教育情報学研究　第18号（2019）
Educational Informatics Research, 2019, No.18, 55-66

Web IDE のためのエラーメッセージを改善した初心者向けプログラミング学習支援システム

住田　智雄 *，小嶋　秀樹 *

** 東北大学大学院教育学研究科*

要旨：2020年以降，小学校から始まるプログラミング教育の必修化に向けて，多くのプログラミング学習支援システムが提案・運用されている．現在提案されている初心者向けのプログラミング学習支援システムの多くは，プログラムをテキストで記述するのではなく，視覚的にわかりやすく直感的にプログラミングができる Scratch のようなビジュアルプログラミング言語を利用している．しかし，ビジュアルプログラミング言語を利用したプログラミング学習支援システムでは，実際に利用されているプログラミング言語への円滑な移行が考慮されていないなどの問題点も指摘されている．そこで本研究では，初心者でもわかりやすい Processing を利用したプログラミング学習支援システムの構築を目標とする．Processing は実際に利用されている Java をもとに作られたプログラミング言語であり，初心者にもわかりやすく，本格的なプログラミングスキルを身に付けることができる．この Processing を Web 上で動作させるためには JavaScript に変換した Processing.js を利用する必要があるが，Processing.js のコンパイラは文法や構文の厳密なチェックを行わず，学習者に対して適切なエラーメッセージを提示しない．そこで本稿では，Processing.js のエラーメッセージの提示に関する欠点を改善するために構築した Processing を用いた Web IDE について説明する．

キーワード：プログラミング教育，プログラミング学習支援システム，Web IDE，エラーメッセージ，Processing

1　背景

情報化社会において，コンピュータやスマートフォンを利用したサービスは日常生活の中で必要不可欠なものになっている．コンピュータをより適切に，かつ効果的に活用していくスキルを身に付けることを目的として，図1に示すように2020年から「教育の情報化の推進」として情報活用能力の育成を重要視した新学習指導要領が施行される [1].

この2017年および2018年に公示された新学習指導要領の各学校種におけるプログラミング教育では，小学校の「プログラミング的思考の教育」として，実際のプログラミングを教育するのではなく，各教科での目標を達成するための問題解決能力を育成するプログラミング的思考を授業内で行うことになる．中学校の「計測・制御に関連するプログラミングの教育」はこれまで行われていた内容であり，さらに「ネットワークを利用した双方向性のあるコンテンツのプログラミング」が追加された．高等学校のプログラミング教育では，共通教科情報と専門教科情報で行われ，「より高度なプログラミングの教育」が求められる．さらに，情報と情報技術を活用することで問題発見と解決を探求する内容も含まれており，小学校から高等学校までのプログラミング教育の総合的な問題解決能力の育成が図られている [2].

このように，プログラミング能力の需要が高まっていく中で，この需要に応じるためにすでに多くの高等学校や文系の大学でプログラミング教育が行われている．そして，このプログラミング教育で使用されるプログラミング学習のための学習環境の代表として，図2に示すような Scratch が

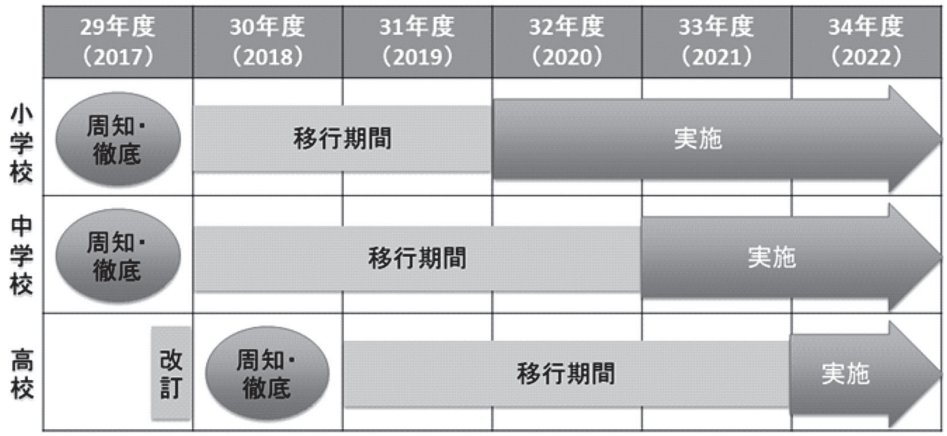

図1　学習指導要領改訂に関するスケジュール

ある [3] [4]．Scratch はビジュアルプログラミング環境の1つである．一般のプログラミング言語は文字をキーボードで入力していくが，Scratch では文字ではなくマウスを使ってブロックを組み合わせていくことでプログラミングしていく．視覚的にわかりやすく直感的にプログラミングできるため，プログラミングの初学者教育に適しているとされている．しかし，ビジュアルプログラミング言語によるプログラミング学習では，C 言語や Java，Python といった実際に利用されているプログラミング言語の文法や構文などに必要な概念を学ぶことが難しく，円滑な移行は考慮されていない [5]．そのため，Scratch では問題解決のためのプログラムを作成することができても，実際のプログラミング言語では作成することが難しいといった問題が生じる．

また，現在提案されているプログラミング学習支援システムは，学習者に対して問題演習を通した学習で支援するシステムや，学習者によるコンピュータの操作を画面で監視し，その動作からプログラムの理解度を把握しようとするものが多い．学習者が記述したプログラムを解析することで学習者個人のエラーの原因やエラーの傾向，理解度などを推論し，その後の学習に役立つような情報を提示するプログラミング学習支援システムはない．

そこで本研究では，入門者や初心者でもわかりやすいプログラミング言語である Processing を利用したプログラミング学習支援システムを Web IDE（Integrated Development Environment）として構築することを目標とする．対象とする学習者は，プログラミング学習の経験のない入門者や初心者，およびこれらの学習者に指導する教員である．Web IDE として開発することで，開発環境の構築に慣れていない初心者や入門者，教員の負担を軽減することができ，さらにインターネットを利用できる環境であれば，どこでも利用することができる．また，Processing は実際に利用されている Java をもとに作られたプログラミング言語である [6] [7]．学習者が動作結果を確認しやすいグラフィックやアニメーション，さらにサウンド機能などを簡潔に記述することが可能であり，初心者にもわかりやすく，本格的なプログラミングスキルを身に付けることができる．しかし，この Processing を Web 上で動作させるためには JavaScript に変換した Processing.js を利用する必要があるが，Processing.js のコンパイラは厳密な文法や構文のチェックを行わず，エラーが発生した場合に学習者に対して適切なエラーメッセージを提示しない．

本稿では，Processing.js のエラーメッセージの提示の欠点を改善するために構築した Processing を用いた Web IDE について説明する．

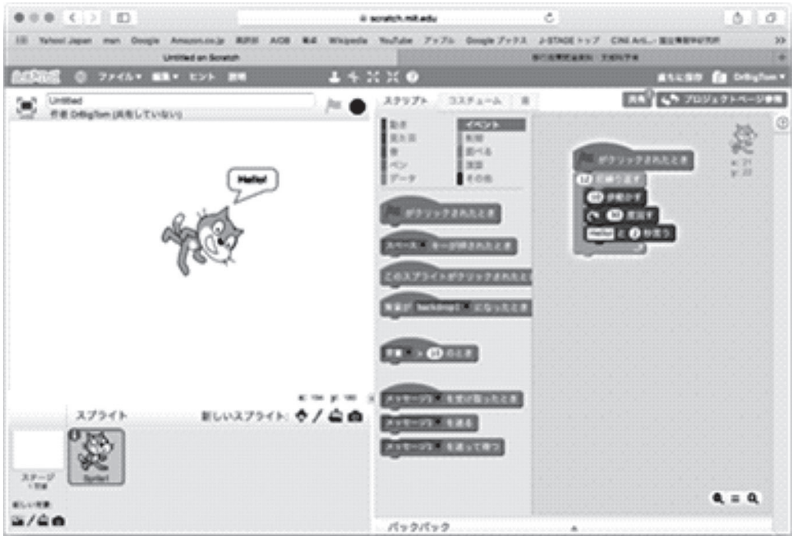

図2　Scratch

2　関連技術

本章では，本稿に関連する技術について説明する．

2.1　Scratch

Scratch は，アメリカの MIT で開発された無償で利用できるプログラミング環境である．図2に示すように，あらかじめ用意されているブロックを組み合わせていくことで簡単にプログラムを作成することができる．そのため，入門者や初心者を対象としたプログラミング学習で利用されることが多い．また，Web アプリケーションとして開発されているので，Web ブラウザから Scratch のサイトにアクセスすることで利用することができる [3] [4].

2.2　Processing と Web

本節では，Processing と Processing を Web で動作させるための手段について説明する．

2.2.1　Processing

本研究で対象としているプログラミング言語 Processing は，ビジュアルアートをプログラミングすることを目的として Java をもとに作成されたプログラミング言語である．2D や3D のグラフィック，アニメーション，サウンドなど初心者でも簡単に作成することができ，動作結果もすぐに確認することができる [6] [7]．また，Java のように本格的なプログラミングも可能であり，Processing を学ぶことで実用的なプログラミングスキルを身に付けることができる．そのため，C 言語や Java，Python など実際に利用されているプログラミング言語への円滑な移行も可能であると考えられる．このような Processing の特徴から，プログラミングの初学者教育だけでなく，メディア作品やそのプロトタイプ製作など，アーティスト，デザイナー，建築家などにも使用されている．ただし，Processing の開発環境をインストールしなければ使用できない．

2.2.2　Processing.js と p5.js

Processing.js は，Processing の文法で記述されたプログラムを JavaScript に変換して Web ブラウザで実行できるようにしたものである [8]．ほとんどの Web ブラウザで動作する簡単さと HTML や JavaScript など他の環境との連携のしやすさが特徴である．

p5.js は，Processing.js とは異なり，Processing の記述をできるだけ使用し，JavaScript としてプログラムを記述する言語である [9]．たとえば，

Processing で使用する関数である setup 関数や draw 関数の場合，「void setup（ ）」や「void draw（ ）」と記述する．しかし，JavaScript では関数の定義は function というキーワードで行うため，Processing の「void setup（ ）」は p5.js では「function setup（ ）」と記述しなければならない．また，実行結果を表示するウィンドウサイズを指定する場合，Processing では size 関数であるが，p5.js では createCanvas 関数となる．

このように，p5.js は通常の Processing とは記述が異なるため，Processing を利用した Web IDE を開発する場合に，学習者が使用するプログラミング言語としては適さない．そこで本研究では，文法や構文が Processing と同じ Processing.js を使用することとする．ただし，Processing.js のコンパイラは厳密な文法や構文のチェックを行わない．そのため，学習者がエラーを含むプログラムを作成した場合，適切なエラーメッセージを学習者に提示しないため，学習者の学習意欲等を妨げる可能性がある．

2.3 Web IDE

ブラウザのみで利用できる Web IDE は，プログラミング学習の環境構築の負担軽減が可能であるため，入門者や初心者に最適である．そのため，プログラミング学習のための Web IDE が研究されている．

Processing のコードをテキストではなく，視覚的なブロックで記述するという方針を取ったプログラミング教育の研究では，Scratch と同じくブロックの色や形状で視覚的なコーディングを可能としており，可読性を高める工夫が施されている [10]．また，現実的な環境で多数の学習者の指導を同時に行う必要性と C や Java などの実際に使用されているプログラミング言語のスキルを身につけさせたいという要請から構築された Processing を利用した Web IDE もある [11].

3 プログラムのエラーメッセージに対する学習者の意識

本研究で対象としている学習者（プログラミング経験のあまりない私立文系大学の学生）にプログラミング学習に関するアンケートを行った．このアンケートはプログラムの講義を履修している 3 ～ 4年生18名に対して，本研究の対象言語であるプログラミング言語 Processing の基本的な内容を理解し，自分で基本的なグラフィックを作成できるようになったあとに行った．

3.1 アンケート項目

アンケート項目は以下の通りである．

1. プログラムを学習したことがありますか
2. プログラムの基本的なことを学習してみて，難しかったですか
3. プログラムのどういうところがわかりにくかったですか（自由記述）
4. エラーメッセージがわかるとやりやすいですか
5. アプリを使ってみて，やりやすくなりましたか
6. アプリを使ってみてどうでしたか（自由記述）

このアンケート項目の結果の中から，本稿に関する項目について述べる．

3.2 関連するアンケートの結果

「プログラムのどういうところが難しかったですか（自由記述）」のアンケート結果を図3に示す．

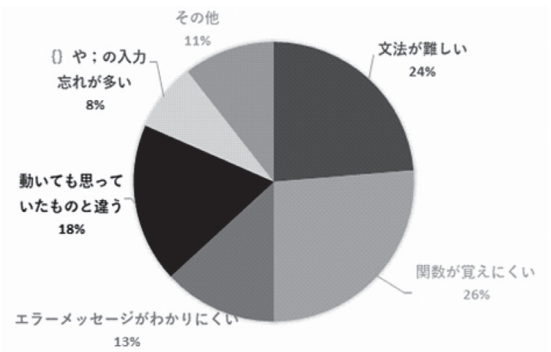

図3 プログラムのわかりにくいところ

プログラミングの経験がない学生が多いため，プログラムの基本的な部分で難しいと考える学生が多い．また，エラーに関連する内容としては，「「 ｛｝ 」や「 ; 」の入力忘れが多い」といった記述がある．「 ｛ 」と「 ｝ 」，「（ 」と「 ）」の数の一致や各関数の呼び出しは常に「 ; 」で終わるといった文法が混乱のもととなっている．またこ

のアンケートでは記述がなかったが，「｛」の次の行のインデントを行わない学生も見受けられる，これはプログラム全体の可読性が悪くなり，エラーの原因になると考えられる．

次に，「エラーメッセージがわかりにくい」という記述の原因の一つに，エラーメッセージが指摘するエラーの場所と実際にエラーがある場所が一致しないことが挙げられる．具体例として，図4に学生が実際に入力したエラーを含むプログラム例を，図5にこのプログラムの実行結果を示す．

```
size(500,500);

for(int x=20 ; x<500; x +=120);{
  for(int y =60 ; y <500; y +=180);{
    triangle(x,y,x+100,y,x+50,y+50); //
  }}
```

図4　エラーを含むプログラム例

次に，このプログラムを作成した学生の思考についてまとめてもらった資料から一部引用する．『変数はちゃんと入力しているのに「変数 “x” は存在しません。」と Processing は私に訴えかけるのだ。私は荒れ狂ったが、よくよく確認すると for 文のお尻に；（セミコロン）がいつもの癖でついていたのだ。私はプログラム初心者であるため、詳しいことはわからないが、この；（セミコロン）はプログラムにとっての句点のようなものなのだろうか。』

エラーエッセージは「変数 ”x” は存在しません」と表示されていて，学生はそのエラーメッセージを参考に何度も変数を確認していることがわかる．しかし，実際には変数とは関係のない場所に不必要な「；」を記述していることがエラーの原因である．ここで学生はエラーメッセージで指摘された場所を何度も確かめるが原因を見つけることができず，そのあとはすべての行を一行ずつ確認していくことになる．プログラムを学び始めた学習者にとってはこの作業はとても負担が大きい．ここであきらめてしまう学生も多く，「エラーメッセージがわかりにくい」といった記述が多くなる．

次に，「エラーメッセージがわかるとやりやすいですか」のアンケート結果を図6に示す．エラーメッセージがわかると，「とてもやりやすい」と「やりやすい」で全体の6割を超える．このことから，エラーメッセージで苦労している学生が多いことがわかる．また、「あまり変わらない」「変わらない」を選択した学生は，エラーメッセージがわかりにくいことが原因でプログラミング学習をあきらめてしまう傾向にある．入門者や初心者を対象

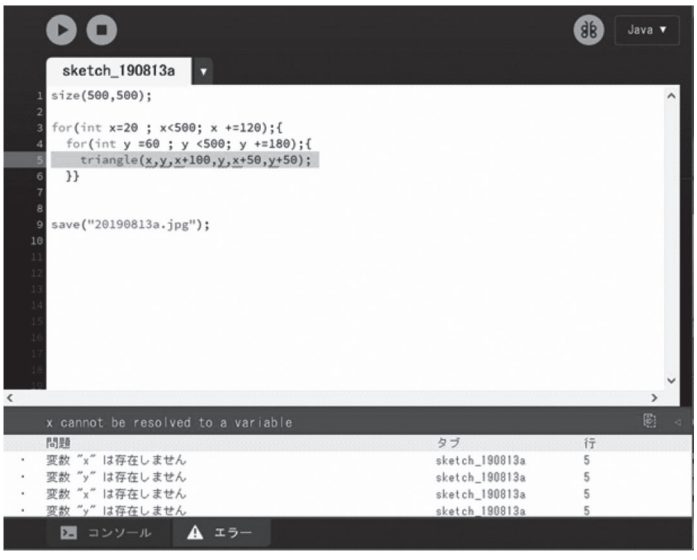

図5　エラーを含むプログラム例の実行結果

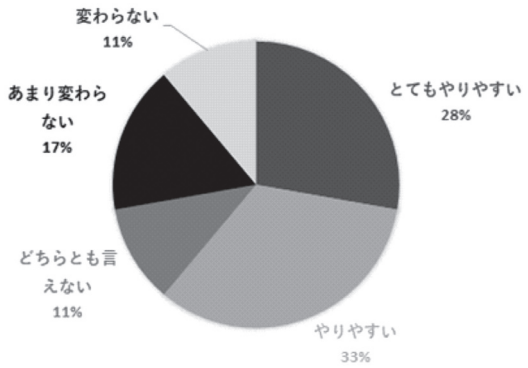

図6　エラーメッセージについて

としたプログラミング学習支援システムには，エラーメッセージに対する支援を重点的に行う必要がある．本研究で構築する Processing を用いたプログラミング学習支援システムの場合，Web ブラウザで実行できるようにするために Processing.js を用いている．しかし，Processing.js のコンパイラは厳密なチェックを行わず，エラーを含むプログラムを実行した場合に，学習者に適切なエラーメッセージを提示しないことが多い．

以上のことから，学習者にわかりやすいエラーメッセージを提示するプログラミング学習支援システムが必要であると考えられる．

4　実装

本章では，本稿で作成したエラーメッセージを正確に提示する Processing を用いたプログラミング学習支援システムについて述べる．次に，従来の Processing.js のエラーメッセージと本システムのエラーメッセージについて比較する．

4.1　システムの流れ

システムの初期画面を図7に，エラーメッセージに関する部分のシステムの動作を図8に示す．

プログラムを作成した学習者は，エラーの有無を確認するために「確認」ボタンを押してサーバにプログラムを送信する．プログラムを受信したサーバは，文法や構文を厳密にチェックするために Processing のコンパイラでプログラムの確認を行う．エラーがある場合はそのエラーメッセージを学習者に提示するとともに，サーバ内にエラーメッセージを保存していく．エラーメッセージが無かった場合は，「実行」ボタンを押して，Web ブラウザで実行するための Processing.js のコンパイラでプログラムを実行する．

4.2　エラーメッセージの比較

本節では，Web ブラウザで実行する場合に一般的に利用する Processing.js のコンパイラのエラーメッセージと Web ブラウザで実行する場合でも Processing のコンパイラを利用するように改善した本システムでのエラーメッセージを比較する．本稿では主にプログラムの入力ミスについて比較した．図9に比較検証するためのサンプルプログラムを示す．

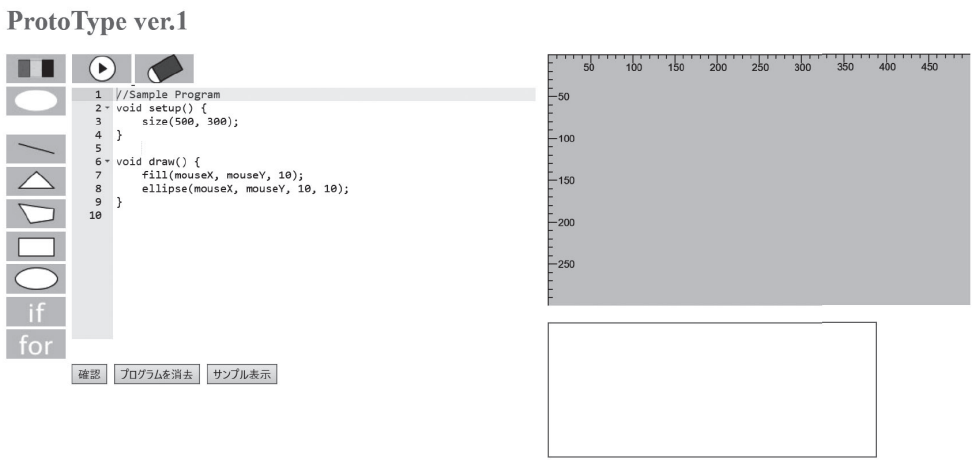

図7　初期画面

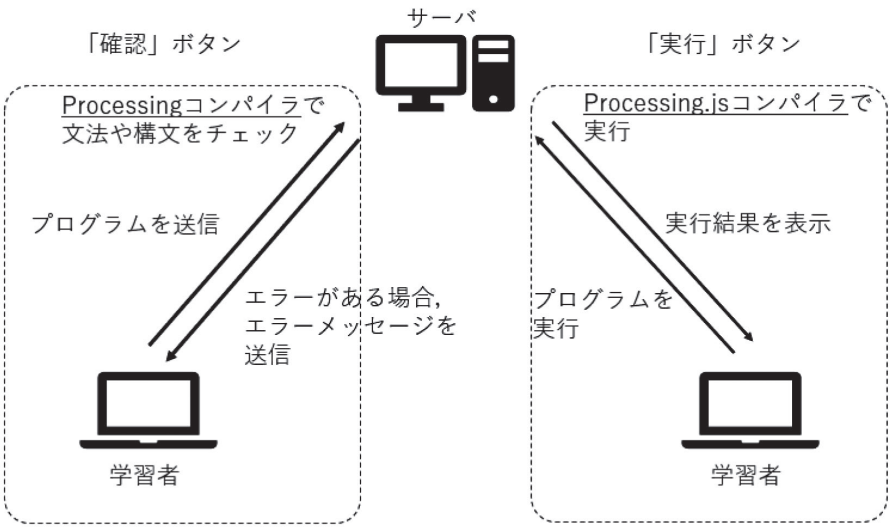

図8　処理の流れ

```
void setup() {
  size(500, 300);
}

void draw() {
  fill(mouseX, mouseY, 10);
  ellipse(mouseX, mouseY, 10, 10);
}
```

図9　サンプルプログラム

4.2.1　関数名を入力ミスした場合のエラーメッセージの比較

本項では，エラーメッセージをいくつかのパターンに分けて説明する．

size 関数を sze と入力した場合，Processing.js では図10のようなメッセージが表示される．

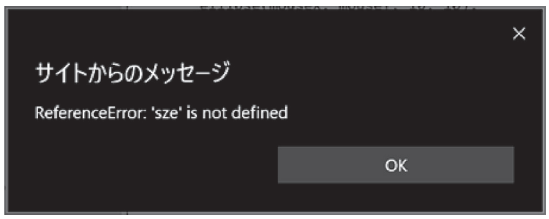

図10　Processing.jsのエラーメッセージ

本システムで実行した場合，厳密に文法や構文を確認する Processing のコンパイラを利用するため，図11に示すように，Processing.js のコンパイラが提示するエラーメッセージよりも詳しいエラーメッセージを提示していることがわかる．

次に，fill 関数を fll，ellipse 関数を ellise と間違えて入力した場合，Processing.js のコンパイラはエラーメッセージを表示しない．これは学習者にとってはプログラムをどのように修正したらいいのかわからないといった不安や学習が進まないことによる学習意欲の減少などの原因となる．同じ入力ミスをしたプログラムを本システムで実行した場合に提示されるエラーメッセージを図12に示す．

本システムで実行した場合は，それぞれの入力ミスに対して以下のようなエラーメッセージを提示する．

- fill を fll と入力した場合
  → The function fll（int, int, int）does not exist.
- ellipse を ellise と入力した場合
  → The function ellise（int, int, int, int）does not exist.

今回のエラーの場合，Processing.js のコンパイラが何も表示しないのに対し，本システムのエラーメッセージはかなり詳細なレベルで学習者に原因を提示していることがわかる．

```
[12/Nov/2019 17:39:12] "POST /form HTTP/1.1" 200 13265
mySketch.pde:5:0:5:0: The function sze(int, int) does not exist.
```

図11　本システムのエラーメッセージ（size）

```
[12/Nov/2019 17:43:28] "POST /form HTTP/1.1" 200 13265
mySketch.pde:13:0:13:0: The function fll(int, int, int) does not exist.
[12/Nov/2019 17:45:09] "POST /form HTTP/1.1" 200 13265
mySketch.pde:15:0:15:0: The function ellise(int, int, int, int) does not exist.
```

図12　本システムのエラーメッセージ（fillとellipse）

```
[12/Nov/2019 17:57:53] "POST /form HTTP/1.1" 200 13265
mySketch.pde:3:0:3:0: Cannot find a class or type named “vid”.
```

図13　本システムのエラーメッセージ（void）

キーワードである void を vid と記述した場合，エラーであるにも関わらず Processing.js のコンパイラはエラーメッセージを提示せず，記述したプログラムどおりに実行される．これでは学習者は正しい文法や構文を学ぶことができない．本システムで同じプログラムを実行した場合，図13に示すように，「Cannot find a class or type named”vid”」と提示され，プログラムが動作した場合でもエラーが含まれていることを学習者に提示していることがわかる．

setup 関数を stup，draw 関数を daw と記述した場合，Processing.js のコンパイラはどちらもエラーメッセージを表示しない．また draw 関数の場合はプログラムも実行されない．ただし，setup 関数の場合は，エラーメッセージは表示されないが，指定した画面サイズ（size（500, 300）;）は無視され，デフォルトの画面サイズ（100×100）のままプログラムは実行される．本システムで実行した場合も Processing.js のコンパイラの結果と同じであった．これは今後改善すべき課題である．

4.2.2　入力不足の場合や関数の引数が足りない場合のエラーメッセージの比較

次に，ブロックを表す「｛」や「｝」が足りない場合や関数の引数が足りない場合などのパターンについて説明する．

はじめに，ブロックを表す「｛」や「｝」が足りない場合について説明する．

draw 関数のブロックの終わりを表す「｝」が足りない場合，Processing.js のコンパイラは図14に示すようなエラーメッセージを提示する．しかし，このエラーメッセージの内容を入門者や初心者が把握することは困難である．本システムで実行した場合のエラーメッセージを図15に示す．本システムのエラーメッセージは具体的に「｛」や「｝」が表示され，直感的にもわかりやすいエラーメッセージが提示されていることがわかる．

次に引数が足りない場合のエラーメッセージについて述べる．

図14　Processing.js のエラーメッセージ（draw）

画面のサイズを指定する size 関数は，第一引数には画面の幅，第二引数には画面の高さを指定する．「size（500, 300）;」を「size（500）;」と記述した場合，Processing.js のコンパイラは画面サイ

```
[12/Nov/2019 18:05:36] "POST /form HTTP/1.1" 200 13265
mySketch.pde:19:0:19:0: Found an extra { character without a } to match it.
```

図15　本システムのエラーメッセージ（draw）

```
2019-11-24 20:46:57.052082 : b'mySketch.pde:15:0:15:0:
The method ellipse(float, float, float, float) in the type PApplet
 is not applicable for the arguments (int, int, int)
```

図16　本システムのエラーメッセージ（ellipse）

```
b'mySketch.pde:5:0:5:0: The function sze(int, int) does not exist.\r
b'mySketch.pde:13:0:13:0: The function fll(int, int, int) does not e
b'mySketch.pde:15:0:15:0: The function ellise(int, int, int, int) do
b'mySketch.pde:3:0:3:0: Cannot find a class or type named \x81gvid\x
b''
b''
b'mySketch.pde:19:0:19:0: Found an extra { character without a } to
b'Exception in thread "main" java.lang.ArrayIndexOutOfBoundsExceptic
b''
b'mySketch.pde:15:0:15:0: The method ellipse(float, float, float, fl
```

図17　保存されたエラーメッセージ（一部）

ズを（500, 100）に変更してプログラムを実行する．本システムで実行した場合も同様である．これは，コンパイラがx座標を500，y座標はデフォルトの100とするためである．

次に，第一引数で赤色を指定，第二引数で緑色を指定，第三引数で青色を指定することで色を塗るfill関数について説明する．さらに，今回のサンプルで使用しているfill関数は，カーソルの位置に合わせて色が変化する処理を行う．はじめにProcessing.jsのコンパイラで「fill（mouseX, mouseY, 10）;」を「fill（mouseY, 10）;」とした場合，カーソルの位置に合わせて色が変わることなく実行される．これは本システムで実行した場合も同様である．fill関数の引数が2つの場合，最初の引数はgray濃度，2番目の引数は透明度として処理されるためである．文法上のエラーがないのでどちらのコンパイラもエラーメッセージを提示しないが，図3の「動いても思っていたものと違う」という結果が示すように，学習者が思い描いている結果とは異なる結果になる原因になりやすい．

最後に，楕円を描画するellipse関数の場合について述べる．ellipse関数には4つの引数があり，順番に，描画する楕円のx座標，y座標，楕円の幅，楕円の高さを指定する．幅と高さに同じ値を指定することで円を描画することもできる．今回のサンプルで使用している「ellipse（mouseX, mouseY, 10, 10）;」を「ellipse（mouseY, 10, 10）;」と記述した場合，Processing.jsのコンパイラはエラーメッセージを提示せず，処理も実行しない．本システムで実行した場合は，図16に示すような詳細なエラーメッセージが提示される．

以上のように，従来のWeb IDEで使用されていたProcessing.jsでは提示しないエラーメッセージでも，Processingのコンパイラを利用するように改善した本システムでは，かなり正確にエラーメッセージを提示していることがわかる．このことから，学習者にとって学習しやすいプログラミング支援システムが構築できると考えられる．

4.3　エラーメッセージの保存

本節では，上述したエラーメッセージの保存について説明する．

プログラムを学習しようとした場合，多くの入門者や初心者がプログラムの作成中に発生するエラーに対処できず，プログラミング学習をやめてしまう傾向にある．図6に示したように，エラー

メッセージを改善し有効活用することで学習者にとって使いやすいプログラミング学習支援システムを構築することができる．

本研究では，エラーメッセージを保存し分析することで学習者にとって有益な情報を提示することを目標としている．そこで本稿では，学習者がプログラムを作成し，「確認」ボタンを押すことで Processing のコンパイラを使用して文法や構文のエラーを確認し，その結果をテキストファイルとして自動的に保存するシステムを構築した．本システムで保存されたエラーメッセージを図17に示す．

発生したエラーを学習者ごとに追加・蓄積し，分析していくことで，その学習者の理解度や進捗状況，エラーの傾向などを把握し，学習者に最適なアドバイスを提示するシステムが開発できると考えられる．

5　まとめと今後の課題

2020年以降，小学校から始まるプログラミング教育の必修化に向けて，多くの問題点が指摘されている．文部科学省が行ったプログラミング教育の実施に向けたアンケート調査の結果，適切な教材の不足が課題である（72.4%）、指導方法の情報不足が課題である（75.4%）という結果が出ている [12]．また，予算が限られていることから，無料で使える教材を必要としている．教員のスキルが不足している場合は，機器の設定等の作業負担などの問題も指摘されている．また，教員一人に対して，学習者の様子を十分に把握し直接指導できる人数も限られてくる．そうした状況の中で，教員に代わって学習者ごとの理解度や進捗状況の把握，適切なエラーメッセージとそのアドバイスを提示するシステムが必要である．

そこで本研究では，上記の問題点を解決するためのプログラミング学習支援システムを構築することを目標とする．

本稿では学習者ごとに適切なアドバイスを提示するための情報の1つであるエラーメッセージについて説明した．従来の Web IDE で使用されている Processing.js ではエラーを含むプログラムに対して，適切なエラーメッセージを提示しないことが多い．そのため，学習者は自分の作成したプログラムに対して必要な情報を得ることができない．この問題点を解決するために，本稿では本来スタンドアロンで使用する Processing のコンパイラを Web IDE でも使用できるようにし，Processing.js よりもかなり正確なエラーメッセージを学習者に提示するように改善した．次に，いくつかのエラーのパターンに対して，改善したシステムでその動作を確認した．また，エラーメッセージを学習者ごとに分析することで，学習者に対して適切なアドバイスを提示するために，エラーメッセージを保存するようにした．

今後の課題として，エラーメッセージを表示しなかった一部のエラーメッセージの表示，エラーメッセージの日本語化など入門者や初心者に対してわかりやすい表現に改善することが挙げられる．また学習者ごとにエラーメッセージを保存し分析することで，学習者ごとに適切なアドバイスを提示するシステムの構築が必要である．さらに，思った通りに動かない原因を推論し，学習者にその原因や改善方法などの適切なアドバイスを提示するシステムの構築を目指していく．

参考文献

[1] 文部科学省 . 教育の情報化の推進：http://www.mext.go.jp/a menu/shotou/zyouhou/index.htm.

[2] 一般社団法人日本産業技術教育学会（編）. 小・中・高等学校でのプログラミング教育実践 . 九州大学出版会 , 2019.

[3] Scratch. https://scratch.mit.edu/.

[4] 鷲崎弘宣 , 齋藤大輔 , 坂本一憲 . 新学習指導要領準拠 Scratch でたのしく学ぶプログラミング的思考 . マイナビ , 2019.

[5] 朝日翔太 . 初等教育におけるテキスト型プログラミング言語 python によるプログラミング教育の効果検証 . 日本教育工学会第34回全国大会講演集 , pp. 231–232, 2018.

[6] Processing. https://processing.org/.

[7] DanielShiffman（著）, 尼岡利崇（訳）. 初めての Processing 第2版 . O'REILLY, 2018.

[8] Processing.js. http://processingjs.org/.

[9]　p5.js. https://p5js.org/.
[10] 栗原あずさ , 佐々木晃 , 脇田建 , 細部博史 . Processing アプリケーション開発のための視覚的ドメイン特化言語の実装 . 日本ソフトウェア科学会第31回大会講演論文集（PPL6-2）, 2014.
[11] 三浦元喜 . Processing web ide を用いたプログラミング基礎教育の試み . 情報処理学会情報教育シンポジウム予稿集
（SSS2013）, pp. 225–231, 2013.
[12] 文部科学省 . 教育委員会等における小学校プログラミング教育に関する取組状況等：
http://www.mext.go.jp/a menu/shotou/zyouhou/detail/1406307.htm.

Programming learning support system for beginners with improved error messages for Web IDE

Tomoo SUMIDA*, Hideki KOJIMA*

* Graduate School of Education, Tohoku University

ABSTRACT

From 2020, many programming learning support systems have been proposed and operated with the compulsory programming education starting from elementary school. Currently, many of the proposed programming learning support systems for beginners use a visual programming language such as Scratch, which can be visually intuitively and intuitively programmed. However, it has been pointed out that a programming learning support system using a visual programming language does not consider a smooth transition to the programming language actually used.

The purpose of this research is to construct a programming learning support system using Processing that is easy for beginners to understand. This programming language is easy for beginners to understand and can acquire fullfledged programming skills. In order to work Processing language on the Web, it is necessary to use Processing.js converted to JavaScript, but the Processing.js compiler does not strictly check the grammar or syntax. Therefore, it does not present an appropriate error message to the learner. This paper describes a Web IDE using Processing that was built to improve the disadvantages of presenting error messages in Processing.js.

Key words: programming education, programming learning support system, Web IDE, error message, Processing