社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

Javaプログラミングの予約語学習のための オンライン穴埋め問題機能の実装

伊永　洋輔† 　松島由紀子†† 　舩曳　信生†† 　中西　　透†† 　天野　憲樹†††

† 岡山大学工学部通信ネットワーク工学科　〒 700–8530 岡山市津島中 3–1–1
†† 岡山大学大学院自然科学研究科　〒 700–8530 岡山市津島中 3–1–1
††† 岡山大学教育開発センター　〒 700–8530 岡山市津島中 2–1–1
E-mail: †korenaga@sec.cne.okayama-u.ac.jp, ††matsushima@bemax.ac.jp,
†††{funabiki,nakanisi}@cne.okayama-u.ac.jp, ††††amano@cc.okayama-u.ac.jp

あらまし　Java は，信頼性，可搬性，学習性に優れたプログラミング言語として，多くの大学や専門学校で教育が行われている．そのため，本グループではテスト駆動型開発手法を用いた *Java* **プログラミング教育支援システム**を提案している．本システムは，Java 言語教育をある程度，受けている学生を対象としていることから，本研究では，その初学者教育において重要な予約語の学習支援を目的として，**オンライン穴埋め問題機能**を提案する．本機能は，教員サービス機能と学生サービス機能で構成される．前者では，学生に提示する穴埋め問題を，予めデータベースに登録した Java コードから指定した予約語をランダムに選択し空欄とすることで，自動作成を可能としている．後者では，学生の自宅学習支援のために，正しく解答できるまで繰り返し解答することを可能としている．既存のシステムに組み込み，Java 言語初学者の学生を対象に評価実験を行い，アンケートを通じてその有効性を検証した．
キーワード　Java, プログラミング教育, e-ラーニング, 穴埋め問題, 問題自動生成

An Implementation of an Online Fill-in-the-blank Problem Function for Learning Reserved Words in Java Programming

Yousuke KORENAGA†, Yukiko MATSUSHIMA††, Nobuo FUNABIKI††, Toru NAKANISHI††, and Noriki AMANO†††

† Faculty of Engineering, Okayama University
†† Graduate School of Natural Science and Technology, Okayama University
††† Center for Faculty Development, Okayama University
E-mail: †korenaga@sec.cne.okayama-u.ac.jp, ††matsushima@bemax.ac.jp,
†††{funabiki,nakanisi}@cne.okayama-u.ac.jp, ††††amano@cc.okayama-u.ac.jp

Abstract　The Java programming has been educated to students in a lot of universities and professional schools due to its reliability, portability, and easy learning. To assist the education, our group has developed a *Java programming education assistant system* based on the test-driven development method. However, this system targets students who have studied Java to some extent. In this paper, we propose the *online fill-in-the-blank problem function* for this Web system to assist the learning of the reserved words of Java for Java beginners. This function consists of the teacher service function and the student service function. By the former function, a teacher can generate a fill-in-the-blank problem automatically by removing the specified reserved words randomly from a sample Java code in the database. By the latter function, a student can repeatedly submit answers of a problem until all the answers are correct for self-study. The function is implemented at the existing system and is applied to students who are Java beginners, where the questionnaire result confirms the effectiveness of our proposal.
Key words　Java, programming education, e-learning, fill-in-the-blank problem, automatic problem generation

1. まえがき

Java は，信頼性，可搬性，学習性に優れたプログラミング言語として，エンタープライズシステムから組み込みシステムまで広範囲に使用されており，産業界からその技術者の育成が強く求められている．そのため，実際に多くの大学や専門学校で Java 言語の教育が行われている．これらの Java 言語教育では，教室における教科書や参考書を用いた文法教育と，コンピュータ（PC）を用いて実際にプログラムコードを書くプログラミング演習の組み合わせにより，学習が進められる．この中で後者のプログラミング演習では，基本的には次のサイクルで学習が進められる．

（1） 学生が演習課題を解く（必要があれば質問をする）
（2） 紙または電子データで教員に解答を提出する
（3） 教員が提出された解答を手作業で評価し，学生にフィードバックする
（4） 学生が解答を修正し，再提出する

このプログラミング演習における学習方法の問題点として，教員の負担が大きいことと，それによる学習機会の喪失の可能性が挙げられる．プログラミング演習では，通常，教員 1 名に対して学生が数十名といった 1 対多の形式が多く，学生全員に対して十分な指導が困難である．また，演習課題の採点や評価にも，教員は相当の時間を費やしている．その結果，学生への評価結果のフィードバックが遅れた場合，学習内容が十分に理解されなかったり，モチベーションの低下につながったりするなど，学習環境の悪化を招く恐れがある．特に，学生や課題の数が増えるに従って，それらの評価作業や評価結果の管理が益々煩雑となり，人為的なミスが生じる可能性が高くなる．

そこで，本グループではこれまでに，テスト駆動型開発手法を用いた，*Java* **プログラミング教育支援システム**の提案を行ってきた [4] [5]．本システムでは，Java プログラム検証ツール JUnit を採用することで，学生から提出された Java コードの自動検証を行っている．Web システムとすることで，学生には自宅からでも演習課題に対する Java コードの提出を可能としている．その提出後，直ちにオンラインでその検証結果をフィードバックすることで，繰り返し学習を可能とし，学習環境の向上を実現している．

ここで，本システムは，課題で求める仕様を満たす，class 全体の Java コードの作成を求める課題での利用が可能である．そのため，ある程度の Java 言語の学習が進んだ段階でのみ，使用可能となり， Java 言語の学習を始めばかりの学生（Java 初学者）を支援することは困難である．Java 初学者は，まず，Java 言語の予約語を覚える必要がある．通常，文法教育では，新しい予約語を紹介しながら，その使い方，規則などの教育を行っている．

そこで，本研究では，Java 言語の予約語学習支援を目的として，Java プログラミング教育支援システムにおける，*Java* **言語予約語オンライン穴埋め問題機能**を提案する．予約語は固定の文字列であり，まずは，それらの暗記が必要となる．そのため，本機能で提供する問題は，Java コードの穴抜きされた箇所に対して，適切な予約語を追加する，穴埋め問題としている．提案機能は，教員サービス機能と学生サービス機能で構成される．前者では，学生に提示する穴埋め問題を，予めデータベースに登録した Java コードから，指定した予約語をランダムに選択し，空欄とすることで自動作成を可能としている．後者では，学生の自宅学習支援のために，正しく解答できるまで，繰り返し解答することを可能としている．

提案する機能を，既存の Java プログラミング教育支援システムに実装し，専門学校で学ぶ Java 初学者に対する評価実験を行った．本実験では，全被験者に対して，本機能で作成した予約語の穴埋め問題への解答を行って貰った後， 5 段階評価のアンケートを実施した．その結果，本機能は，Java 言語の予約語学習に役立つこと，インタフェースや問題作成機能に更なる改善が必要であること，などが明らかとなった．

本論文の章構成を示す．まず，2. で提案機能を述べる．次に，5. で評価実験とその考察を述べる．6. で関連研究を述べる．最後に，7. で結論と今後の課題を述べる．

2. 提案機能の概要

本章では，提案する Java プログラミング初学者教育支援機能の実装環境と機能構成について述べる．

2.1 実装環境

本節では，提案機能の実装環境について述べる．図 1 に示すように，サーバ OS に Linux，Web サーバ兼アプリケーションサーバに Tomcat を利用し，サーバプログラムは JSP と Servlet を用いて記述している．また，データベースには MySQL を用いている．これらは，すべてオープンソースソフトウェアである．ユーザは，Web ブラウザを用いてサーバにアクセスし，本機能のサービスを受けることとしている．

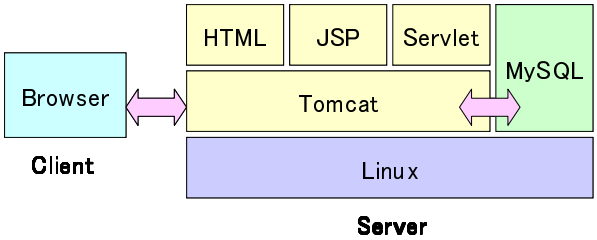

図 1 提案機能の実装環境

2.2 機能構成

提案機能の構成を図 2 に示す．本機能は，ユーザとして教員向けの**教員サービス機能**と学生向けの**学生サービス機能**で構成される．ユーザ登録などの機能は，既存のシステムのものを利用している．

本機能における**問題**，**課題**について，次のように定義する．1 つの**問題**は，問題コード，正答，問題コメントで構成される．問題は，Java コードにおける，予約語の穴埋め問題である．1 つの**課題**は，課題名，1 つ以上の問題，課題コメントで構成される．1 つの Java 言語教育科目では，通常，複数の課題が提示されることを想定している．

教員サービス機能では，問題作成，課題登録，成績参照の各サービスが提供されている．**学生サービス機能**では，課題解答，成績参照各サービスが提供されている．

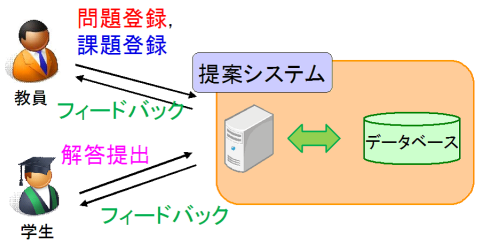

図 2　提案機能の構成

2.3　利 用 手 順

本機能の教員，学生の両サービス機能全体での利用手順を述べる．以降の章で，各サービスの詳細について述べる．

（1） 教員による Java コードの登録
（2） 教員による予約語と Java コードの選択
（3） 教員による問題作成
（4） 教員による作成済み問題からの課題登録
（5） 学生による課題の解答
（6） 教員，学生による成績確認

3.　教員サービス機能

本章では，教員サービス機能の詳細について述べる．

3.1　Java コードの登録

まず，教員は，問題に使用するのに適切な Java コードを，データベースに登録する．この問題作成に使用する Java コードを，以後，**サンプルコード**と呼ぶこととする．データベースへのサンプルコード登録時に，そのコードに含まれているすべての予約語の出現回数を数え，データベースに登録しておく．これにより，次の問題作成時に，学習させたい予約語がそのコードにどの程度含まれているかを瞬時に知ることが可能となる．

サンプルコードからの予約語の抽出は，コード中のアルファベットのみの単語を抜き出し，全予約語を記述した予約語リストに含まれているか否かを確認することで行う．予約語と判定された場合，その単語を含む行番号と，その行の先頭からの文字数をデータベースに登録する．

3.2　予約語と Java コードの選択

次に，図 3 の画面において，教員は，今回の問題で学習させたい予約語，サンプルコードの中で，その予約語の総数に対して空欄とする（穴埋め問題化する）割合を入力する．その入力後，教員は，今回選択された予約語を 1 つ以上含むサンプルコードが，図 4 のように示されるため，その中から問題に含むサンプルコードを選択する．

3.3　問題の作成

サンプルコードからの問題作成では，まず，今回学習させたい各予約語を 1 つずつ，コードの中からランダムに選択し，そこを空欄とする．これは，学習させたい予約語が作成された問題で出されない可能性を除くためである．その結果，サンプルコード中の空欄とした予約語数の割合が，予め設定した割合に到達していない場合，それに到達するまで，ランダム選択による空欄化を繰り返す．この予約語が空欄とされたコードのことを**問題コード**と呼ぶこととする．

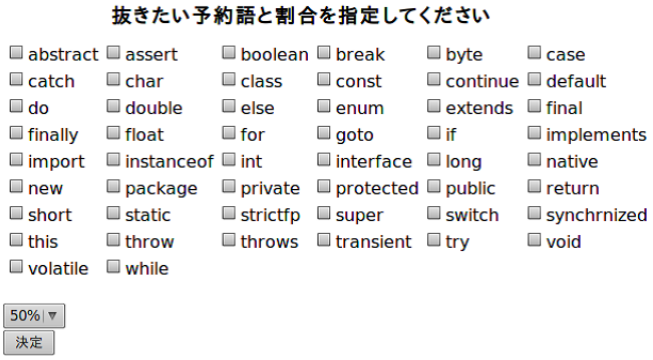

図 3　予約語の選択

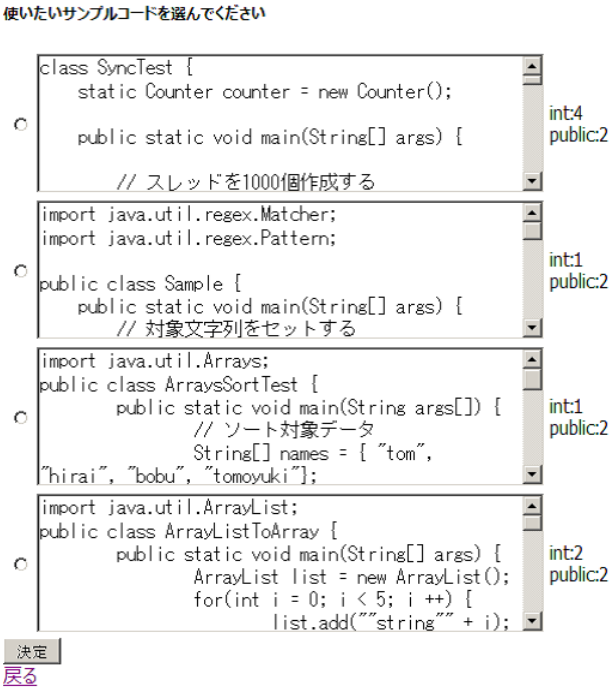

図 4　サンプルコードの選択

空欄とした予約語の位置には，問題番号を挿入する．同時に抜き出した予約語を，正答データとして文字列に変換し，その問題番号と共に保持しておく．そして，図 5 の画面を用いて，教員は，問題コード，正答データのプレビューした後，問題内容を把握するための問題に対するコメントを入力し，問題データベースへ登録する．これで 1 つの問題の作成が完了する．

3.4　課題の登録

本サービスでは，教員は，データベースに登録されている問題を選択することで，課題を登録することができる．課題登録時には，課題名，課題に対するコメントを付加し，課題としてデータベースに登録される．なお，各課題は，科目と紐付けており，その科目の担当教員のみが参照できるようになっている．

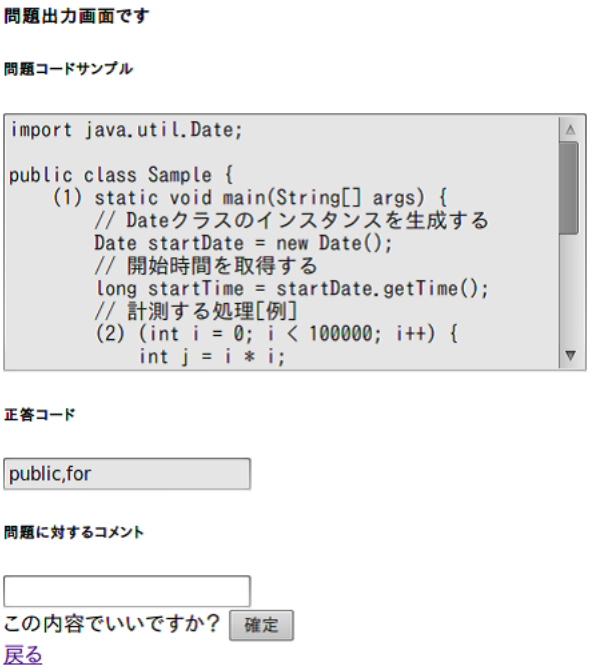

図 5　問題プレビュー

3.5　成績の参照

教員は，受講学生の学習進度を確認するために，各課題に対する学生の解答結果を閲覧することができる．ここでは教員は，課題毎の提出学生数，平均点を，1 つの科目での全課題に対して，一覧で閲覧できる．また，課題毎の詳細として，各学生の問題毎の正答情報，課題の提出日時が表示される．成績参照画面の例として，図 6 を示す．

学籍番号	設問1	設問2	設問3	設問4	設問5	設問6	設問7	設問8	設問9	設問10	繰り返し回数
50000003	○	○	○	○	○	×	×	×	×	×	6
50000004	○	○	○	○	○	○	○	○	○	○	16
50000002	×	○	○	○	○	○	○	○	○	○	7
50000002	○	○	○	○	○	○	○	○	○	○	2
50000000	○	○	○	○	○	○	○	○	○	○	6
50000001	○	○	○	○	○	○	○	○	○	×	2
50000006	×	×	×	×	×	×	×	×	×	×	2
20000011	○	○	○	○	○	×	×	○	○	×	2
20000007	○	○	○	○	○	○	○	○	○	○	4
20000002	○	○	○	○	○	○	○	○	○	○	7
20000011	○	○	○	○	○	○	○	○	○	○	8
09421922	○	○	○	○	○	○	○	○	○	×	3
09421944	○	○	○	○	○	○	○	○	○	○	14

この課題の平均提出回数:6

図 6　成 績 参 照

3.6　科目・学生管理

提案機能の実装では，データベースを用いて，科目の管理，履修学生の管理を行っている．教員は，新規に本機能を利用する科目を担当する際，教員の情報，科目名，履修学生のリストを，データベースに登録する．それにより，教員は登録科目に対して，課題作成や提出結果の閲覧を行うことができる．同時に，登録された学生は，その科目に対して，課題への解答や結果の閲覧が可能となる．

また，ある教員が作成した問題は，他の教員にも利用可能としており，複数の科目で同じ問題を出題することが可能である．その結果，問題のデータベース（問題集）の充実後には，新規に問題を作成することなく，課題を出すことが可能となり，教員の大幅な負担軽減につながることが期待される．

4.　学生サービス機能

本章では，学生サービス機能の詳細について述べる．

4.1　問題の解答

まず，課題に対する，学生の解答提出での問題反復解答のサービスについて述べる．予約語は，Java 言語の改定が起きない限り，変更にはならないため，暗記を基本とした学習が有効である．そのため，課題のすべての予約語を，正しい綴りで解答できるまで，繰り返し，解答の提出と採点を可能とした．

学生は，Web ブラウザを用いてサーバにアクセスした後，本機能を利用する受講科目の一覧を見ることが可能となる．その中から，対象の科目を選択すると，その科目で出されている課題の一覧が表示される．ここでは，課題毎に，課題番号，課題名，提出状況が表示され,「解答」ボタンのクリックにより，図 7 で示す，その課題の解答画面に遷移し，解答を行うことができる．

本画面では，課題名，教員からの課題に対するコメント，問題コード，解答フォームが表示される．学生は，課題に対するコメントをヒントとしながら，問題コード中に埋め込まれた空欄（設問）を見つけ，対応する解答フォームにその解答を入力する．本実装では，問題コードの表示に CodePress を用いており，コードハイライト機能によって問題コードを読みやすくしている．

解答フォームには,「採点」ボタンと「確定」ボタンがあり,「採点」ボタンがクリックされた場合，解答フォームに入力されている文字列が正答と一致しているか否かを判断し，一致している場合は ”OK ”を，異なっている場合は ”NG ”を表示する.「確定」ボタンがクリックされた場合，実施中の課題を終了し，最後に採点した時の解答情報，繰り返し数カウンタ，解答終了フラグ，解答提出時間を登録する．学生は，全問正解するか，解答が不明で諦めた時に,「確定」ボタンを入力する．

ここで，繰り返し数カウンタは,「採点」ボタンのクリック回数を数えるためのカウンタであり，学生が「確定」ボタンを押すまでに何回解答を訂正し，採点を行ったかを把握するためのものである．なお,「確定」ボタンを押すまで，解答は何回でも行うことが出来る．

4.2　成績の参照

学生は，図 8 に示す，課題毎の解答正誤表示画面を閲覧することで，解答結果を参照することができる．各課題の提出後，学生は各課題でどの程度正解していたか知ることで，復習に役立てることが期待される．そのため，教員に申し出ることで，課題に再チャレンジできる．

5.　評 価 実 験

提案機能の評価実験として，Java 言語初学者となる学生 18 名（本学学生 8 名，専門学校生 10 名）を対象に本機能の適用を行い，アンケートによる 5 段階評価を行った．

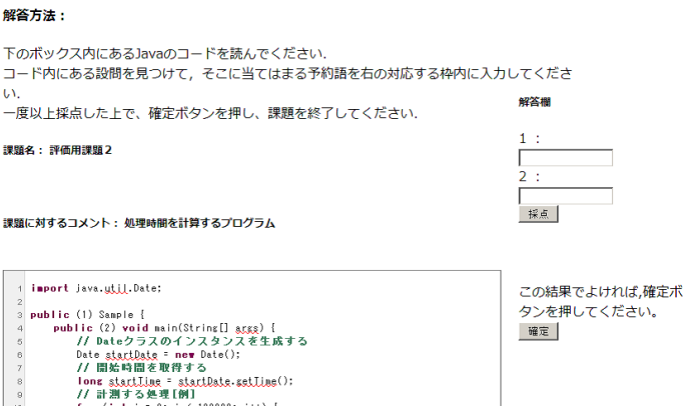

図 7　解 答 画 面

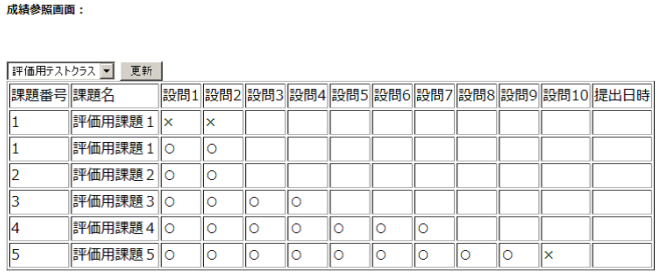

図 8　解答正誤表示画面

表 1　課題ごとの正答，繰り返し数

課題番号	問題数	平均正答数	平均繰り返し回数	最大繰り返し数
1	2	1.8	2	4
2	2	2	2	5
3	4	3.7	3	8
4	7	6.1	4	11
5	10	8.5	6	16

5.1　実 験 課 題

評価実験用の課題として，5 問題の課題を用意した．問 1〜問 3 では，int，class などの比較的出現頻度の高い予約語を選択した．問 4，問 5 では，通常，初学者は使用しない，synchronized といった出現頻度の低い予約語を選択した．また，問題の番号が大きくなるほど，空欄の割合を増やした．

5.2　課題の解答結果

表 1 評価に使用した課題の各問題における正答数，繰り返し回数の平均値を示す．

5.3　アンケート結果

表 2 にアンケートの設問，表 3 に回答結果を示す．表 3 において，Q1 でほとんどの学生が 4 以上を選んでいることから，本機能は Java 言語の予約語学習に役立つものと言える．

Q4，Q5 より，出現頻度の高い予約語を少数空欄とした問題では易しすぎる反面，その逆の場合には，難度が高すぎるといった結果を得られた．そのため，学生の学習進度に応じて，適切に問題とする予約語や空欄の割合を選択することが必要と言える．特に，空欄の割合を大きくしすぎた場合には，元のプログラムコードが理解できなくなり，解答不可となる場合もあり得るため，要注意である．その場合，問題に対する解説を充実させることで対応することも挙げられる．

表 2　アンケート項目

項目	内容
Q1	このシステムは予約語の学習に役立つか
Q2	課題選択から解答提出までの流れはやりやすいか
Q3	解答欄の位置は分かりやすいか
Q4	問 1 は難しかったか
Q5	問 5 は難しかったか
Q6	解答結果の情報は見やすいか

表 3　アンケート結果（5 段階評価）

項目		1	2	3	4	5	
Q1	役立たない	0	1	3	7	7	役立つ
Q2	やりにくい	1	4	1	9	3	やりやすい
Q3	分かりにくい	3	4	3	5	3	分かりやすい
Q4	難しかった	1	0	1	0	16	やさしかった
Q5	難しかった	6	6	3	1	2	やさしかった
Q6	見にくい	1	3	8	3	3	見やすい

また，これらの問題で，解答がニアミス（一部のみ間違い）であった場合や，何度も不正解を提出している場合には，それに応じて適切なヒントを与えられるようになれば，学習の手助けになるものと考えられる．それには，多肢一選択のような選択方式を取り，暗記が完全に出来ていない場合でも，解答できるようにすることが有効と思われる．今後の課題である．

Q2，Q3，Q6 は，本機能のユーザビリティに関するアンケートであるが，全体を通して提出はやりやすいが，解答画面や結果参照画面において評価平均が 3 に近く，やや不満足となる結果となった．これは，問題コードを表示文字が小さく，設問の位置が確認し難かったことや，成績表示においてどの問題で間違ったのかがわからなかったことが原因として考えられる．今後は，これらの改善が必要である．

最後に，自由記述として，「知らないクラスの知らないメソッドの返り値の型が答えになるような問題は難しい」といった意見もあり，空欄となる位置を選択する方法の改善も必要である．

6.　関 連 研 究

文献 [2] では，Moodle を用いて，穴埋め問題でプログラミング支援を行うことを提案している．文献 [1] での問題生成における，PDG(Program Dependence Graph) を用いた空欄箇所の選定後，空欄とした語に対する，正答の自動補完を行う．ここでは，正答として条件文が抜かれた場合には，それと等価な条件文をすべて生成し，正答に加えている．PDG とは，プログラムの中間表現の一つで, 命令をノードとし, プログラム中に存在するデータ依存関係や制御依存関係をエッジで表した有向グラフである．依存関係を読み取り，プログラムにおいて重要な箇所を空欄箇所に選択する方法を取っている．また，Moodle の成績参照機能を拡張し，問題毎にデータを残せるように改変している．文献は，C 言語に適用したものであり，Java 言語に対して有効かどうかは検証されていない．なお，今後，提案

システムによる学習内容を拡張する際に，PDG の利用を検討する．

文献 [3] では，個々の学生の理解状況と学習意欲に応じて演習課題を出題するプログラミング教育支援を提案している．まず，難易度に幅を持つ演習問題を豊富に用意し，協調フィルタリングを用いて，学習者の各問題に対する達成度（理解度）を推定する．次に，学習者の学習意欲をアンケートと課題提出結果より評価する．そして，学習意欲が閾値に満たない学習者には達成度の最も高い課題を，以上の学習者には達成度の最も低い課題を与えることで，学習意欲の向上，プログラミング能力の向上を図る．ここで，協調フィルタリングとは，ユーザの物事に対する傾向や嗜好を過去の行動という形で記録し，そのユーザと似たような行動をとっている他のユーザの情報をもとに，そのユーザの今後の傾向や嗜好を推測するための手段である．本研究の提案機能では，難易度に幅を持つ問題を容易に作成できることから，今後，本文献のアイデアの利用を検討する．

7. ま と め

本研究では，Java 言語の初学者教育において重要な予約語の学習支援を目的として，**オンライン穴埋め問題機能**を提案した．本機能は，教員サービス機能，学生サービス機能で構成される．既存の Java プログラミング教育支援システムに組み込み，Java 言語初学者の学生を対象に評価実験を行い，その有効性を検証した．今後の課題には，問題コードでの空欄設定の適正化，解答ミスに対するヒント機能，多肢一選択問題の設定，問題コードや成績の表示方法の改善，講義での活用などが挙げられる．

文　献

[1] 柏原昭博, 久米井邦貴, 梅野浩司, 豊田順一, “ プログラム空欄補充問題の作成とその評価, ” 人工知能学会論文誌, vol.16, no. 4C, pp.384-391, 2001.

[2] 新開純子, 早勢欣和, 宮地功, “ Moodle におけるプログラム穴埋め問題の生成と活用に関する検討, ” 電子情報通信学会技術研究報告. ET, 教育工学 108(247), 5-10, 2008.

[3] 田口浩, 糸賀裕弥, 毛利公一, 山本哲男, 島川博光, “ 個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援,” 情処論, vol.48, no.2, pp. 958-96, Feb. 2007.

[4] 松島由紀子, 笠井康裕, 舩曳信生, 中西透, 天野憲樹, ”テスト駆動型開発手法による Java プログラミング教育支援システムの提案,” 信学技報, ET2009-8, pp. 7-12, June 2009.

[5] 福山裕輝, 舩曳信生, 中西透, 天野憲樹, ”テスト駆動型開発手法の Java プログラミング教育応用におけるテストコード提出機能,” 情処研報, 2009-CE-103, no. 21, March 2010.