

令和 3 年 5 月 27 日現在

機関番号：33917

研究種目：基盤研究(C) (一般)

研究期間：2017～2020

課題番号：17K01154

研究課題名(和文) Webによる統合プログラミング学習環境を利用したプログラミング・ポートフォリオ

研究課題名(英文) Programming Portfolio based on WebIDE

研究代表者

蜂巢 吉成 (Yoshinari, Hachisu)

南山大学・理工学部・教授

研究者番号：30319298

交付決定額(研究期間全体)：(直接経費) 2,400,000円

研究成果の概要(和文)：初学者が学びやすく、実践的なスキルも獲得できるWeb統合プログラミング学習環境を実現し、学習者のプログラミング・プロセスを分析する研究を行った。初学者にわかりやすいWebに適した入出力ライブラリ関数の設計・実現、および、マウスシミュレータを用いた学習教材を提案した。実践的スキルを獲得のためにブルーフリーダを提案した。学習者のプログラムの編集過程から、正解への近づき度合いを表す進捗度と単位時間あたりに変化したソースコード文字数を活動度を定義し、進捗度を横軸、活動度を縦軸とした平面上に学習者のプログラミング状況を時間毎にプロットしてグラフで表し、学習者のプロセスを分析する方法を考察した。

研究成果の学術的意義や社会的意義

社会的にプログラミングの重要度が増す中で、本研究で提案したWeb統合プログラミング学習環境により、スマートフォンなどに慣れ親しんだ世代の学習障壁を下げて、理解しやすく、実践的なスキルも獲得できるようなプログラミング教育が可能になる。大学等では教員やTA数人で何十人もの学習者の指導にあたることが多いが、学習者の進捗状況の把握が難しい。本研究で提案した進捗度と活動度の観点から学習者の状況がある程度わかるので、教員による支援が必要な学習者を効率的に発見することができる。さらに多くの学習者のデータが集まれば、学習者のプログラミング言語の概念に対する理解度も分析できるようになると考えている。

研究成果の概要(英文)：We have implemented Web-based Integrated Development Environment which is easy to learn programming and enables a learner to acquire practical programming skills. We have proposed new input functions which are suitable for Web, learning materials using a micro-mouse simulator for novice programmers, and a proofreader to write good program codes. We have also proposed a way to analyze learners' programming processes using a two-dimensional graph, whose horizontal axis is degree of progress, which shows how much a learner makes progress compared to a model answer program, and the vertical axis is degree of activity, which shows how many characters in source code are changed per unit time.

研究分野：ソフトウェア工学

キーワード：プログラミング学習 WebIDE ポートフォリオ

1. 研究開始当初の背景

スマートフォンやクラウド環境が身近になり、2020年度からの小学校プログラミング教育必修化の検討や、Scratchなどのビジュアルプログラミング環境が注目を集めるなど、プログラミングへの社会の関心が高まっている。われわれが所属する学科では、1年次の最初にScratchによる簡単なプログラミングを教え、その後、C、Javaなどを教えていく。そこで次のような問題を実感している。

(1) 学習者の普段のコンピュータ利用環境(スマートフォンなど)とプログラミング環境が大きく異なり、学習意欲の低下、理解の低下を招きかねない。Scratchは楽しみながら学習するが、CやJavaなどの学習に抵抗を感じる学生が多い。特にCやJavaなどの学習において必要となるコマンドやプログラムの入出力(コンソール端末による入出力)といった概念がスマートフォンなどに慣れた学習者には理解しにくいようである。

(2) 「きれいな」プログラムを書く、テスト技法を実際のプログラムに適用するなどの実践的なスキルを系統的に教えることが難しい。現在のプログラミング演習ではプログラムの作成が中心で、課題の実行例通りの出力が得られればプログラムができたと思っている学習者も多い。

(3) 複数のプログラミング言語を教えているが、学習者がそれらを有機的に結びつけてプログラミング言語の基本的概念を総合的に理解することが難しい。Scratchなどにより、条件分岐、繰り返し、イベント駆動、並列処理、オブジェクトとメッセージパッシングなどのプログラミング言語の基本的概念を学ぶことができるが、Scratchで学んだことがCやJavaなどの学習に活かされているとは必ずしも言えない。CとJavaでも相互補完的に理解することはそれほど簡単ではない。

2. 研究の目的

上記の問題を解決するためにWebによる統合プログラミング学習環境(WebIDE4LP, Web-based Integrated Development Environment for Learning Programming)によるプログラミング・ポートフォリオについて研究する。

(1)の解決のために、学習者が興味を持ちやすいよう、プログラムの入力にはWebに適したフォームを作成して行い、出力はテキストの他に、グラフィクスなどを描画する方法を提案する。

(2)の解決のために、模範解答プログラムと比較して「きれいに書かれたプログラム」であるかをチェックするプルーフリーダなどにより、学習者に実践的スキルを獲得させる方法を提案する。

(3)の問題の解決支援を図るためにプログラミング・ポートフォリオ(pポートフォリオ)の実現について考察する。ソースプログラムなどの学習者のプロダクトとそれを作成する過程(プロセス)を分析して、プログラミング言語の基本的概念の理解度を測定する方法について考察する。

3. 研究の方法

(1)について、Webに適したC言語学習における入力方法の概念を整理し、初学者にもわかりやすくWebに適した入出力ライブラリ関数を設計・実現した。学習者のプログラムから自動でテキストボックス(1つの値)やテキストエリア(値のリスト)のフォームを生成する環境を実現した。さらに、学習意欲促進のためにマイクロマウスシミュレータを用いた学習教材を提案した。学習者はJavaScriptプログラムによりマイクロマウスを操作し、迷路を解く。シミュレータで使用する迷路の大きさやゴールの位置を変えることで、手続き型言語の基礎概念である順次、分岐、反復や配列に加えて、左手法などのアルゴリズムも学習できる教材を作成した。

(2)について、実践的なスキルを獲得するために、学習者が作成した「きれいでない」プログラムに対して、効果的にフィードバックするプルーフリーダを試作した。制御構造に加えて、式などの細かい粒度で教育者が判断基準を独自に追加できるカスタマイズ機能と、識別子名などの特定の字句の出現回数を計測して、「きれいな」プログラムかどうかを判断する方法を提案した。また、文法学習のための学習者の構文理解を支援するために、学習者のプログラムから構文図式を表示するツールを提案した。

(3)について、学習者のプログラミングプロセスを整理して分析するために、学習者プログラムと正解プログラムとの近づく具合を表す進捗度と学習者が編集作業などをどの程度行っているかを表す活動度の2つの観点を考え、横軸を進捗度、縦軸を活動度とする平面上に学習者の状況をプロットしてグラフ化する方法について考察した。

4. 研究成果

(1-1) WebベースのC言語プログラミング学習環境のための入力フォーム自動生成方法の提案

(1)について、上記題目で日本ソフトウェア科学会コンピュータソフトウェアに論文を投稿し

た。Web ベースの C 言語プログラミング学習環境に適した入力フォームを、ソースプログラムの入力関数の呼出しから自動生成する方法を提案した。Web によるプログラミング環境も普及しつつあるが、プログラム実行における入力コマンドラインを前提としているので、Web に適していないという問題点がある。われわれは C 言語学習におけるプログラム実行の入力モデルを従来の端末による対話的なストリームモデルから Web の一括入力に適した連想配列モデルに変更した。従来の C 言語学習で多く用いられていた scanf 関数に代わる入力関数群を次のように提案した。TYPE は int, double, string(文字列)などの型を表す。

TYPE getTYPE(string label)

引数 label を名前としたテキストボックスを生成し、テキストボックスに入力された 1 つのデータを読み込んで返す。

TYPE nextTYPE(string label)

引数 label を名前としたテキストエリアを生成し、テキストエリアに入力された複数のデータを 1 つずつ読み込んで返す。空白文字を区切り文字とする。読み込むデータがないときは、実行時エラーでメッセージを表示してプログラムを終了する。

int hasNext(string label)

引数 label を名前としたテキストエリア内に読み込まれていないデータがあるかを調べる。データがある場合は 1 を、ない場合は 0 を返す。

getTYPE 関数と nextTYPE 関数は入力されたデータをそれぞれの型で返すので、入力を a = getInt("a"); のように代入の形式で記述できる。各関数の引数 label はフォーム名であり、ブラウザでは label= として出力される。代入の分かりやすさから変数名を記述することを想定している。nextTYPE, hasNext 関数はテキストエリアで複数の値を入力する際にイテレータとして使えるように Java の Iterator インタフェースを参考に設計した。

Web プログラミング環境は実行前にソースプログラムを解析して、入力関数の呼出しから Web の入力フォームを動的に生成する。フォーム名が連想配列のキーに、入力されたデータが値となる。

提案した入力方法による Web プログラミング環境を設計・実現した。ソースプログラムの編集、コンパイル、実行などの基本的な機能に加え、フォーム生成を行う機能を持ち、それぞれの機能は Ajax 通信により処理される。Web プログラミング環境の画面例を右上図に示す。提案した入力方法を用いて、参考書に記載されているプログラム例を書き直し、コンパイル・実行できることを確認した。

(1-2) マイクロマウスシミュレータを用いたプログラミング学習教材の提案

上記題目で情報教育シンポジウム SSS2020 にて論文発表した。マイクروマウスシミュレータは、マイクロマウス競技に用いられる小型の移動ロボットが迷路を駆け抜けるための迷路探索プログラムのシミュレーションを行うためのものである。マイクロマウスシミュレータを用いることで、プログラムの実行結果がマイクロマウスの動きとなり、視覚的にわかりやすく理解しやすい。右図が実行環境の画面例である。下は迷路で、車がマイクロマウスで旗がゴールである。学習者は JavaScript でマウスを操作するプログラムを記述し、実行すると車が移動する。

手続き型言語の基礎概念である順次、分岐、反復や配列に加えて、左手法などのアルゴリズムも学習できる教材を作成した。

(2-1) プログラミング学習用ブルーフリーダの試作

(2)について、上記題目で日本ソフトウェア科学会コンピュータソフトウェアに論文投稿した。プログラミング学習において学習者が作成したプログラムが教育者の意図に合致しているかをチェックするブルーフリーダを試作した。プログラミング学習では、学習者は課題の実行結果例などを参考にプログラムの動作確認をし、同じ出力が得られたら

プログラム入力欄

```
/* 柴田望洋: 新・明解 C 言語 入門編 p. 86 List4-9改
  指示された個数だけ整数を読み込んで合計値と平均値を表示 */
#include <stdio.h>
#include "webio.h"
int main(void) {
    int i = 0, sum = 0, num, tmp;
    /* printf("整数は何個ですか : "); scanf("%d", &num); */
    num = getInt("num");
    while (i < num) {
        /* printf("No.%d : ", ++i); scanf("%d", &tmp); */
        ++i;
        tmp = nextInt("tmp");
        sum += tmp;
    }
}
```

COMPILE

FORM

RUN

実行結果

合計値 : 1074

平均値 : 179.00

変数入力欄

num=

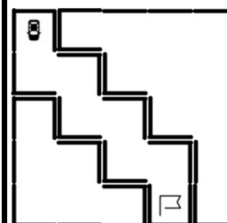
tmp=

ソースコード編集エリア

```
1 mm.go(1);
2 mm.rotate(LEFT);
```

実行 テキスト読み込み テキスト保存 リセット

迷路



課題を終了していることが多い。しかし、学習者が作成したプログラムは教育者が意図した「きれいに書かれたプログラム」とは限らない。「きれいに書かれたプログラム」の判断基準は課題毎に異なるが、本研究ではこれを整理し、教育者が作成した模範解答プログラムと学習者のプログラムを比較して、教育意図に合っていない「汚く書かれたプログラム」を発見して指摘するブルーリーダを試作した。

実数配列に入力して、平均値と最小値を求める次の課題を考える。

実数値を配列に入力し、その平均値と最小値を出力するプログラムを作成しなさい。
配列の大きさにマクロを用い、次の関数を定義し、利用すること。

```
/* 要素数 size の実数配列 d の平均値を求める */
double avgDbfArray(double d[], int size)

/* 要素数 size の実数配列 d の最小値を求める */
double minDbfArray(double d[], int size)
```

実行結果例:

```
input data: 56.7 85.9 45.9 98.9 88.4
avg=75.160000, min=45.900000
```

模範解答は次の通りである。

```
double avgDbfArray(double d[], int n) {
    int i;
    double sum;
    sum = 0;
    for (i = 0; i < n; i++)
        sum += d[i];
    return sum / n;
}
```

```
double minDbfArray(double d[], int n) {
    int i;
    double min;
    min = d[0];
    for (i = 1; i < n; i++)
        if (d[i] < min)
            min = d[i];
    return min;
}
```

次のプログラムは学習者によるプログラム例である。実行結果は正しいが「汚く書かれたプログラム」である。

```
double avgDbfArray(double d[], int n) {
    int i;
    double sum, avg;
    sum = d[0];
    for (i = 1; i < n; i++) {
        sum += d[i];
        avg = sum / n;
    }
    return avg;
}
```

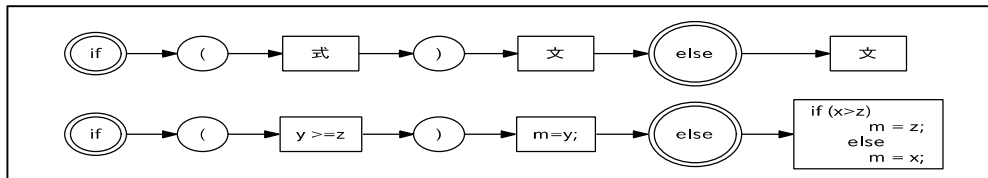
```
double minDbfArray(double d[], int n) {
    int i;
    double min;
    min = d[0];
    for (i = 0; i < n; i++)
        if (d[i] < min)
            min = d[i];
    return min;
}
```

$X=d[0]$; for (i = 1; ... については、最小値を求める関数ではきれいなコードであるが、平均を求める関数ではきれいではない。同じような記述でも問題によってきれいかきれいでないかが変わるので、同一の基準で判定することはできない。

本研究では、課題の教育意図が反映されたプログラム箇所を抽象化した記述を意図パターンとして定義した。学習者のプログラムと模範解答プログラムに対して、意図パターンにマッチしたコード断片を比較することで、教育意図に合ったプログラムかをチェックする方法を提案した。問題毎に適用する意図パターンを変えることで、上記プログラムでも問題それぞれについて個別にきれいさを判定することができる。

(2-2) プログラミング学習における構文図式を用いた構文理解支援方法の提案

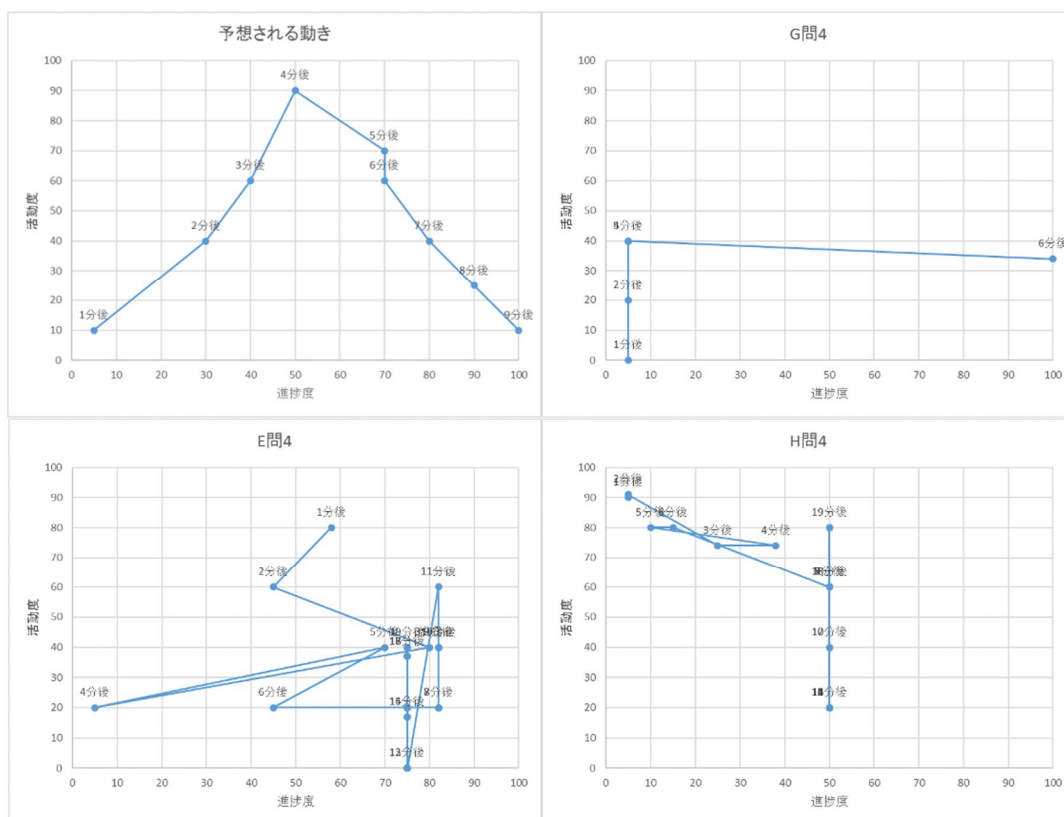
上記題目で情報教育シンポジウム SSS2018 にて論文発表した。文法学習のための学習者の構文理解を支援するために、構文図式の非終端記号のノードに学習者が記述したプログラム断片を記述した図式(構文図式インスタンスと呼ぶ)を提案した。構文図式と構文図式インスタンスを表示するツールを試作し、構文の誤りによるコンパイルエラーや期待する実行結果を得られないプログラムなどに利用することで学習者の構文理解を支援する。本ツールは Web で公開している。下図はツールの実行結果例である。



(3) 学習者のプログラミング過程の分析

本研究では、学習者のプログラミングの概念の理解度を測定するために、プログラミング過程を進捗度と活動度を用いて分析する方法について考察した。学習者のソースコードの模範解答への近付き具合を進捗度として定義した。識別子名やコーディングスタイルの違いを吸収するためにソースコードを正規化する。プログラミングの概念の分析では制御構造を正しく記述することが重要と考えて、制御文の予約語と入れ子関係を模範解答と学習者のソースコードから抽出し、その編集距離として進捗度を定義した。ただし、学習者が別解を記述している場合は、この進捗度の定義では変化がないので、コンパイル成功や、実行結果が模範解答の実行結果と一致した場合に進捗度の値が上昇するような定義も加えた。活動度は単位時間あたりに変化したソースコード文字数として定義した。

進捗度を横軸、活動度を縦軸とした平面上に学習者のプログラミング状況を時間毎にプロットして、グラフで表して分析する方法について考察した。下図のそのグラフである。左上は予想される動きを示したグラフである。学習者は問題に取り組んだときは考えているので活動度は低いが、解法がわかると活動度が増え、進捗度も増える。正解に近づくとも入力に必要な文字が減るので、活動度は減少すると予想した。右上、左上、右下は実際の演習問題を解いているときの学習者のグラフである。予想した動きとは異なり、進捗度の増減を繰り返すなど、試行錯誤していることがわかった。グラフを分析し、サポートが必要な学習者を特定する方法について考察し、「プログラミング演習における個別指導のためのコーディング状況把握方法の提案」として情報処理学会コンピュータと教育研究会にて発表した。



全体の評価と今後の展望

以上述べた研究成果から、研究目的の(1)(2)は達成したと考える。(3)のプログラミング・ポートフォリオは研究中である。プログラムの作成過程を進捗度と活動度から分析する方法を提案したが、進捗度の計算で抽出した制御構造の予約語とプログラミングの分岐や反復の概念を対応づけて、その変化を調べることで概念理解度を測定する方法について検討している。

5. 主な発表論文等

〔雑誌論文〕 計9件（うち査読付論文 9件／うち国際共著 0件／うちオープンアクセス 6件）

1. 著者名 蜂巢 吉成, 石元 慎太郎, 吉田 敦, 桑原 寛明	4. 巻 46
2. 論文標題 プログラミング学習者の編集途中のスースコードと模範解答における変数の対応づけ方法の提案	5. 発行年 2020年
3. 雑誌名 ソフトウェア工学の基礎 XXVII (FOSE 2020)	6. 最初と最後の頁 109-114
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 小山 哲明, 蜂巢 吉成, 吉田 敦, 桑原 寛明	4. 巻 2020
2. 論文標題 マウスシミュレータを用いたプログラミング学習教材の試作	5. 発行年 2020年
3. 雑誌名 情報教育シンポジウム (SSS2020)	6. 最初と最後の頁 54-61
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 蜂巢 吉成, 吉田 敦, 桑原 寛明, 阿草 清滋	4. 巻 36
2. 論文標題 WebベースのC言語プログラミング学習環境のための入力フォーム自動生成方法の提案	5. 発行年 2019年
3. 雑誌名 コンピュータ ソフトウェア	6. 最初と最後の頁 4_18 ~ 4_24
掲載論文のDOI (デジタルオブジェクト識別子) 10.11309/jssst.36.4_18	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 蜂巢 吉成, 森本 朱音, 松尾 翔馬, 加藤 ちひろ, 吉田 敦, 桑原 寛明	4. 巻 45
2. 論文標題 ポインタ型の仮引数を持つ関数の呼出しに対するプログラミング学習支援ツールの提案	5. 発行年 2019年
3. 雑誌名 ソフトウェア工学の基礎 XXVI (FOSE 2019)	6. 最初と最後の頁 103-108
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 安達 有希、蜂巢 吉成、吉田 敦、桑原 寛明、阿草 清滋	4. 巻 2018
2. 論文標題 プログラミング学習における構文図式を用いた構文理解支援方法の提案	5. 発行年 2018年
3. 雑誌名 情報教育シンポジウム論文集	6. 最初と最後の頁 118 ~ 125
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 石元 慎太郎、蜂巢 吉成、吉田 敦、桑原 寛明、阿草 清滋	4. 巻 2018
2. 論文標題 プログラミング演習における構文要素の種類毎のビューによるコーディング状況把握方法の提案	5. 発行年 2018年
3. 雑誌名 情報教育シンポジウム論文集	6. 最初と最後の頁 158 ~ 165
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 蜂巢 吉成、吉田 敦、桑原 寛明、阿草 清滋	4. 巻 35
2. 論文標題 プログラミング学習用ブルーフリーダの試作	5. 発行年 2018年
3. 雑誌名 コンピュータ ソフトウェア	6. 最初と最後の頁 129 ~ 135
掲載論文のDOI (デジタルオブジェクト識別子) 10.11309/jssst.35.129	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 吉田 敦、加藤 大貴、蜂巢 吉成、桑原 寛明、阿草 清滋	4. 巻 2018
2. 論文標題 字句列の共通性に基づく例外処理条件文の抽出手法の提案	5. 発行年 2018年
3. 雑誌名 ソフトウェアエンジニアリングシンポジウム2018論文集	6. 最初と最後の頁 15 ~ 23
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 蜂巢吉成, 吉田敦, 桑原寛明, 阿草清滋	4. 巻 43
2. 論文標題 プログラミング演習用ブルーフリーダの試作	5. 発行年 2017年
3. 雑誌名 ソフトウェア工学の基礎 XXIV (FOSE 2017)	6. 最初と最後の頁 53-62
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

〔学会発表〕 計2件 (うち招待講演 0件 / うち国際学会 0件)

1. 発表者名 久保田 詩門, 蜂巢 吉成, 吉田 敦, 桑原 寛明
2. 発表標題 プログラミング演習における個別指導のためのコーディング状況把握方法の提案
3. 学会等名 情報処理学会 コンピュータと教育研究会
4. 発表年 2019年

1. 発表者名 石元 慎太郎, 蜂巢 吉成, 吉田 敦, 桑原 寛明
2. 発表標題 プログラミング学習者の編集途中のソースコードに対するフィードバック方法の提案
3. 学会等名 日本ソフトウェア科学会 ソフトウェア工学の基礎 XXVI (FOSE 2019)
4. 発表年 2019年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

構文図式インスタンスの表示
<https://akg.se.nanzan-u.ac.jp/syndia/>

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究 分 担 者	吉田 敦 (YOSHIDA ATSUSHI) (50283495)	南山大学・国際教養学部・教授 (33917)	

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関