

Processing 言語による手続き型プログラミング復習コンテンツの試作

廣川大樹[†] 土肥紳一[‡]東京電機大学大学院システムデザイン工学研究科[†]東京電機大学システムデザイン工学部[‡]

1. はじめに

本学のシステムデザイン工学部デザイン工学科では「コンピュータプログラミング I」という講義が開講されている。この講義では手続き型プログラミングの基礎を学習する。プログラミング言語は Processing を使用している。受講者の 70% がプログラミングを初めて行う。2019 年以降、新型コロナウイルスの影響を受けて、講義は Zoom で録画・配信されている。2022 年度に対面形式を再開したが、録画・配信は続行している。各授業で理解度調査アンケートを実施し、新しく教わった内容を理解できたか否かを「はい」か「いいえ」で回答する。その結果、「いいえ」と回答する受講者が見られた。彼らのような学習者は理解できなかった内容を復習するが、復習しても理解できない場合がある。プログラミングを復習しても理解できない学習者を対象に、筆者らは復習を支援するコンテンツを制作して、理解度向上を試みている[1]。

2. コンテンツの試作

復習コンテンツは PowerPoint を用いて制作した。コンテンツはプログラムと、プログラムの動作をイメージした図で構成した。プログラム上のオレンジ色の太字で強調した部分を図で表現する。さらに、強調部分をより強調するために、表現しない部分の文字の色は薄くした。図では対象となる図形の枠内をオレンジ色で塗りつぶし、値や文字はオレンジ色の太字で記述した。公開は Microsoft SharePoint で作成した Web サイトにて行った[2]。Microsoft SharePoint は Microsoft 社の提供するソフトウェア製品で、コンテンツを管理したりコミュニティで共有したりすることができる。コンテンツを利用した人数やデバイス別の利用状況などのデータを収集することが可能である。なお、本 Web サイトは復習を支援するため、講義実施日に該当するコンテンツを公開した。受講者 60 名に Web サイトへのアクセスを許可した。公開したコンテンツは変数、for 文、条件分岐、関数、配列である。

2.1 変数

変数を使って 2 つの四角形を表示するプログラムである。変数の宣言と値の代入、データ型の違いに

着目した。変数と値には四角形を用いた。値を代入する場面では、アニメーションを用いて値の四角形が変数の四角形の内部に収まるよう表した。データ型の違いについて、Processing では int 型の変数に浮動小数点数は代入できず、エラーが生じる。int 型と float 型の代入可能な値の大きさに着目し、int 型の四角形は float 型のものより小さくした。これによって、int 型の変数に浮動小数点数を代入するプログラムの場合、int 型の変数の四角形に浮動小数点数の四角形が収まらないよう表した。int 型の変数 x に浮動小数点数 25.0 を代入する場合の図とエラーの様子を図 1 に示す。

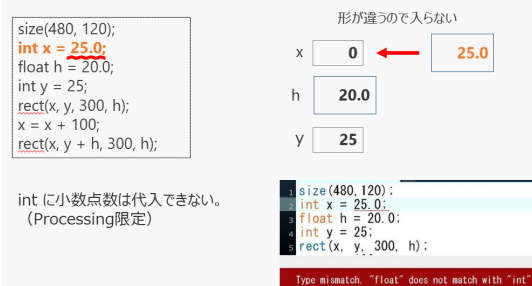


図 1 型が一致しない図とエラー画面

2.2 for 文

for 文を使って 7 本の直線を表示するプログラムである。制御変数 i の値が 380 のとき、7 本の直線を表示した実行結果の様子を図 2 に示す。

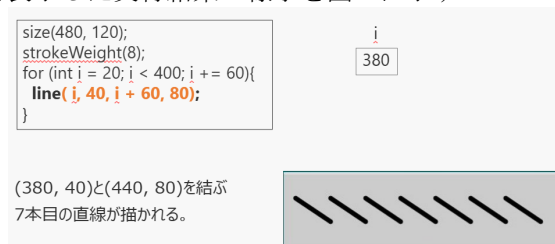


図 2 制御変数 i の値と実行結果

制御変数と実行結果に着目した。制御変数には四角形を用いた。繰り返す条件において、真偽を判定する場面では、true では青い比較演算子、false では赤い比較演算子で表した。プログラムを実行するとき、for 文の動作は学習者には見えない。そこで、1 回ごとの制御変数の値と実行結果を表示した。

2.3 条件分岐

if 文を使ってマウサーソルが画面右半分にあるときに四角形を表示するプログラムである。比較演算および論理演算による条件、boolean 型に着目した。boolean 型の変数および条件式には菱形を用いた。真偽を判定する場面では、カーソルが画面

Developing review contents of Procedural programming with Processing language

[†] Taiki Hirokawa, Graduate School of System Design and Technology, Tokyo Denki University

[‡] Shinichi Dohi, School of System Design and Technology, Tokyo Denki University

右半分にあるときに青い矢印を、画面左半分にあるときに赤い矢印を用いた。マウカーソルの位置が画面左半分と右半分にある場合の実行結果の様子を図3に示す。

```
void setup() {
  size(480, 120);
}

void draw() {
  background(255);
  if(mouseX > width/2){
    rect(width/2, 0, width/2, height);
  }
}
```

カッコ () 内の条件式を判定する。条件式がtrue (条件を満たす) のとき、ブロック内 { } のコードが実行される。

カーソルのx座標が幅の半分よりも大きい、つまりカーソルが画面右半分にある場合に実行される。

図3 マウカーソルによる図形表示の操作

2.4 関数

関数を自ら定義して火星上での体重を計算するプログラムである。プログラムの実行時に一度だけ処理する関数 `setup` 関数、火星上での体重を計算する関数が `calculateMars` 関数である。 `setup` 関数内の変数 `marsWeight` に、 `calculateMars` 関数内の変数 `newWeight` の値 50.16 を代入する様子を図4に示す。

```
float gravity = 9.8;
void setup() {
  float yourWeight = 132.0;
  println(yourWeight * gravity);
  float marsWeight = calculateMars(yourWeight);
  println(marsWeight);
}

float calculateMars(float w) {
  float newWeight = w * 0.38;
  return newWeight;
}
```

return によって、 `calculateMars` 関数内部の変数 `newWeight` の値 50.16 が、 `setup` 関数内部の変数 `marsWeight` に代入される。

図4 実引数による値の受け渡し

引数、返却値、および関数の型に着目した。変数には直角、関数には丸角の四角形を用いた。値の引き渡す・値の返却ではアニメーションを用いて変数の図形内に値の図形が収まるよう表した。

2.5 配列

配列を使って5つの円形を表示するプログラムである。変数 `x` が整数型の要素を持つ配列を参照する様子を図5に示す。

```
int[] x;
void setup() {
  size(200, 200);
  x = new int[5];
  x[0] = 30;
  x[1] = 110;
  x[2] = 190;
  x[3] = 270;
  x[4] = 350;
  for(int i = 0; i < x.length; i++){
    ellipse(x[i], height/2, 40+10*i, 40+10*i);
  }
}
```

図5 変数に配列への参照を代入する様子
配列の宣言と生成、要素への代入、および `length` フィールドに着目した。変数、および配列の要素と `length` フィールドには四角形を用いた。配列は丸角の四角形と鋸状の図形を接合した。参照情報は丸い

始点の矢印と鋸状の図形を組み合わせた。参照情報の代入では、アニメーションを用いて配列の図形と参照情報の形が一致するよう表した。

3. 結果

10月1日から28日までのWebサイトの利用者数の推移を図6に示す。

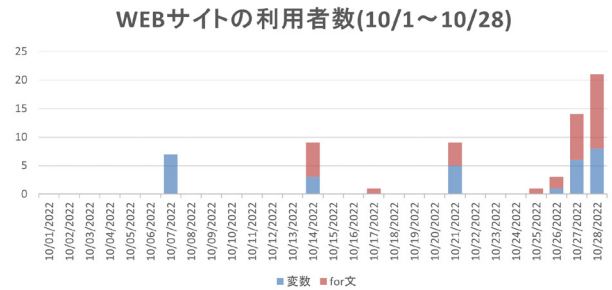


図6 Webサイトの利用者数(10/1~10/28)

10月28日に21名が利用した。中間審査が実施された日である。前日には14名が利用した。5名以上利用した日には授業が行われた。次に、12月1日から23日までのWebサイトの利用者数の推移を図7に示す。

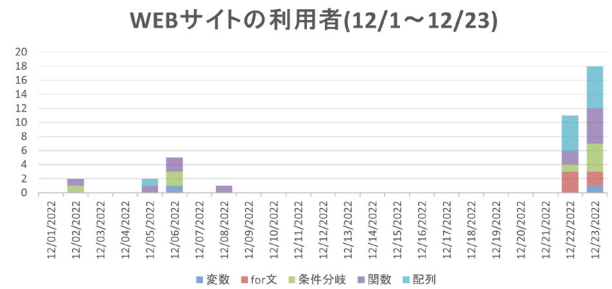


図7 Webサイトの利用者数(12/1~12/23)

12月23日に18名が利用した。この日、期末審査が実施された。前日には11名が利用した。

以上より、審査前日と当日に利用された。審査前に学習内容を確認するためだと考えられる。また、12月では授業実施日に利用されなかった。中間審査は受講者にとって初めてのプログラミングテストであり、不安だったため利用したと考えられる。

4. まとめ

本研究の目的は手続き型プログラミングの復習支援であり、対象は手続き型プログラミングを理解できていない学習者である。コンテンツを評価するためのクイズとアンケートを作成したが、回答を得られなかった。別の評価方法を検討する必要がある。また、利用者から回答を得る方法が課題である。

参考文献

- [1] 廣川大樹, 土肥紳一, Processing 言語による学習者のつまずきに着目したプログラミング復習コンテンツの試作. 情報処理学会, 全国大会講演論文集(4), pp.625-626, 2022
- [2] コンピュータプログラミング I Web サイト, <https://dohi.chiba.dendai.ac.jp/~dohi/computer-programming-1/ad/>, 2022年12月24日閲覧