

プログラミング教育の補助システムの提案

広瀬研究室3年

C1172081 山口円馨

令和2年1月15日

概要

2020年度から小学校をはじめとしたプログラミング教育が始まる。しかし、初めての取り組みへの教員の不安や、必修としてプログラミング学習を行っている東北公益文科大学(以下本学)の学生を対象としたアンケート結果から、プログラミング初学者が考える問題点、また教員側が望むシステムを考察した。これにより、利便性を含んだプログラミング教育への補助システムを提案する。

目次

| | | |
|------------|-----------------|-----------|
| 第1章 | はじめに | 5 |
| 1.1 | 背景 | 5 |
| 1.2 | 基礎プログラミング | 5 |
| 1.3 | 目的 | 5 |
| 第2章 | 事前調査 | 7 |
| 2.1 | プログラミング学習サイトの概要 | 8 |
| 2.1.1 | paiza ラーニング [8] | 9 |
| 2.1.2 | Progate [9] | 9 |
| 2.2 | プログラミング経験の有無 | 10 |
| 2.3 | 既存の研究 | 11 |
| 2.3.1 | Wapen の改良 | 11 |
| 2.3.2 | 試験システム track | 11 |
| 第3章 | 提案 | 13 |
| 3.1 | 問題 | 13 |
| 3.1.1 | 学生側 | 13 |
| 3.1.2 | 教員側 | 13 |
| 3.1.3 | プログラミング学習サイト | 13 |
| 3.2 | 対策 | 13 |
| 3.2.1 | 学生側 | 14 |
| 3.2.2 | 教員側 | 14 |
| 3.2.3 | プログラミング学習サイト | 14 |
| 3.3 | システムの設計 | 14 |
| 3.4 | システムの説明 | 16 |
| 3.4.1 | 教員ページ | 16 |
| 3.4.2 | エディタページ | 16 |
| 3.4.3 | 実行ページ | 16 |
| 3.5 | システムの機能 | 16 |
| 3.5.1 | プログラムの実行・評価 | 16 |
| 3.5.2 | メソッド・想定の説明 | 16 |
| 3.5.3 | エラー文 | 17 |
| 3.5.4 | データベースへの格納 | 17 |

| | | |
|------------|-----------------------------|-----------|
| 第4章 | 補助システムの開発 | 19 |
| 4.1 | 開発環境 | 19 |
| 4.2 | システムの作成 | 19 |
| 4.2.1 | プログラムの実行・評価 | 19 |
| 4.2.2 | 問題文・出力結果例・メソッドの説明 | 20 |
| 4.2.3 | エラー文 | 21 |
| 第5章 | 考察 | 23 |
| 5.1 | 脆弱性の問題 | 23 |
| 5.2 | 脆弱性の問題への対策 | 23 |
| 第6章 | 結論と今後の展望 | 25 |
| 6.1 | 結論 | 25 |
| 6.2 | 課題 | 25 |
| 6.3 | 今後の展望 | 25 |

第1章 はじめに

本章では研究の背景と目的について説明する。

1.1 背景

新学習指導要領 [1, 2, 3, 4, 5, 6] が公示され、令和2年度から小中高生を対象としてプログラミング的思考の育成やプログラミング学習が必修となる。なお、中学校から実際のプログラミングが始まるが、これに対して、教職員の実態と意識調査の結果では、98%が「授業の実施に不安」を感じているといった結果が出た [7]。また、第2章で述べるアンケート結果により、プログラミングに対しては「難しい」「将来役に立つのかわからない」「動く楽しい」といった意見が見られた。この結果を受けて、プログラミング教育への補助システムの重要性があると考え、検討した。

1.2 基礎プログラミング

本学ではプログラミングを学ぶ基礎プログラミングI・IIが開講されている。これらの講義は必修科目であり、在学する学生が必ず受講する。初めての取り組みに苦戦し、プログラミング自体に苦手意識を持つ学生が見えた。更に、プログラムのエラーで生じるエラー文を読み解くことが出来ないことも苦手意識をもたせる原因となっている。

講義の課題はMewを使用したメールで送信される。そのため、教員側はメールを確認し、記載されたプログラム文を目視した上で実行し、プログラムが成功しているかを判断する必要がある。また、講義内で説明として表示されているプログラム文を実行する時間には個人差があり、考慮すると説明すべき点を十分に説明できない事態となる。

1.3 目的

現状の課題を解決するために、基礎プログラミングIIで学習するCGI¹を用いてプログラミングの実行、評価を行うシステムの作成を提案する。これを用いることで、生徒側と教員側の課題を緩和し、よりプログラミング教育の難易度を下げることが目的とする。CGIを利用したシステムは授業で実際に全生徒が学ぶため、改良の余地が十分に広がる。対象はプログラミング初学者であり、使う環境は外部からアクセスが出来ないローカル環境を想定している。

¹Common Gateway Interface の略で、Web サーバー上でプログラムを使って処理した結果を表示させる機能のことである [16]。

第2章 事前調査

本章では、初学者である基礎プログラミング受講者が授業を受ける際に、難しいと感じている現状や問題点をアンケートを用いて集計した。アンケートの体裁を以下に示す。

基礎プログラミング II アンケート

所属コース()

質問 1. プログラミングについてどう感じますか。

質問 2. 大学入学前にプログラミングを体験した経験がありますか？

(Progate など学習サイトを含む)

ある

使用した学習サイトなどあれば記入をお願いします。

()

ない

質問 3. 基礎プログラミングの難易度はどのくらいですか。

簡単 少し簡単 普通 難しい とても難しい

1

2

3

4

5

質問 4. プログラミングで特に難しいと感じた部分はどこでしたか。

動かない・エラー文が読めない

メソッドが覚えられない

そもそも組み方がわからない

その他()

質問 5. プログラミングを楽しいと思う・理解したきっかけは何でしたか。

動いたとき

難易度が高いプログラムを作れたとき

緻密な作業(デバッグなど)

その他()

質問3、質問4に対する結果は図2.1,2.2のとおりである(2019-11-19時点)。

アンケートにより、プログラミングを難しいと考えている、また特に難しいと感じた部分では選択肢を3つに絞った内、ほぼ均等に苦手意識が現れていることが分かった。

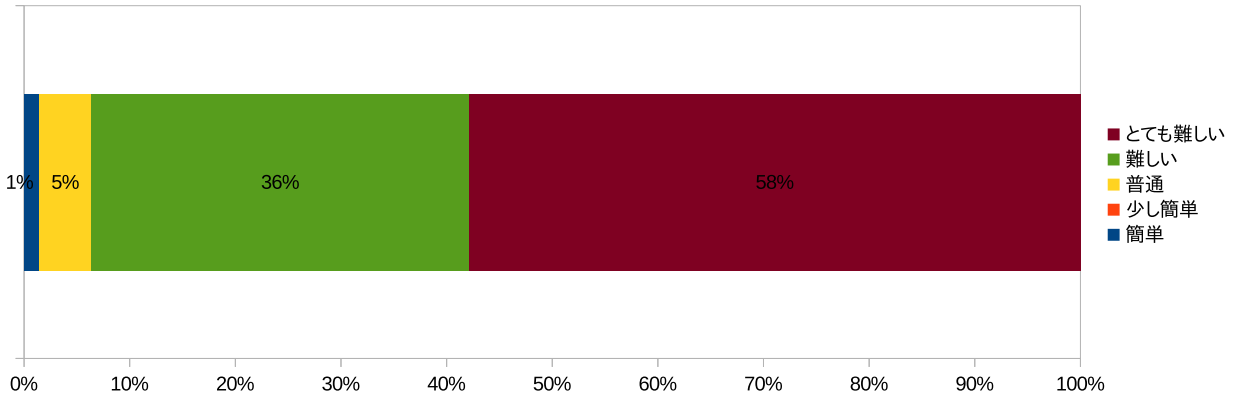


図 2.1: 基礎プログラミングの難易度

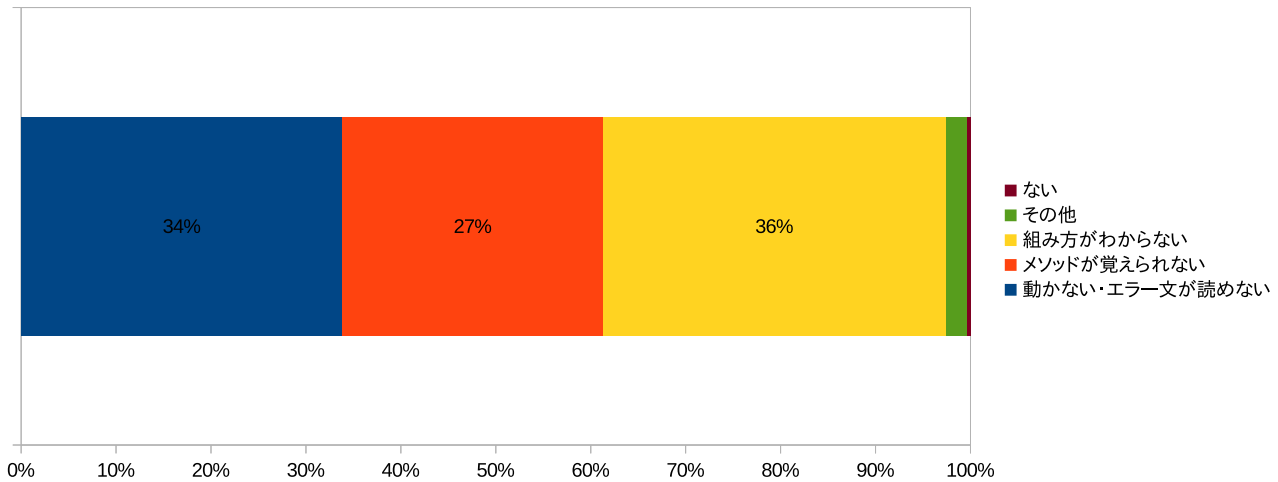


図 2.2: プログラミングで特に難しいと感じた部分

2.1 プログラミング学習サイトの概要

インターネット上にはプログラミング学習を行えるサイトがいくつか存在している。以下に例として2つの学習サイトの詳細を示す。どちらにも共通するデメリットとして

- 用意されている問題のみで、自分が疑問に思った問題に取り組むことが出来ない。
- 一部のレッスンは有料となる。

といった点が挙げられる。

2.1.1 paiza ラーニング [8]

ギノ株式会社が開発した転職サイトである。実行画面は図 2.3 のとおりである。目的別に名称が分かれており、プログラミングを行う学習サイトが paiza ラーニングである。動画を見ながら学習する。各言語の最初の動画では言語の説明や使うことで何が出来るか、採用されている例はどういったものかの説明が行われる。paiza.IO というオンライン実行環境によって実行環境が Web 上に構築されている。有料会員になることで実際のエンジニアに質問を行えるようになるなど、疑問に対する取り組みも存在する。間違いやすいポイントなども Tips として扱われている。一つのチャプターの説明が終わると演習問題に取り組むことが出来る。演習問題では起こりうるエラーなどに対しての対策を問題として取り上げており、ここでは実際にコードを記入し、判定が行われる。



図 2.3: paiza の実行画面

2.1.2 Progate [9]

株式会社 Progate が開発した、アプリケーション化もされており、スマートフォンなどの端末上からでもプログラミング学習が可能なプログラミング学習サイトである。実行画面は図 2.4 のとおりである。はじめにスライド式で言語やメソッドの説明を行い、実際にコードを入力する演習に取り組む。問題に正しく答えられた場合は次の演習へと進む。基礎的なレッスンは無料で受講できるが、高度な技術を要するレッスンは有料となる。

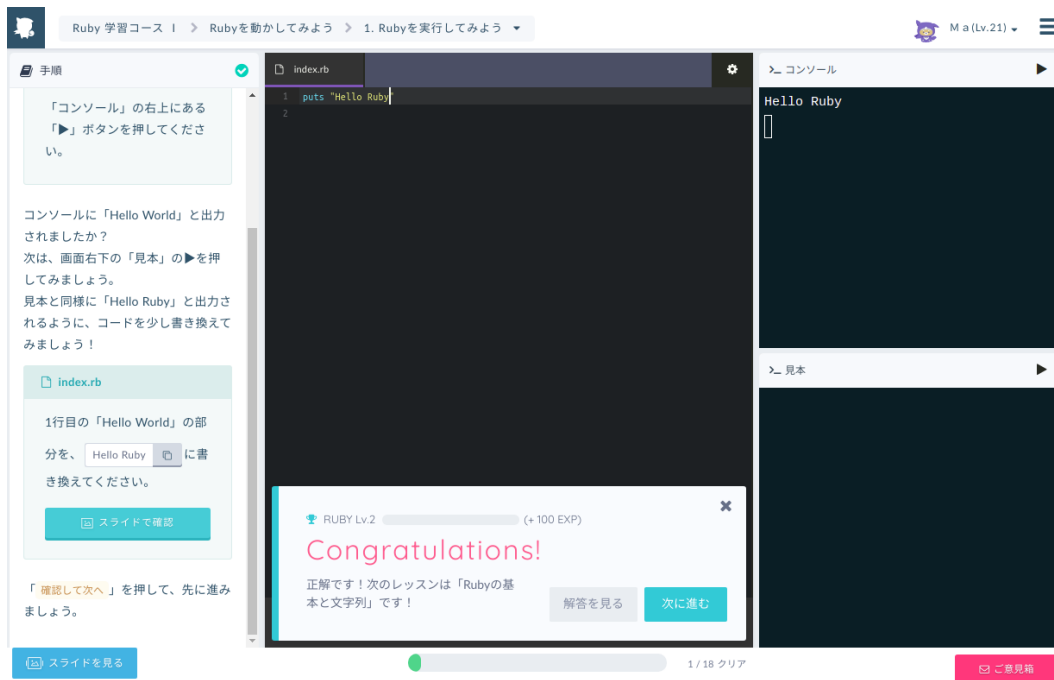


図 2.4: Progate の実行画面 PC 版

2.2 プログラミング経験の有無

上記のようなプログラミング学習サイトを大学入学前に経験している学生の割合は 4.43% となった。アンケートの結果は図 2.6 である。

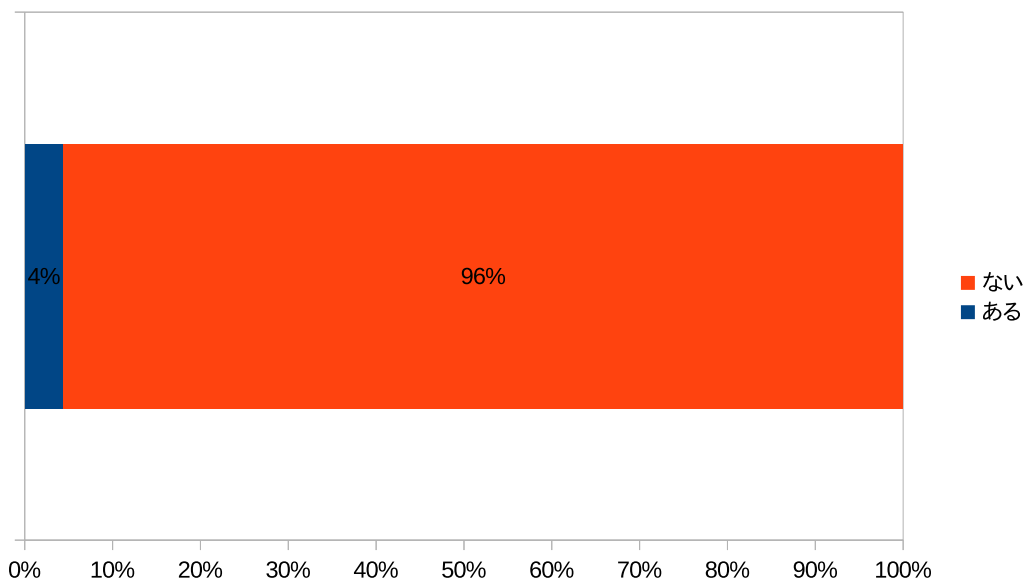


図 2.5: プログラミング経験の有無

2.3 既存の研究

本研究に類似する既存の研究を調べた。

2.3.1 Wapen の改良

中西による Web ブラウザ上のプログラミング学習環境 WaPEN の改良 [10] では、プログラミング学習環境を Web ブラウザ上で実行できる環境として WaPEN [11] を開発し、勤務校での授業で用いている。教員等がサンプルプログラムを簡単に差し替えられるようなプログラムも開発されており、授業を円滑に進めるためのシステムが多数存在する。この開発環境では、プログラムをコードとしてではなく日本語としての説明として入力、フローチャート化するため、流れを理解するためのシステムである。考察にあるとおり、これは初学者にとって有効であるがある程度上達した者にとっては足かせとなっている面もある。

The image shows the WaPEN web interface. On the left is a code editor with a list of lines (1-21) and a 'プログラム' (Program) label. The code includes variable declarations, random number generation, and conditional logic for comparing 'a' and 'b'. Below the code editor are input fields for integers, real numbers, strings, and booleans, along with control buttons like '実行' (Execute) and '入力支援ボタン' (Input Support Button). On the right is a flowchart labeled 'フローチャート' (Flowchart) that visualizes the program's logic: starting with 'はじめ' (Start), it sets 'a=0', generates a random number 'b', prompts for input, and then enters a loop to compare 'a' and 'b' until they are equal, finally displaying 'あたり' (Hit) and ending at 'おわり' (End).

図 2.6: WaPEN の使い方

2.3.2 試験システム track

新田, 小西, 竹内らの複数言語に対応しやすいオンラインプログラミング学習・試験システム track [12] では、プログラミング学習環境の構築や改善への期待が高まっていることを鑑みて、学校や企業におけるプログラミング教育に用いることのできるシステムとして track というオンラインプログラミング学習・試験配信プラットフォーム [13] を提案している。track の導入には料金が発生し、一般企業などで「採用選考で活用したい」「社員育成/評価で活用したい」といった用途に合わせた料金プランが用意されている。

導入している会社には株式会社バンダイナムコスタジオ、LINE 株式会社がある。

第3章 提案

本章では第2章までで問題となる部分を踏まえ、対策を講じた上でシステムを提案する。

3.1 問題

それぞれで考えられる問題を以下に示す。

3.1.1 学生側

学生側の問題としては以下が挙げられる。

- プログラミング経験がないため苦手意識が起こる。
- エラー文の解読が出来ず苦手意識に繋がる。

3.1.2 教員側

教員側の問題としては以下が挙げられる。

- プログラムの評価に手間がかかる。
- 講義時間内に説明できる事柄が学生側の技量で変化する。

3.1.3 プログラミング学習サイト

プログラミング学習サイトの問題としては以下が挙げられる。

- 決まった問題にしか取り組むことが出来ない。
- 企業が制作しているため、本格的な学習を行うには料金が発生する。

3.2 対策

3.1 で示した問題に対しての対策案を以下に示す。

3.2.1 学生側

学生側には表示によって視覚的に情報を伝え、また苦手意識の原因となる部分を改善していく。学生側の問題への対策を以下に示す。

- プログラミングはどういったものか、説明を視覚的に表示する。
- エラー文をより分かりやすく、調べやすく表示する。

3.2.2 教員側

教員側のヒューマンリソースを鑑みて、一連のプロセスをシステム化することで手間の軽減を図る。教員側の問題への対策を以下に示す。

- プログラム文の判定を自動で行う機能の実装。
- 簡易なプログラム文を実行できるエディタの用意。

3.2.3 プログラミング学習サイト

プログラミング学習は教員によって指導方法が異なる。どのような方法にも柔軟に応じるために、問題文は固定したものではなく自由に変更できる必要がある。プログラミング学習サイトで起こる問題に対する対策を以下に示す。

- 問題を教員側が自由に設定できるシステムの実装。

3.3 システムの設計

3.2で述べた対策案をもとに、システムを設計する。一連の流れを図3.1に示す。

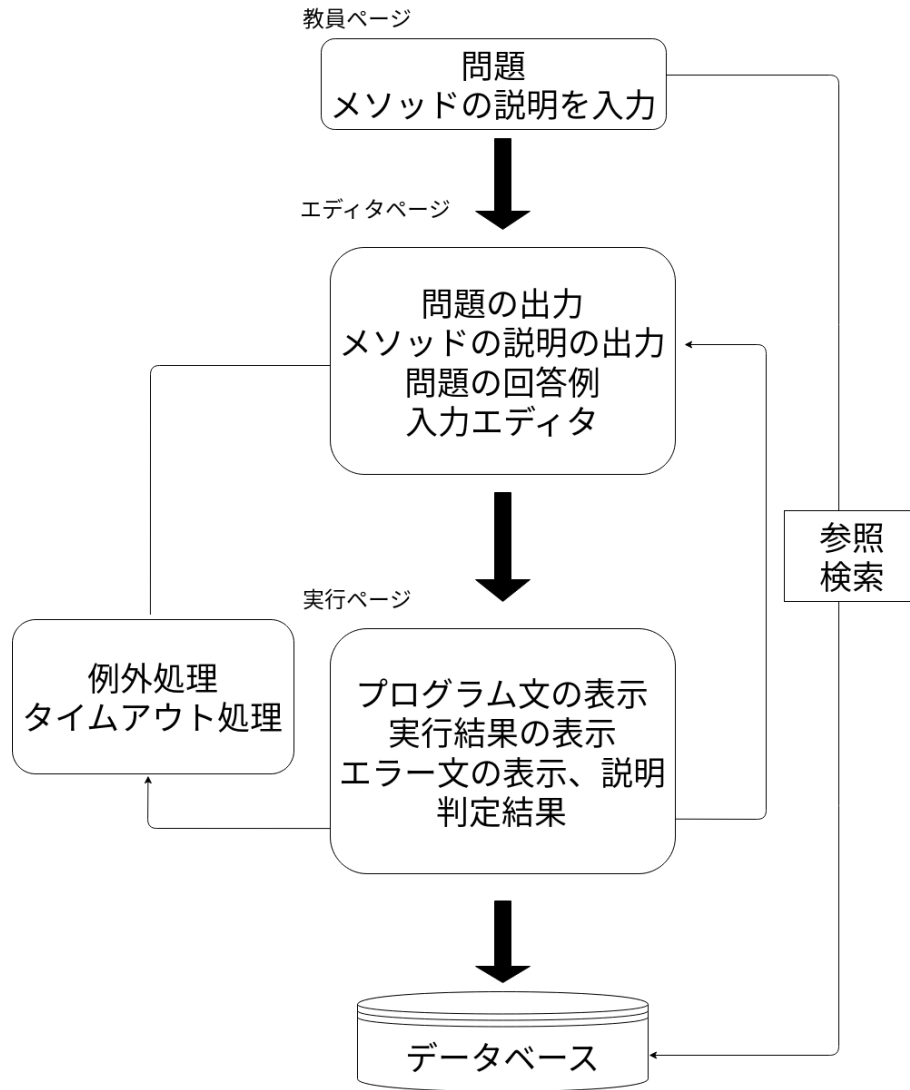


図 3.1: システムの流れ

3.4 システムの説明

実装するシステムについて説明する。

3.4.1 教員ページ

問題、メソッドの説明、出力結果例は課題ごとに変わるため、変数として扱う。これらを入力するページを作成し、生徒側には変更の権限がないように取り扱う。

3.4.2 エディタページ

教員ページで入力された問題、メソッドの説明、出力結果例が表示される。学籍番号またはユーザ名を入力欄、テキストエディタを用意し、生徒側がプログラム文を入力する。

3.4.3 実行ページ

エディタページで入力されたプログラム文を Ruby で実行し、判定を行う。実行結果を表示し、エラー文が出た場合は対応する文字列を表示する。成功していた場合はデータベースに格納される。特定の文字列が入力された場合は例外処理、繰り返し処理に失敗した場合はタイムアウト処理を行う。

3.5 システムの機能

本システムでは以下の機能をつける。

3.5.1 プログラムの実行・評価

打ち込まれたプログラムを実行・評価する。Ruby で実行され、出力結果、エラー文を実行ページに表示する。結果に応じた文字列や説明が表示される。

3.5.2 メソッド・想定の説明

問題を解く前にどういった場面で使うかの想定の説明をエディタページに表示する。これにより、どんな場面で使うかわからない、といった疑問に対して答えることを想定している。この部分は教員が入力できる仕組みになる。

3.5.3 エラー文

出力されたエラー文の説明を簡易的に伝える。アンケート結果にもある通り、エラー文が読み解けずに挫折する学生が多いため、出力頻度の高いエラー文には対策を講じた文章を表示する。その他、環境によってエラー文が英語で表示される場合がある。慣れていない文に対して苦手意識が多く見られるため、対策の説明は日本語で行う。

3.5.4 データベースへの格納

教員の採点のため、実行し成功したプログラム文はデータベースに格納する。検索を行えるように学籍番号またはユーザ名での登録を行う。格納の形態は以下の図 3.2 である。

| 学籍番号 | プログラム文 |
|---------|---------------------------------|
| C117208 | For I in 1..10 Puts I End |
| 空 | 空 |

図 3.2: データベース

第4章 補助システムの開発

本章では、以上の機能を含めたシステムの開発を行う。

4.1 開発環境

開発環境は以下の通りである。

- Ruby

Ruby とは、1995 年にまつもとゆきひろによって開発されたオープンソースの動的なプログラミング言語である。本システムは ruby 2.5.1p57 を使用した。 [14]

- SQLite3

SQLite とは、軽量で速く、高い信頼性を持つ SQL データベースエンジンを実装する C 言語ライブラリである。本研究でのバージョンは、SQLite3.22.0 である [17]。

4.2 システムの作成

実装した部分を以下に示す。

4.2.1 プログラムの実行・評価

プログラム文を入力するエディタを作成した。プログラム文を入力して送信すると送信先でプログラムを実行し、評価する。実行した際の画面を以下に示す。

学籍番号

ここが問題文

プログラム入力画面

プログラムのコードを入力

出力結果例

メソッド・想定の説明

ここがメソッドの説明の変数

Let Go reset

図 4.1: エディタ画面

4.2.2 問題文・出力結果例・メソッドの説明

入力を教員が行えるように変数として扱った。授業に応じて内容を変えることが出来る。コードソースは以下のとおりである。

Listing 4.1: editor.rb

```

1 #問題文
2 question = "ここが問題文"
3
4 #問題の出力結果例
5 code =
6   "i_=1
7   while i_<=10
8     puts i
9     i_+=1
10  end
11  "
12

```

```
13 #メソッドと想定の説明の変数
14 method = "ここがメソッドの説明の変数"
```

4.2.3 エラー文

出現する頻度の高いエラー文に対して、あらかじめテキストファイルに説明・対策を記載し、出現した際には説明と対策の文章を表示した。また、エラー文をそのまま検索エンジンで検索できるように href を付与した。エラーメッセージをクリックした画面が以下である。

結果画面

入力したコード

```
i = "get"
print r
```

実行結果

NO!!!実行結果が違います

エラー文

```
-e:2:in `': undefined local variable or method `r' for main:Object (NameError)
```

[\[main:Object\]](#)→オブジェクトを指定してないです。

[\[NameError\]](#)→変数が間違えているかも。

図 4.2: エラー文の表示



図 4.3: 検索結果の画面

第5章 考察

本章では、システムを実装した上で起こりうる問題を考察する。

5.1 脆弱性の問題

- SQL インジェクション
不正な SQL を投入することで、通常はアクセスできないデータにアクセスしたり更新する攻撃である。
- クロスサイトスクリプティング (XSS)
悪意あるスクリプトを標的となるサイトに埋め込む攻撃である。悪意のある人が用意したサイトにアクセスした人のブラウザを経由して XSS の脆弱性のあるサイトに対してスクリプトが埋め込まれる。
- DoS 攻撃
サーバなどのネットワーク機器に大量のパケットを送るなどしてサービスの提供を不能にする攻撃である。
- シェル実行 Ruby には、`system`、`exec`、`open`、といったシェル実行のカーネルがある。これによって、コンピュータ内の情報を抜き取ることが可能である。

エディタには任意の文字を入力できるため、脆弱性の問題が発生する可能性がある。以下に問題の説明を示す。

5.2 脆弱性の問題への対策

SQL インジェクションやクロスサイトスクリプティングの対策として、CGIの標準ライブラリである `CGI.html` を使用しエスケープ処理を行った。シェル実行の対策として、特定の文字列が入力された際に例外処理を起こすシステムを実装した。しかし、Ruby の実行を回避してシェル実行を行うことも可能なため、完全な対策とは言えない。

第6章 結論と今後の展望

本章では、結論と今後の展望を述べる。あとでいっぱいかく。

6.1 結論

プログラムを入力するテキストエディタや、プログラムが実行された結果に対しての表示を行うことが出来た。

6.2 課題

プログラミングを学ぶために、必要だと思われる対象について具体的に探す必要がある。教育のために伝えやすい視覚的效果を探す。

脆弱性による問題がある。テストを行って問題を見つけ、対策を講じる。また、プログラムだけではなく運用する環境に応じて対策の方針を変える。

6.3 今後の展望

実際に授業などで使うことで生徒側の問題点を深掘りする。ローカル環境以外で使う際に、脆弱性を補うシステムを考えていく。実用性の高いシステムにする。

参考文献

- [1] 文部科学省.”「高等学校学習指導要領」(平成 30 年告示)(2019)”.
- [2] 文部科学省. ”教育の情報化の推進”. http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/__icsFiles/afieldfile/2019/10/03/1421730_001.pdf.2019-11-18.
- [3] 文部科学省.”小学校プログラミング必修化に向けて” http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/004/siryo/__icsFiles/afieldfile/2018/10/05/1409851_6.pdf.2019-11-15.
- [4] 文部科学省.”小学校プログラミング教育の手引の改定(第二版)”. http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2018/11/06/1403162_02_1.pdf,2019-11-18.
- [5] 文部科学省.”小学校プログラミング教育の手引の改定(第二版)について”. http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2018/11/06/1403162_01_1.pdf.2019-11-18.
- [6] 文部科学省,”プログラミング教育に関連する研究教材”. http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2019/05/21/1417094_004.pdf.2019-11-18.
- [7] ReseMom,”小学校のプログラミング教育、先生の98%が「授業の実施に不安」”. <https://resemom.jp/article/2019/04/26/50334.html>.2019-11-18.
- [8] paiza ラーニング.”環境構築不要!初心者でも楽しく学習できるプログラミング入門サービス【paiza ラーニング】”.<https://paiza.jp/works>.2019-12-4.
- [9] Progate.”Progate | プログラミングの入門なら基礎から学べる Progate[プロゲート]”.<https://prog-8.com/>.2019-12-4.
- [10] 中西渉.”Web ブラウザ上のプログラミング学習環境 WaPEN の改良”. 情報教育シンポジウム論文集.2019,130-135,2019-11-12.
- [11] WaPEN.”自作プログラム”.<https://watayan.net/prog/>.2019-12-4.
- [12] 新田章太, 小西俊司, 竹内郁雄.”複数言語に対応しやすいオンラインプログラミング学習・試験システム track”. 情報教育シンポジウム論文集.2019,114-121,2019-11-12.

- [13] track.”エンジニアの採用と育成を支援するプログラミング「学習・試験」プラットフォーム”.<https://tracks.run/>.2019-12-4.
- [14] Ruby.”オブジェクト指向スクリプト言語 Ruby”.<https://www.ruby-lang.org/ja/>.2019-12-3.
- [15] HTML クイックリファレンス,”HTMLの基本”.<http://www.htmq.com/htmlkihon/001.shtml>.2019-12-3.
- [16] Perl UNIX/Linux windows,”CGI の基礎知識”,<http://www.tryhp.net/first.htm>.2019-12-3.
- [17] OSS × Cloud,”オープンソースのデータベース/SQLite とは”,https://www.ossnews.jp/oss_info/SQLite.2019-12-3.