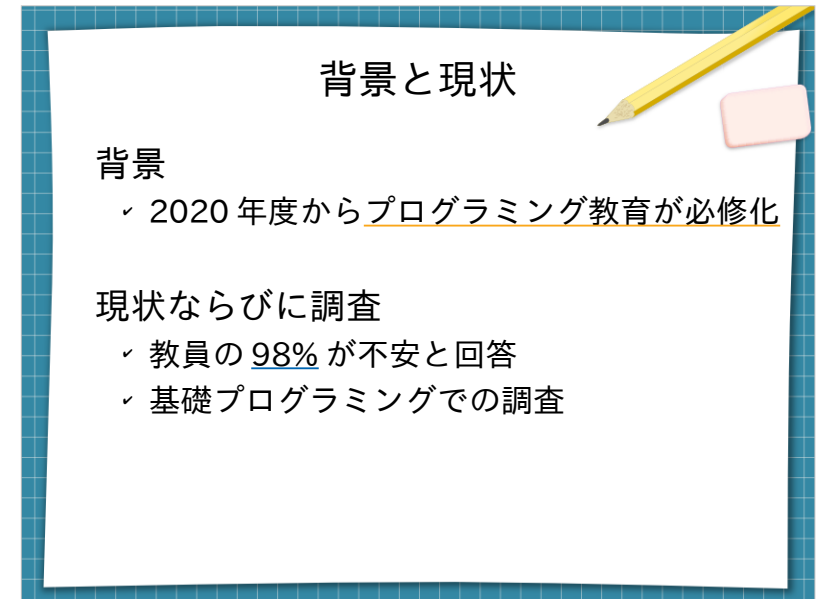


プログラミング教育の補助システムの提案
を発売します。



まずは背景と現状についてです。
背景としては、今年からプログラミング教
育が小学校からの必修科目になります。
また、背景に対しての現状ですが、
教員の98%が不安と感じていることや、後
に説明する基礎プログラミングでの調査か
ら不安が伺えました。
次のスライドから詳しく説明します。

アンケート概要

目的：プログラミングへの意識調査

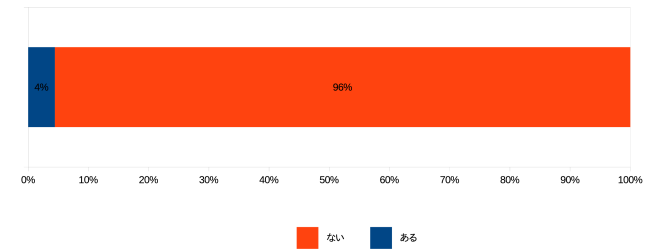
対象：2019年度基礎プログラミングIIを受講する学生

人数：203名

設問：自由記載形式1問、選択肢形式4問

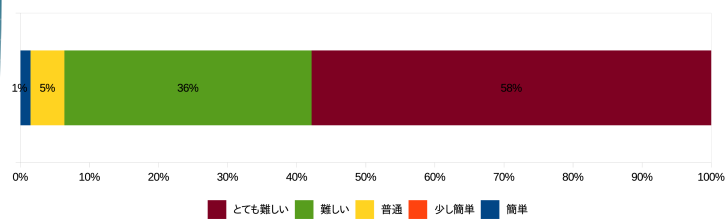
基礎プログラミングでのアンケートは、意識調査のために行いました。対象は2019年度の基礎プログラミングIIを受講する学生で、人数は203名です。アンケート内容は、自由記載形式が1問と、選択肢形式が4問です。

アンケート結果1 入学までにプログラミング経験があるか



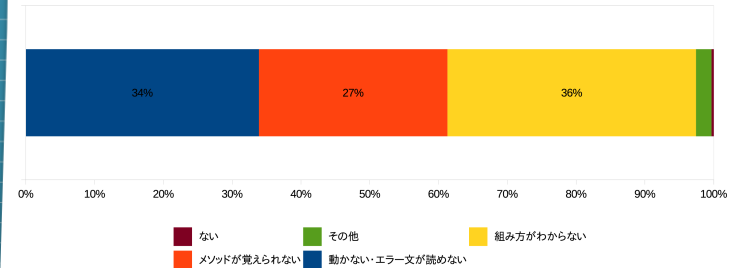
選択肢で選んでもらう質問の1つ目が入学までにプログラミング経験があるかです。ここでは、～%が無いと答えました。

アンケート結果2 プログラミングの難易度は どれぐらいか



次が、プログラミングの難易度はどれぐらいか。です。ここでは、難しいと考えている結果が約～%でした。

アンケート結果3 プログラミングで難しいと感じる点は どこか



次に、プログラミングで難しいと感じる点はどこか、です。ここでは3つの具体的な選択肢を設けましたが、同じ割合になりました。

プログラミングについてどう感じますか (自由記載欄)

難しい

使うかわからない

そもそも分かっていない(一部抜粋)

一方で...

達成感がある

動いたら面白い・楽しい(一部抜粋)

自由記載欄では、プログラミングについてどう感じますか、という質問をしました。

ここでは、難しい、使うかわからない、そもそもわかっていない、といった意見が多く見られましたが、その一方で、達成感がある。動いたら面白いなどのポジティブな意見もありました。

教員の声

「個人の採点は容易だが、母数が増えると途端に作業量が多くなる」

「実際に添付されたコードを見ただけで判別は出来るが、出来ないものは保存して実行するため手間はかかる」

また、講義を担当する教員に聞き取り調査をすると、

個人の母数が増えると作業が多い、や難しいプログラムになるほど、実際に動かすこともあるため、評価に手間がかかる

などの意見がありました。

問題点の発見

▶ 学生側

- ✓ 苦手意識がある
- ✓ エラー文の解釈が出来ない
- ✓ 全体的に意識向上が見られない

▶ 教員側

- ✓ プログラムの評価の手間が大きい

提案

▶ 学生側の問題を解決する提案

- ✓ エラー分の出力を見やすく表示
- ✓ プログラムを使う想定を表示

▶ 教員側の問題を解決する提案

- ✓ 情報を分かりやすく管理・表示

その他にも補助のシステムを作成 ...

以上の結果を受けて、学生側と教員側に見られる問題を発見しました。

学生側は、苦手意識がある、エラー文の解釈ができない。全体的に向上意識が見られないなどの問題がありました。

教員側は、プログラムの評価の手間が大きいという問題がありました。

そこで、このシステムではどちらの問題も改善して、よりプログラミングを学びやすい動きや表示を行うことを目的としました。

システムでは、学生側の問題を解決するために、

エラー分の出力を見やすく表示することや、プログラムを使う想定を表示。

教員側の問題を解決するために、情報を分かりやすく管理・表示することを提案し、作成しました。

その他にも学びやすい環境を独自に考えて追加していきました。

システムの完成図1 エディタページ

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回数

Let Go reset

システムの完成図2 プログラム実行ページ

結果画面

入力したコード

```
for i in 1..10
  puts i
end
```

実行結果

おめでとう！成功です！！

☆dbにかきこみました。お疲れ様です☆

エラー文はないよ！

[エディタ画面へ戻る](#)

実際に作ったシステムについて説明します。
最初に使う側が見るページとなるエディタページでは、ユーザ名の入力やプログラムの入力を行います。
またこのページに、問題文と、正しいプログラムを実行した時の出力結果や、こういった場面でこの動きをするプログラムが使われるかの想定が表示されます。
これにより、こういった場面で使うかわからない、といった意見の答えを出し、またこのページ内で問題を見る、プログラムを打ち込むと行った動作を一気に行うことができます。

次に、エディタページに打ち込んだプログラムを実行し、判定するプログラム実行ページの説明です。
ここでは、入力されたプログラムの出力結果が表示されます。成功していた場合には成功の文字が表示され、プログラム文とユーザ名がデータベースに登録されます。
間違えていた場合はエラー文を表示します。

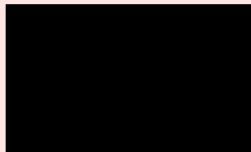
プログラム実行ページで エラー文が表示された場合

結果画面

入力したコード

```
for i in 1..10  
  puts i  
end
```

実行結果



実行結果が間違っているよ!!

エラー文

```
-e:2:in `block in
```

```
': undefined local variable or method `r' for mainObject (NameError) from -e:1:in `each' from -e:1:in `'
```

[mainObject]→オブジェクトを指定していません。

[NameError]→変数が間違えているかも。

[エディタ画面へ戻る](#)

エラー文が表示された場合の画面です。
エラー文の説明は日本語で行い、またその
文字をクリックすることで Google の検索画
面を開くことができます。
これにより、エラー文が読めないの問題
と、また調べる手間を省いています。

システムの完成図 3 個人評価確認ページ

ユーザ名とコード

ユーザ名	プログラム
momo	for i in 1..10 puts i end
madoka	cnt = 1 while cnt <= 10 puts cnt cnt += 1 end
madoka	for i in 1..10 puts i end
momo	for i in 1..10 puts i end
madoka	for i in 1..10 puts i end

ユーザ名検索欄

ユーザ名を入力

最後に、成功したプログラム文を確認でき
る個人評価確認ページです。

このページは教員が閲覧し、こういったプ
ログラム文で正解したかを確認できます。
ここに表示されている時点でプログラムは
正解の動きをしているので、本当に動くか
動かないか、の確認は行う必要がありません。
また、人数が増えると一気に見るプロ
グラムが多くなるため、名前でこういった
プログラムを打ち込んだかを検索する検索
窓を設置しました。

基礎プログラミングでの運用

実施回数：4回

時間：10分

人数：約40人

設問 1:1 から 1000 の数を表示しなさい

設問 2:1 から 1000 の数のうち、偶数のみ表示しなさい

設問 3:1 から 1000 の数のうち、3 の倍数のみ「ばか」と表示しなさい

設問 4:1 から 1000 の数のうち、3 の倍数のみ「わん」と表示しなさい

次にレジユメには載っていないのですが、基礎プログラミング II で実際にシステムを運用したので、内容と結果を説明します。

まず、実施回数は週ごとに4回で、時間は10分程度でした。受講する学生約40人に同時に使ってもらいました。

設問は主に1から1000を表示し、階数ごとに条件を増やしました。

運用して起こった問題

▶ エラーが発生しないプログラム文があった

▶ ユーザ名検索が使いなかった

▶ サーバ過負荷によりシステムがダウンした

次に、結果です。結果的には、システムに求めていた動きを確認することが出来ました。

一方で問題も起こりました。

まず、プログラムが失敗しているのにエラー文が表示されないものがありました。while などの制御構造を使う際に演算子を間違えたことで起こりました。

次に、学生側にユーザ名を学籍番号で固定することを伝え忘れたため、各自が任意の文字列を入力したため、ユーザ名検索が行なえませんでした。

システムとしては、40人が一齐に使うことで動作が一気に遅くなったり、動かなくなったりしました

今後の課題

▶ 課題

脆弱性への配慮

- テストを行う
- 運用する環境に応じて対策を講じる

▶ 使用する条件の発見

- 機能を見直す

展望

▶ 展望

運用を見据えてシステムの構築

- 教員が問題を挿入するためのページ作成
- 実際に運用した上で出てくる新しい問題への対応

このシステムを作って実際に運用した上で、今後の課題と展望です。

課題としては、脆弱性への配慮があります。プログラムを入力する場所に悪意のある文字列を入力することで、意図しない動きをする可能性があります、この対策を運用する環境に応じて考えます。現時点では大学の roy 環境のような閉鎖的な場所で使うことを前提にします。

運用した際のエラー文がでないプログラムに関しては対策が難しいので、そちらも考えていきます。

また、使う際にユーザ名の統一を図るなど、予め条件を決めてつかわなければ意味のない機能もあるため、こういった条件かも考える。

展望としては、実際に運用することを見据えてシステムを見直します。自由に入力可能な問題文やプログラムの想定のある場所を入力する教員ページと、実際に使い、使った側の意見を聞いた上で分かる新しい問題の対策を考えていきます。

参考文献

文部科学省."「高等学校学習指導要領」(平成30年告示)(2019)".
(参照 2019-12-3).

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. (参照 2019-11-15).

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>. (参照 2019-11-18).

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,(参照 2019-11-12).

参考文献は以下のとおりです。