

アルゴリズム／プログラミング学習支援システムの開発*

梶 田 建 夫
程 鵬

概 要

アルゴリズム／プログラミング学習のための支援システムの開発を行った。プログラミング技法のみでなく、流れ図によるアルゴリズムの理解を含めて学習できる機能をもつシステムとしている。このシステムの主な特徴としては、穴埋めの個数と箇所を変化できるように自動生成された穴埋め型問題を学習することや、アルゴリズムの流れ図によるプログラミング技法を学習するサポート環境、ならびに複数言語プログラムの実行が可能になるテスト環境を提供することなどが挙げられる。

キーワード プログラミング学習, アルゴリズム学習, 自動問題生成, 学習支援システム

Development of a Support System for the Learning of Programming and Algorithm

Tateo Kajita
Peng Cheng

Abstract

This paper aims to propose a support system for efficiently acquiring the skills of programming and algorithms. In this system, we introduce a new mechanism to automatically generate the problems having various degrees of difficulty, such that all of the problems are of the same answer type (multiple choice) and are repeatedly given to reinforce learning. In particular, this system also contains some effective functions, such as a support environment by which a learner efficiently learns the programming techniques based on a specified flow-chart of an algorithm, a test environment by which a learner easily tests the validity of the programs coded by using various programming languages, and so on.

Key words: Programming learning; Algorithm learning; Automatic problem generation; Learning support system

1. はじめに

コンピュータで問題を解決しようとする場合、まずその問題を解くためのアルゴリズム（すなわち、問題解決手順）を明示することが必須である。図形を用いて記述したアルゴリズムを流れ

* 本稿は名古屋学院大学大学院教育研究振興補助金（2009年度）による研究成果の一部である。

図（または、フローチャート）といい、プログラミング言語を用いて記述したアルゴリズムをプログラムという。つまり、流れ図とプログラムともそれぞれの記述方法で表現したアルゴリズムのことである。つぎに、問題解決のアルゴリズムが決まれば、それに対応したプログラム（または、コードともいう）をどのように書くかを考える必要がある。どのような言語でもアルゴリズムが決まれば、それぞれの文法にしたがってプログラムを書き下すこと（以下、プログラミングという）ができる。ただし、実際のソフトウェア開発においては、アルゴリズムの複雑化や開発の効率化などの理由によりアルゴリズムの流れ図にしたがってプログラミングすることが多い。

一方、C言語をはじめとして種々のプログラミング教育が行われており、そのための支援システムの開発も行われている。特にプログラミングということが教育の目的となると、言語の文法に重きがいつてしまう傾向がある。この点が学習する場合に分かりにくくなるという問題が生じることとなる。また、どのようにコードを書くかだけでなく、問題をどう分析し、その解析手順をどのように決めるかということを含む必要がある。特定の言語のコードが書けることも必要であるが、一般に利用される言語は時代とともに変化してゆくので、どのような言語にも対応できるような問題の分析力も必要である。ただ、カリキュラム上アルゴリズムとプログラミングが独立した講義として行われることも多く、このような目的のためにそれぞれが機能することが難しいことがある。

これまで、プログラミングにおける学習、教育支援に関しては多くの提案がなされている。倉田ら^{(1),(2)}はC言語プログラミング授業のために、実行テストによるプログラム判定をサーバ側で行うシステムの開発を行っている。阿部⁽⁵⁾はプログラミング教育において、C言語を教育するのでなく、C言語を利用してプログラミングを教える場合のC言語が教えにくいものであるということから、初心者向けのBasicCというものを提案している。内田⁽³⁾はソースコードから作成した穴埋め式のドリルを自動作成し、Webを利用して解答が即座に自動採点されるシステムを開発している。新開ら⁽⁶⁾は一斉授業を基本として、学習者の理解度を把握しながらC言語プログラム教育を行うため、グループ活動、他者評価などを取り入れた授業展開を行うことを提案している。特に、プログラム教育の目的であるアルゴリズムを組み立て、プログラムを作成する能力を育成するため、プログラム作成プロセスを重視した指導を行い、その学習効果を明らかにしている。

またアルゴリズム教育においては、飯田ら⁽⁴⁾はアルゴリズム的な思考法をどのように教えるのかについての試みを行っている。すなわち、プログラミングのための前の段階としてのアルゴリズムでなく、より幅広く物事を分析し、表現できるようにする教育を目指している。

このように学習効果を向上させるという目的でアルゴリズム／プログラミング教育に関する多方面からの試みが行われているが、特に穴埋め型問題による学習方法において穴埋めの個数と箇所を固定した問題（すなわち、多者択一問題）を利用していることが多い。とりわけ実際の教育現場において種々の典型的な問題のアルゴリズムの流れ図とプログラムを用いてアルゴリズム／プログラミングに関する知識と技能を教えることが多い。そこで本研究では、一つ一つの典型問題に対してアルゴリズム／プログラミングにおいて学習すべき部分のすべてを穴埋めの候補（例

えば、20箇所）としてあらかじめ指定しておき、その候補の中からランダムに幾つかの穴埋め箇所を選んで穴埋め型問題を作成するという穴埋め型問題の自動作成の仕組みを提案し、この仕組みに基づいてアルゴリズム／プログラミング学習支援システムを開発した。

一般的には穴埋め箇所が多ければ多いほどその問題の困難度が高くなると考えられる。問題の難易度が穴埋め箇所の個数によって表現されると考えられるが、問題難易度の自由選択を可能にすることにより学習の効果を図ることが期待される。また、アルゴリズムの流れ図と実際のプログラミング技法の学習を同時に支援するための機能とか、よく利用されている各種言語プログラムの実行テストが可能になるような機能とか、既存資源（アルゴリズムの流れ図とプログラム）の再利用（例えば、新規プログラム作成の際に既存のプログラムをその部分コードとして利用、見本として参照）が可能になるような機能などが用意されている。これによりアルゴリズムをはじめ、プログラミング技法、各種プログラミング言語に対する理解を一層深めることが望まれる。このシステムではこれらの有効と思われる諸学習支援機能が含まれている。

本稿の残り部分は次のように構成されている。2. では、システムの開発目的をはじめ、システムを実現するための基本的な考え方を概説する。3. では、システムの基本構成を記述するとともにシステムの諸機能を説明する。さらに4. では、システムの活用法としてシステムの利用例（すなわち、効果的な学習方法）を取り上げて紹介する。5. は問題を自動的に生成して問題集に追加するための支援機能について論じるものである。最後に6. では、システム有効性の分析とか、さらにシステムに追加すべき機能などに関する今後の課題が挙げられる。

2. システム概要

ここでのシステムの開発目的は、コンピュータでの問題解決を行うための学習を支援するものであり、図1に示すように「アルゴリズム学習」部分、「プログラミング学習」部分、「プログラム作成学習」部分よりなっている。ここで、アルゴリズム学習とはその流れ図を用いて学習することとしている。

アルゴリズム学習とプログラミング学習とも穴埋め形式の問題が表示され、それを繰り返し解くことにより学習を行うようになっている。最初に難易度のレベルを指定すると、レベルに応じて穴埋めの個数、位置が変化するようになっている。すなわち、レベルを上げることにより穴埋めの個数が増加するようなシステムとなっている。問題のレベルを変えることで、穴埋めの個数、位置がランダムに変化して生成した問題を繰り返し解くことによりアルゴリズム、プログラムを理解しやすくできるようにしている。また、プログラミングについては、作成されたコードを実際にコンパイル、実行して結果が表示できるようにしている。

プログラムに関する問題は1問1ファイルよりなっており、ここに、問題文、複数言語のコードを含ませるようにしている。コードの穴埋めの空白として指定する部分を「」で囲むことにより、穴埋め型問題を自動的に生成している。したがって、問題ファイルの作成は既存の問題ファイル等から容易に作成することができる。アルゴリズム学習における流れ図の場合は、流れ図を

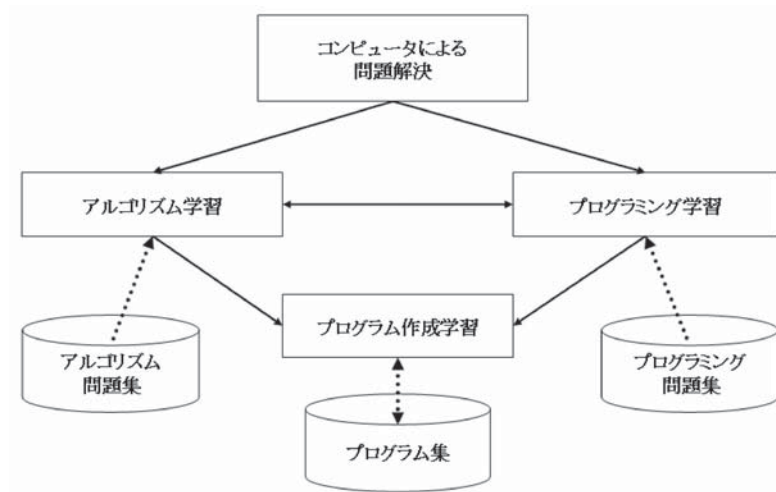


図1 システム概念図

分解し、空白とする部分の図をランダムに指定することにより問題を作成している。

穴埋め型問題を繰り返し実行させるようなシステムとしているのは、大学におけるプログラミング教育では言語の詳細な文法、例題のコードを理解させる教育は容易であるが、アルゴリズムを含めて与えられた問題をどのように分析し、それをコンピュータにどのように実行させるかを理解させることはかなり難しいことである。したがって、流れ図、プログラムなどを全てを書かせるようなことでなく、ある程度必要な部分のみを書くことにより流れ図、プログラムが完成できるようにするとか、他のコードから必要な部分のみを取り出し目的のプログラムを完成させるというようなことを理解することにより、アルゴリズム、プログラムということの全体像が理解できればよいのではないと思われる。

プログラム作成学習の部分については、他のコードを利用して目的の問題を解くためのコードを画面上で作成できるようにしている。

システムの開発環境にはMicrosoft Visual Studio 2008 Express Editionを用いている。システムの開発にはVisualC#2008, C言語の実行にはVisualC++2008を, Java, Pythonはjdk-6u13, Python-2.5.4をそれぞれ利用している。

システムの実行には前に示した環境であればすべての機能が利用できるが、Windows環境のみでも作成したプログラムの実行以外は可能である。

3. システム基本構成

3.1 初期画面

システムを起動すると図2のような画面が表示される。

図2 初期画面

この画面から問題を選択することにより問題文が表示され、アルゴリズムまたはコードを選択することにより問題画面が表示される。システムでは本体と、問題ファイルとは別の場所に保存するようにされており、問題ファイルは外部の記憶場所に置いている。したがって、最初にどの位置に問題ファイルが置かれているか指定する必要があるため、データドライブ名で表示されるメニューからドライブを選択する。

問題の選択は、「問題選択」ボタンをクリックして表示される図3で選択される。ここで、問題がプログラミング、アルゴリズム、プログラム作成という3つの部分に分類され、ツリー構造のメニューで表示される。メニューには具体的な問題のタイトルが表示され、そのタイトルをクリックすることにより問題が選択される。

問題を選択すると、問題文が表示される。問題文については出来る限り解析手順に対応した分析手順が含まれるようにしているが、文法的な説明は省かれている。アルゴリズムの場合には、問題と簡単な解説が表示されるようにしている。



図3 問題選択画面

アルゴリズムを選択した場合は、アルゴリズム、コード選択で「流れ図」しか選択できないが、プログラミングを選択した場合には、C、Java、Pythonの言語が選択できるようになっている。現時点ではC言語のコードは全ての問題に含まれているが、Java、Python言語のコードは部分的にのみ含まれている。プログラム作成の部分では、プログラミングの問題や流れ図を表示させた状態でプログラム作成ができるようにしている。

3.2 アルゴリズム学習画面

アルゴリズムの問題を選択し、「問題画面」ボタンをクリックすると、図4の画面が表示される。問題画面表示時点で基本的な部分以外はすべて空白となるようになっているので、左上の「最大13」のようにこの問題での最大の空白（円で囲まれた数字）数が表示される。空白部分の最大は20となっているが、最大の空白数は問題により異なっている。ここでメニューから空白数を指定する必要があるが、最低は2で最大までの範囲内の数を指定する。

この状態で「問題」ボタンをクリックすると、指定した数だけの空白を含む流れ図が表示される。空白部分の番号に対応する内容を、右端に表示されるメニューから選択する。

すべての解答を選択したならば、「選択終了」ボタンをクリックする。問題に含まれなかった項目を含めて選択された項目が流れ図の右側に表示される。選択した項目を変更する場合は、再度右端のメニューから項目を選択して、「選択終了」ボタンを押す。「結果表示」ボタンを押すと、

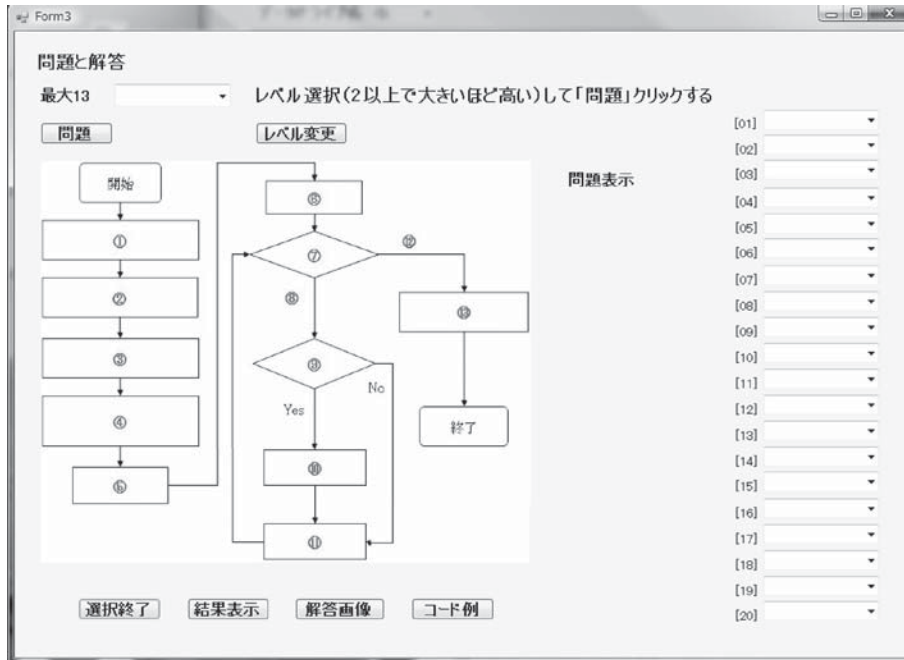


図4 流れ図の画面

図5 コード画面

選択された項目が正しいかどうかが表示される。間違っている場合は、間違っている位置が表示される。

「解答画像」ボタンをクリックすると解答の流れ図が表示される。「コード例」をクリックすると、ファイルに含まれているコード例が表示される（図5参照）。コードは問題により異なるが、VBコードが含まれる場合は、例えば、Excelを起動してマクロとして実行させるというようなことができる。

3.3 プログラミング学習画面

プログラミングの問題を選択し、アルゴリズム、コード選択で「cコード」を選択、「問題画面」ボタンを押して表示される画面が図6である。

問題と解答

最大9 レベル選択(2以上で大きいほど高い)して「問題」をクリックする

問題 レベル変更

```
#include <stdio.h>
int main(void)
{
    int i;
    printf(" i ");
    scanf(" %d", &i);
    {
        printf(" i %d\n", i);
    }
    {
        printf(" i %d\n", i);
    }
    return 0;
}
```

[01] [02] [03] [04] [05] [06] [07] [08] [09] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20]

選択終了 結果表示 実行

図6 コードの問題画面

問題作成時点で基本的な部分、問題のレベルで不要となる部分以外はすべて空白となるようになっているので、アルゴリズムと同様にその問題の最大の空白部分（ここでは最大9）が表示される。空白部分の最大は20であるので、ここで2から最大までの範囲内で空白部分を指定する。

レベルで5を選択し、「問題」ボタンをクリックすると、指定した空白を含むコードが図7（コード部分のみ）のように表示される。

空白部分には数字（[01]から[05]）が表示されているので、右端のメニューから空白の数字に対応する内容の項目を選択する。

すべてを選択したならば、「選択終了」ボタンをクリックする。選択された項目がコードの右側に表示される。選択した項目を変更する場合は、再度メニューから項目を選択して、「選択終


```
#include <stdio.h>
int main(void)
{
    int i[01];
    printf(" 整数を入力する:");
    scanf("%d%d",&i[02],&i[03]);
    if(i[04])
    {
        printf(" [%05] %d\n",
    }
    else
    {
        printf(" 入力されたのは奇数である %d\n");
    }
    return 0;
}
```

図7 問題画面のコード部分

了」ボタンを押す。「結果表示」ボタンを押すと、選択された項目が正しいかどうかが表示され、間違っている場合は再度修正し、「選択終了」ボタンで再度確認する。

コードが完成したならば、実際にコードを実行させることができる。「実行」ボタンを押すと、コードがコンパイルされ、実行結果が表示される。このように実行できるようにするためには各コードに対応する実行環境が必要である。ここでは、コマンドプロンプトの画面で結果が表示されるようにするため、Cコード、Javaコードはコンパイルで生成されたファイルを実行するバッチファイルを作成し、これにより結果が表示されるようにしている。Pythonコードについてもバッチファイルで実行結果を表示できるようにしている。

結果は図8に示されるようにコマンドプロンプトの画面に表示される。

問題における空白の数が同じでも、「問題」ボタンを再度クリックすることにより空白の位置を変更することができる。また、空白の数を変更する場合は「レベル変更」ボタンをクリックして、再度空白数を指定し、「問題」ボタンをクリックすることにより行うことができる。

どのようにこの画面を利用することにより学習効果が得られるかについては検討していないが、利用方法の自由度はかなりあるのではないと思われる。すなわち、最初からすべてを表示させてある程度内容を理解し、順に空白を多くして全体の構成が理解できるようにするとか、最初からできるだけ空白を多くし、修正を繰り返すことにより正しいコードを作り上げるようにするなどのことが考えられる。

3.4 プログラム作成学習画面

図2の初期画面の下の部分には、プログラム作成を支援するための画面がある。問題ファイルに含まれる問題は基本的なものであり、実際的な内容のものは少ない。これらはある程度理解することにより、より複雑な問題に対応するコードを作り上げる学習も必要である。

一般的にプログラミングの講義においては、プログラミング問題集のような基本的な問題を多く実行することにより文法的な内容説明が多くなり、実際への適用ができにくいというようなこ

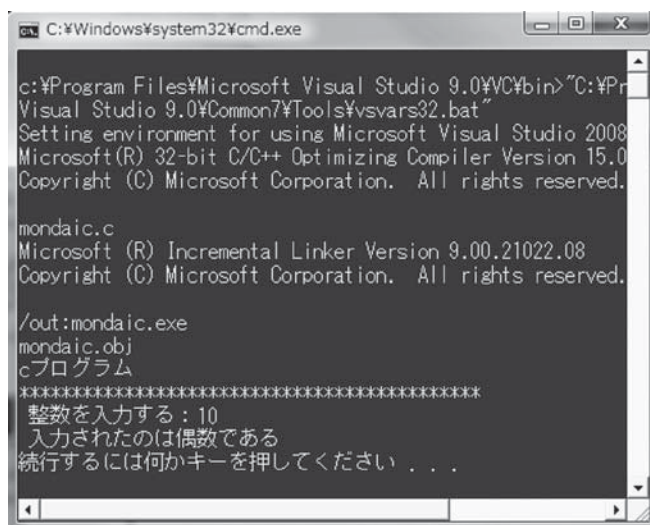


図8 実行画面

とが多い。場合によってはプログラミングの講義を受けた学生が実際の問題をプログラムすることができないというような状態が生ずる。すなわち、問題が分析できてもそれをプログラム言語のコードに結びつけることができないというようなことが起こる。

ここでは、基本的なコードを利用して、実際的なコードを作り上げることができるような機能を含ませている。プログラミング問題集の問題ファイルには、具体的なファイル名がつけられており、実際的な問題を分析してどのようなコードの部分が必要か理解できれば、それを組み合わせて必要なコードを作り上げることができる。

このプログラム作成の部分には問題文（具体的なプログラム内容）のみが含まれており、この問題を分析し、分析手順からどのようなコードが必要であるということがわかれば、「参照問題選択」ボタンをクリックして必要となるコード部分を表示させる。必要な部分は右側の枠内にコピーすることができる。複数のコードを組み合わせ、問題に対応するコードに作り直すことができる。

画面上でコードを編集することができ、必要なコードをこの画面上で完成させられる。このようにすることにより、最初からすべてのコードを書いていくことに比べれば比較的容易に必要なプログラムを完成させることができる。また、アルゴリズム問題集の流れ図部分には解答画像が含まれているので、これを表示させて対応するコードを編集画面で作り上げることもできる。

ここで作成されたプログラムは「プログラム実行」ボタンを押すことにより実行結果も確認することができる。ただし、コードに関するデバックはできるが、適正なコードであるかどうかについての判断まではできるようにはなっていない。

4. システムの利用例

このシステムの利用については、「アルゴリズム学習」部分、「プログラミング学習」部分についてそれぞれ問題を選択して、繰り返し問題を解くことに利用することもできるが、問題によっては連携させて利用することもできる。これによって、アルゴリズムの流れ図とプログラミングの書き換えに関する学習が実現できる。なお、アルゴリズム、プログラミングそれぞれ独自の学習が必要な部分もあるので、現時点ではすべてで連携することはできていないが問題によっては連携して利用できる。

4.1 アルゴリズムの流れ図とプログラミングの連携における利用例

アルゴリズムの講義で扱われる探索、並び替え等は、当然プログラミングの繰り返し、判断の部分に結びつくのであるが、これをそれぞれに学習する場合と、同時に学習できる場合ではその効果が異なると思われる。

例えば、アルゴリズムの最小値を求める問題を解くための流れ図の解答画面を図9のように表示できる。

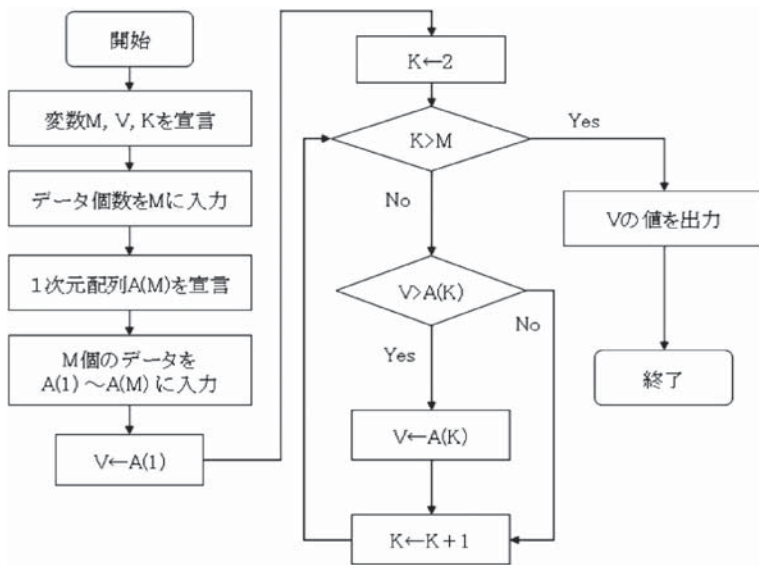


図9 解答画像（流れ図部分のみ）

この解答画像になるように繰り返し学習を行った後、これに完全に対応させたC言語のコードを次のように作成できる（「コード例」を押して表示される画面でコードを編集できる）。この場合最小値を求める配列の内容は指定されたものとしている。

```

#include
<stdio.h>
int main(void)
{
    int M=10,V,K;
    int A[]={33,28,23,41,5,3,21,18,26,16};
    V=A[0];
    K=1;
    modoru;;
    if(K>M) { printf("最小値 %d である \n",V); }
    else
    { if(V>A[K]) { V=A[K]; }
      K=K+1;
      goto modoru;
    }
    return 0;
}

```

プログラミングの問題集にも配列の問題の中に同様に配列に含まれる要素の最小値を求める問題があり、次のようなコードとなっている。

```

#include <stdio.h>
int main(void)
{
    int data2[]={33,28,23,41,5,3,21,18,26,16};
    int i,j,minv=100;
    for(i=0;i<10;++i)
    {
        if(data2[i]<minv)
        {
            minv=data2[i];
            j=i;
        }
    }
    printf("最小はdata2の %d 番目にある %d である \n",j+1,minv);
    return 0;
}

```

このような両方のコードの対応を利用することにより、プログラミングにおける繰り返し、判断というものがより理解しやすくなると思われる。

4.2 プログラム作成の例

例えば『名前 (xx) を入力し、現在時刻が12より小さい場合は「おはようございますxxさん」、そうでない場合は「こんにちはxxさん」と表示させよ』というような問題に対して次のような作成手順を用いることができる。

まず時間を取得する例題を探すと、問題選択画面の「プログラム」→「構造体」→「時刻」に関連する問題があるのでこのコードを表示させ、これをコード編集画面にコピーする。次に判断を行っている例題で同じようなものを探して表示させる（図10の左側に示す）。これの必要な部分をコード編集の画面にコピーする。

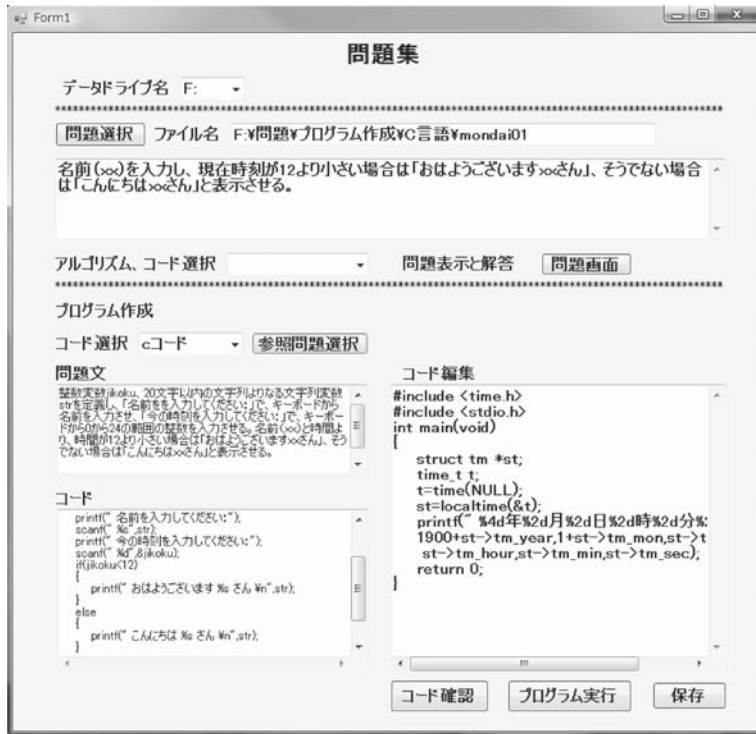


図 10 プログラム作成画面

コード編集画面上で両方のコードを編集して必要なコードを完成させた結果は次のように表示することができる。

```
#include <time.h>
#include <stdio.h>
int main(void)
{
    int jikoku;
    char str[20];
    struct tm *st;
    time_t t;
    t = time(NULL);
    st = localtime(&t);
    jikoku=st->tm_hour;
    printf("名前を入力してください：");
    scanf("%s", str);
```

```

if(jikoku<12)
{
    printf("おはようございます%sさん\n", str);
}
else
{
    printf("こんにちは%sさん\n", str);
}
return 0;
}

```

「コード確認」ボタンでコードを確認し、「プログラム実行」ボタンを押せば結果を確認することができる。完成したプログラムは「保存」ボタンでプログラム集に保存することができる。

5. 問題集作成支援について

この学習支援システムは問題集の学習によりアルゴリズム／プログラミングにおける技法を習得するためのものである。このため、学習効果が得られるような問題集の開発が必要となると思われる。この節では、問題集の作成支援について論じる。

システムに含ませるプログラムの問題については、これまでの講義で利用してきた例題を利用して作成した。しかし、各例題について空白とする部分を判断し「」で囲むことはかなり手間のかかる問題である。このため、プログラミングの例題についてはある程度自動的に問題が作成できるシステムを作成した。

この問題作成については、全体のシステムとは別に支援するシステムとして作成している。穴埋め型問題を基本としているので、プログラム言語のコードのどの部分を空白とするかは内容により異なってくるので、問題ごとに空白部分を指定する必要がある。しかし、問題表示で空白部分の数を指定できるようにしているので、基本的な部分とか初歩的な部分以外については、すべて空白に指定しても問題とはならない。

したがって、学習段階により異なるが、各時点での基礎的な部分以外を自動的に空白に指定し、内容により空白とする部分を手動で訂正できるような問題作成支援システムを作成した。このシステムでは問題作成のみでなく、問題ファイルへのデータの追加もできるようにしている。

5.1 プログラミング問題作成

1問について1個の問題ファイルを作成する。講義で利用している次の問題のCコードから問題ファイルを作成する。

```
#include <stdio.h>
int main(void)
{
    int price, kosuu;
    printf(" 価格（税込）を入力して下さい：");
    scanf("%d", &price);
    printf(" 個数を入力して下さい：");
    scanf("%d", &kosuu);
    printf(" 販売金額は %d 円です ￥n", price*kosuu);
    printf(" そのうち消費税分は %.0f 円です ￥n", 0.05*price*kosuu/1.05);
    return 0;
}
```

このコードに対応する問題文は次のようである。

***** 問題文 *****

整数変数 `price`, `kosuu` を定義し、「価格（税込）を入力して下さい：」と表示させ、キーボードから入力された値を変数 `price` に代入する。次に「個数を入力して下さい：」と表示させ、キーボードから入力された値を変数 `kosuu` に代入する。「販売金額 x x 円です」で `price*kosuu` の結果を表示させる。「そのうち消費税分は x x 円です」で `0.05*price*kosuu/1.05` の結果を表示させる。

これらを入力またはコピーし、処理して次のような問題ファイルを作る。

(問題文)
整数変数 `price`, `kosuu` を定義し、「価格（税込）を入力して下さい：」と表示させ、キーボードから入力された値を変数 `price` に代入する。次に「個数を入力して下さい：」と表示させ、キーボードから入力された値を変数 `kosuu` に代入する。「販売金額 x x 円です」で `price*kosuu` の結果を表示させる。「そのうち消費税分は x x 円です」で `0.05*price*kosuu/1.05` の結果を表示させる。

(cコード)
#include <stdio.h>
int main(void)
{
 「int」「price」, 「kosuu」;
 「printf」("「価格（税込）を入力して下さい：");
 「scanf」("「%d」", 「&price」);
 「printf」("「個数を入力して下さい：");
 「scanf」("「%d」", 「&kosuu」);
 「printf」("「販売金額は」「%d」「円です」￥n", 「price*kosuu」);
 「printf」("「そのうち消費税分は」「%.0f」「円です」￥n", 「0.05*price*kosuu/1.05」);
 return 0;
}

C言語のコードの基本的な部分、括弧、コンマ、コロンなどについては「」で囲まないが、それ以外についてはすべて「」で囲むようにしている。この段階では「int」「printf」「scanf」等を含めているが、次の学習段階となる問題ではこのようなものは含めないようにしている。

作成の画面は図11のようであり、作成手順は次のようである。

- 1) 問題ファイルを作成するため、問題ファイルを作成するドライブ名、フォルダ名を指定し、問題文を枠内に入力する。
- 2) 問題コードを入力、または他よりコピーすることにより、コードの枠内に表示させる。
- 3) 問題コードの穴埋めとする文字の前後に空白を挿入する。
- 4) 「プログラム確認」でプログラムコードが正しく実行されるかどうかを確認する。
- 5) 「コード表示」でコード編集枠内に「」が挿入されたコードが表示される。
- 6) 「ファイル書込み」で問題ファイルが作成される。



図 11 問題作成例

5.2 プログラミング問題への内容追加

問題ファイルを最初に作成した場合には1個のプログラム言語のコードが含まれているが、同一問題で複数言語のコードを含ませることができるようにもしている。ここでは、基本的にC言語のコードを含む問題を対象としているが、その他のJava言語とかPython言語のコードを追加する場合には次のようにする。

同一問題への追加であるので、ドライブ名、ファイル名、コード種類のみを指定して、追加部

分のコードをコードの枠内に入力するか、他よりコピーする。図11の「プログラム確認」で結果を確認して、「コード表示」、「コード追加」ボタンで最初の言語のコードの下に追加の言語のコードが追加される。

5.3 アルゴリズム問題の作成

本システムでは流れ図を用いてそのアルゴリズムを学習する。アルゴリズム問題の作成はその流れ図問題の作成になる。流れ図の問題では解答画像、穴埋め部分を含む画像を作成する。穴埋めを含む画像では流れ図の項目で穴埋めとする部分を最大20までの円で囲まれる数字に置き換えたものとしている。それぞれの全体画像を7×3の均等の部分に分解し、丸数字部分が範囲内（すなわち、分割した1枚1枚の画像内）に収まるようにした。画像分割の処理のためには画像分割ソフト「分割結合.exe」⁽⁷⁾を利用している。

6. おわりに

アルゴリズム／プログラミングに関する学習支援システムの開発を行った。「アルゴリズム学習」部分、「プログラミング学習」部分を独立して利用できるが、同時に利用することによりそれぞれの内容をより理解できるようにすることも試みた。今後は実際に利用した結果によりその効果を評価し、よりシステムの改良を行っていくつもりである。また、学習した記録を保存し、それを教育等に利用できるようにすることも必要であり、このような機能もシステムに含ませるようにする予定である。

参考文献

- (1) 倉田英和, 富永浩之, 林敏浩, 垂水浩幸 実行テストによるプログラム判定を用いた初級Cプログラミング演習支援と授業支援 情報処理学会研究報告, CE91, pp. 11-18, 2007
- (2) 富永浩之, 倉田英和, 林敏浩, 安藤一秋, 垂水浩幸 コンテスト形式による初級Cプログラミングの演習支援 情報処理学会研究報告, CE94, pp. 49-56, 2008
- (3) 内田保雄 初級プログラミング技法学習のための自動作問システム 情報処理学会研究報告, CE92, pp. 109-113, 2007
- (4) 飯田周作, 飯田千代, 清藤武暢, 佐藤創 アルゴリズム的思考法の教育 情報処理学会研究報告, CE93, pp. 57-64, 2008
- (5) 阿部圭一 C言語プログラミング教育についての省察 情報処理学会研究報告, CE98, pp. 205-212, 2009
- (6) 新開純子, 宮地功 プロセスの重視と評価活動を取り入れたCプログラミング教育の効果 教育システム情報学会誌, Vol. 26, No. 1, pp. 16-28, 2009
- (7) ダウンロードサイト <http://www.vector.co.jp/soft/dl/winnt/art/se346589.html>