンドライン引数を使用した実行例

したプログラムの記述方法も含め、学習者にはイベント駆動方式を理解することが求められる。また、この例ではマウスカーソルの座標を入力値としており具体的な数値が扱えないため、if 文を使用する場合には閾値の確認が困難になる。

4.3 GUI ライブラリ

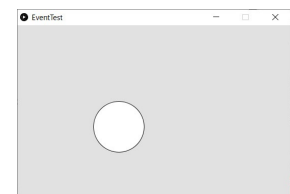
Controlp5[4]などのGUIライブラリを用いることで、テキストボックスやダイアログ等のインタフェースによる入力処理が実装可能である。しかし、変数を用いた入力値の受け取り処理の記述方法が一般的な代入処理と異なることや、数値を扱う際に型変換が必要になることから、入力値を分岐処理の判断に用いるまでの流れが把握しづらくなる。そのため、初学者が外部入力機能の存在とif文の必要性を併せて理解することは難しい。

プログラム 2 マウスイベントのプログラム例

```
1 int x;
2 void setup(){
3   size(540, 340);
4   x = 0;
5 }
6
7 void draw(){
8   background(225);
9   if(x>270) ellipse(200, 200, 100, 100);
10  }
11
12 void mousePressed(MouseEvent e){
13   x = e.getX();
14 }
```



(a) マウスクリック前の画面



(b) マウスクリック後の画面

図 4 プログラム 2 の実行画面

4.1 節から 4.3 節の入力方法には以上のような問題がある。これらの方法の問題点を受けて、本稿では事前に要求される予備知識が少ない、新しい外部入力機能を導入する。

5. 研究方針と入力機能の考案

5章では、「学習に使用する入力機能によってif文の必要性の理解度に優劣が生じるか確かめる」という目的に合わせて立てた研究方針と、設計した外部入力機能について述べる。

5.1 研究方針

2章のような入力機能を使用しない環境での学習において、if文の必要性を理解するにはプログラムの動作中に値を与えられるような機能の存在が重要である。そこで、4章で紹介した既存の外部入力方法における問題点を踏まえて新しい外部入力機能を開発し、他の入力機能を用いた学習方法と比較してどちらがif文の必要性をより理解できるのか調査することを目的とした比較実験を行う。

5.2 提案する外部入力機能

本稿で提案する外部入力機能は次の特徴を持つものである。

- (1) Processing 実行ウィンドウ内に入力インタフェースを用意する。
- (2) ソースコード内での構文一つの記述に対して値を一つ

プログラム 3 提案する入力機能を用いたプログラム例

```

1 //考案した入力機能の構文
2 readInt("入力値の用途などを示す文字列");
3
4 //入力値を変数に代入する
5 int n = readInt("n の値");
6
7 //入力値をメソッドの引数として指定する
8 ellipse(100, 100, 30, readInt("縦の直径"));

```

入力できる。

(3) 一つの入力ごとに入力待ちが発生する。

(4) 多くの予備知識を必要としない。

このうち、1 については入力ダイアログやテキストボックス、入力決定ボタンなどのインタフェースを Processing 実行ウィンドウ上に描画する。2 については、変数への代入値やメソッド呼び出し時の引数として本機能の構文を記述するものとす。3 については、本機能の構文を複数記述している際、入力インタフェースが実行順に一つずつ表示されるものとす。4 については、入力機能の利用に必要な記述を一行に留め、引数についても各インタフェースを区別する名称の文字列のみとすることで容易に記述可能な構文とす。プログラム 4 に構文の記述例を示す。

5.3 既存の値取得方法との相違点

4 章で解説した既存の値取得方法とは、それぞれ以下の点で異なっている。

- コマンドライン引数による入力方法では配列と型変換が必要になるが、本機能の前提知識は基本型の変数の知識だけでよい。
- マウス等でのイベント入力ではイベント駆動方式に関する知識が必要となるが、本機能では基本的な手続き型言語の実行順序に従って構文を記述できる。
- GUI ライブラリを使用した入力では変数を用いた値の受け取りに独自の記述を用いるが、本機能では一般的な代入文として構文を記述する。

提案入力機能は、入力のための構文 1 つを新たに学習することが必要である。しかし、構文は単文で記入できること、動作が比較的分かりやすいことから、配列やイベント処理に比べて学習にかかる負担を小さく抑えることが可能である。

5.4 提案入力機能の動作例

提案入力機能の使用例をプログラム 4 に示す。

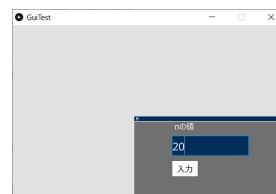
プログラム 4 の入力ダイアログを表示して入力待ちする(図 5(a))。ユーザ入力が完了すると入力値を変数に代入する。9 行目では入力値 n を if 文の条件判断に使用して、円か四角形のどちらかを表示する。 n が 20 のときに円を表

プログラム 4 ソースコードの記述例

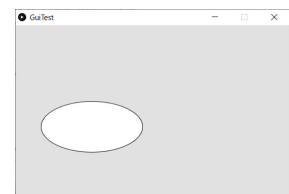
```

1 void setup(){
2     size(540, 340);
3 }
4
5 void draw(){
6     background(225);
7     int n = readInt("n の値");
8
9     if(n<50){
10        ellipse(150, 200, 200, 100);
11    }
12    else{
13        rect(150, 200, 200, 100);
14    }
15 }

```



(a) 7 行目で入力ダイアログを表示



(b) $n \geq 50$ のときに円を描画表示

図 5 プログラム 6 の実行例

示す図 5(b) に示す。

本稿で提案するインタフェースの構文では一文の記述につき一つの値を入力する。ソースコード内に構文を複数回記述した場合は実行順に一つずつインタフェースが表示され、値が入力されるごとに次のインタフェースに切り替わる。また、各インタフェースの表示ごとに入力待ちを発生させて、入力のタイミングとプログラム実行の順番を明示する。この機能により、入力値と if 文の動作変化の関係がより意識しやすくなることが期待される。

6. 評価実験

6 章では、初学者が学習に使用する入力機能の種類によって if 文の必要性の理解に差が生じるかを明らかにするための評価実験の実施方法について述べる。

6.1 実験の参加対象者

実験の参加対象者は、if 文の必要性に対する理解度の変化を調査するため、拓殖大学の情報工学科で開講しているプログラミング初学者向け科目を受講中の 1 年生とした。この科目は情報工学科の学生がプログラミングを学習する最初の科目であり、Java 言語を使用して講義を行っている。そこで、受講生を対象に実験への協力を呼びかけ、6 名の参加により実験を実施した。また、事前に理解度を測

る質問をした結果、6名のうち3名は実験開始時点でif文の必要性と入力機能の関係について理解していなかったと判断した。

6.2 評価実験の内容

評価実験では、プログラムの編集や動作確認などを資料に沿って参加者に各自で進めてもらう。実験の大きな流れを以下に示す。

- (1) Processingの演習(35分)
- (2) 確認テスト実施(15分)
- (3) インタビュー

演習では、参加者が本稿で提案する入力機能を使用するグループとコマンドライン引数を使用するグループの2つに無作為に分けた。各グループには使用する入力機能に合わせた資料を配布し、各自で演習を行ってもらった。

確認テストでは、参加者にif文の必要性についてテストを受けてもらった。

評価実験の最後に、確認テストの解答について詳細な分析を行うために口答でインタビューを行った。全参加者共通の質問として各テスト問題の解答理由を、不正解の解答についてはより掘り下げて質問を行った。

これらの調査結果からグループごとにif文の必要性の理解度を比較し、優劣が生じているかどうかを分析する。

6.3 演習資料

評価実験では、参加者にProcessingに関する基本動作からif文に関する内容までを一から学習してもらう。これは参加者が普段の授業でJava言語を使用しており、Processingの学習経験がないためである。演習資料は使用する外部入力機能によって2種類存在し、ともに以下の0章から4章で構成されている。

0章 基本操作

1章 図形描画

2章 外部入力機能

3章 変数

4章 条件分岐(if文)

6.4 確認テスト

確認テストは、「外部入力機能の存在によるif文の必要性」の理解を確かめるために実施する(付録A.1参照)。問題は大きく分けて以下の3種類を出題した。

- (1) if文の利用が適した処理について問う問題(多肢選択問題)
- (2) 外部入力とif文の動作関係を問う問題(4択問題)
- (3) if文が必要な処理を考えて記述する問題(記述式)

問題(1)は、if文の利用が適していると考えられる処理を5つの選択肢から全て選ぶというものである。ここでは、if文が必要になる場面について知識的に理解している

表1 確認テストの解答結果

| 参加者 | 使用した入力機能 | (1)if文の利用が適した処理 | (2)if文の動作確認 | | | (3)if文を活用した処理 |
|-----|----------|-----------------|-------------|-------|-------|---------------|
| | | | (2-1) | (2-2) | (2-3) | |
| Aさん | 提案入力機能 | A,B | A | C | A | 分岐 |
| Bさん | 提案入力機能 | A,B | A | C | B | 入力,分岐 |
| Cさん | 提案入力機能 | A,B | A | C | A | 入力,分岐 |
| Dさん | コマンドライン | A,B | A | C | A | 入力,分岐 |
| Eさん | コマンドライン | A,B,C | A | C | A | 分岐 |
| Fさん | コマンドライン | A,B | A | C | A | 分岐 |

か確認することを目的としている。

問題(2)は3つの小問で構成されており、いずれもプログラムの実行により描画する図形を予想する問題である。外部入力値を条件式とするif文があるプログラムも出題されるが、この場合の正解選択肢として「描画する図形は実行するまで分からない」を置いている。

問題(3)はif文の機能を活用した処理を考えて記述する問題である。ここでは参加者がif文の機能を活用するのに、入力機能の利用を含めた解答を自発的にするかどうかを確かめる。

演習後にこれらの問題を出題し、その解答結果から学習理解度の検証を行う。各問題ごとに出題意図が異なるため、単純な正答数だけではなくグループごとの解答傾向や選択内容の組み合わせなどから考察を行う。

7. 実験結果

7章では、評価実験の結果とその考察を述べる。本稿では本学情報工学科のプログラミング1を受講している1年生に対して実験参加を呼びかけ、内6名を参加者として実験を実施した。6名の参加者に対しては、if文の必要性と入力機能の関係について事前に理解度の調査を行った。その結果、実験前の時点でAさん、Bさん、Fさんはif文の必要性と入力機能の関係を理解していたと分析した。またCさん、Dさん、Eさんは同内容について、実験前には理解していなかったと分析した。

7.1 確認テストの結果

確認テストの結果を表2に示す。赤く色づけされた要素は不正解、不完全な解答である。

7.1.1 問題(1)の結果

問題(1)はif文の利用が適していると考えられる処理を5つの選択肢から全て選ぶものであり、AとBの2つを選んでいる人が正解となる。AとBの選択肢はそれぞれ、分岐を使用した処理、入力値を条件に用いた分岐処理となっている。ここではEさんを除く5人が正解となっており、不正解のEさんについてもAとBの選択肢が選ばれている。

Eさんの解答についてインタビューを行った結果、Cの選択肢を選んだ理由は数値の比較による分岐を行うからだとの回答があり、文章の意味を読み違えていたと見られる。また、その他の5人については分岐を意識して解いたとい

う回答があったものの、入力に関する理由を含めた参加者はいなかった。

結果として、両グループで知識的な理解度に大きな差は見られなかった。

7.1.2 問題 (2) の結果

問題 (2) は 3 つのプログラムを読んでそれぞれ表示される図形を選択肢から選ぶものであり、2-1、2-2、2-3 の順に A、C、A が正解となる。2-1 と 2-2 については全員が正解しており、2-3 については B さんが誤った選択肢 B を選んでいた。

インタビューでは、D さんが 2-2 においてコマンドライン引数を格納している `args[0]` という配列の意味がよく分かっておらず、内容を理解しない状態でたまたま正解の C を選んだという回答が得られた。また、B さんの解答についても質問をし、2-3 の結果がケアレスミスであることが分かった。

テストの解答とインタビューの結果、問題 (2) については D さんを除く 5 人が入力と if 文の動作関係を理解していることが分かった。

7.1.3 問題 (3) の結果

問題 (3) は if 文の機能を活用した処理を考えて記述するものであり、解答内容に入力と分岐に相当する記述があるかという点で分析を行う。結果、B さん、C さん、D さんの解答には入力と分岐の両方について記述があり、A さん、E さん、F さんの解答には入力に関する記述がなかった。解答理由についてインタビューを行った結果、6 人とも外部入力について意識している様子は見られなかった。

結果としては、両グループでばらつきがあるものの自作入力機能を使用したグループの方が入力を含めた解答者は多かった。

7.2 考察

まず本稿における調査項目としていた「学習時に使用する外部入力機能の種類によって if 文の必要性の理解度に優劣が生じるか」という点について、調査結果からは入力機能による理解度の明らかな差は見られなかった。

確認テスト (1) では不正解の E さんを含めた全員が正解の選択肢を含んだ解答をしていたが、解答理由を質問したところ外部入力を意識して答えたという発言は誰からも得られなかった。確認テスト (2) の結果とそれぞれの解答理由から、D さんを除く両グループの参加者が入力機能を用いた際の if 文の動作の不確定性については理解していることが分かった。D さんは問題 2-2 について、入力処理に使われている配列 `args` が理解できず無作為に解答を選んだ結果正解となっている。しかし、確認テスト (3) では入力を含めた解答を行っていることから、無意識ではあるが if 文の活用と入力機能を併せて理解できたと考えられる。確認テスト (3) では全員に解答理由を質問したが、外部入力

について意識している人は誰もいないことが分かった。

各グループの正答率としては自作入力機能を使用したグループの方が高い傾向にある。しかし正答率に大きな差がないことや、インタビューにおいて外部入力に関する供述がなかったことから、各グループ間での理解度に明らかな差は見られなかった。

また、if 文の必要性と外部入力機能の関係を理解しているかという点では、確認テスト (2) と (3) の結果から、入力値を条件式に用いた際の if 文の動作については全員が理解できていると考えた。このことから、元々 if 文の必要性と入力機能の関係を意識していなかった人に対して、それらの理解を促進できたと考えられる。

考察のまとめとして、各グループの正答率としては自作入力機能を使用したグループの方が高い傾向にあるが、決定的な差が見られないことやデータ数の不足から各グループ間での理解度に優劣の差は見られなかった。しかし実験による学習結果として、if 文の必要性と入力機能の関係について参加者の理解が見られた。

また、今後の展開としてより多くの参加者を集めて実験を行うことや、こちらの想定していない解答意図が見られた評価問題の改善などが考えられる。

8. おわりに

本研究では、Processing による導入教育で if 文の必要性を理解するにあたって外部入力機能を用いた演習が効果的であると推測し、その種別によって理解度に優劣が生じるかを検証するために新しい入力機能を導入した。また、新しい入力機能と既存の値取得方法を使用した比較実験を実施し、得られたデータに対して分析と考察を行った。その結果として、各グループ間での理解度に明らかな差は見られなかったが、外部入力機能を用いた学習によって if 文の必要性と入力機能の関係性の理解が促進できることが示唆された。

参考文献

- [1] 吉村正照：Processing 学習ノート，入手先 (<https://www.d-improvement.jp/learning/processing/>) (参照 2021-8-26)。
- [2] 野口靖：2019 年度 メディアプログラミング演習 I，入手先 (<https://r-dimension.xsrv.jp/classes/j/programming-1st-2019/>) (参照 2021-8-20)。
- [3] 米屋美雪 内田奈津子：情報処理学会研究報告,2021.No.9, 分岐を学ぶために効果的な教材を考える-お絵描きプログラムによる入門科目を振り返って- 情報処理学会 (2021)。
- [4] AndreasSchlegel：processing GUI，controlP5，入手先 (<http://www.sojamo.de/libraries/controlP5/>) (参照 2021-8-20)。

付 録

A.1 確認テスト

A.1.1 問題 1

問題 1: if 文の利用が適していると考えられる処理を全て選んで、選択肢を丸で囲みなさい

- A それぞれ整数が代入された変数 a と変数 b を足した値が 20 以上かそれ未満かで異なる文章を表示する
- B 入力された値が 80 より少なければメッセージを表示する
- C 整数が代入された変数 x に対して 5 を足す処理と 5 を引く処理を同時に行う
- D 2 つの入力値を受け取って掛け算を行う
- E 乱数を 1 つ生成し、その 3 倍の数値を表示する

プログラム 6 問題 2-2

```

1 int n = readInt("n");
2 int xy = 50;
3 if(4 < n){
4     ellipse(200, 200, xy, xy);
5 }
6 else{
7     rect(200, 200, xy, xy);
8 }

```

A.1.2 問題 2

問題 2: 以下の 3 つのプログラムは条件分岐によって円か四角形のどちらかを表示するプログラムである。(プログラム 5, プログラム 6, プログラム 7) それぞれ何が表示されるか、選択肢を丸で囲みなさい。

- A 円
- B 四角形
- C 実行するまで分からない
- D 正解が分からない

A.1.3 問題 3

問題 3: if 文の機能を活用した処理を 1 つ考えなさい。(実行の動作や手順, 何が表示されるかなどについて日本語で記述, 実際のプログラムを書いて説明しても構わない)

プログラム 5 問題 2-1

```

1 int n = 50;
2 if(n < 100){
3     ellipse(0, 100, 15, 15);
4 }
5 else{
6     rect(0, 100, 15, 15);
7 }

```

プログラム 7 問題 2-3

```

1 int n = 14;
2 int xy = readInt("n");
3 if(n < 100){
4     ellipse(200, 200, xy, xy);
5 }
6 else{
7     rect(200, 200, xy, xy);
8 }

```
