

# プログラムの書式チェックを行う コンピュータシステムの提案

伊藤雅人\*1・杉山宏太\*1・北英彦\*1

Email: itou2@ce.elec.mie-u.ac.jp

\*1: 三重大学大学院電気電子工学専攻

◎Key Words プログラミング教育, プログラミングスタイル, 演習システム, 自己学習

## 1. はじめに

大学教育の一環として、プログラミング演習が多く  
の大学で行われている。これらのプログラミング演習  
において学生が身につけるべきことは大きく分けて次  
の3つである。一つ目に、プログラミング言語の各機  
能を適切に使用できるようになること。二つ目に、課  
題で指示されたプログラムを作成できるようになるこ  
と。三つ目に読みやすいプログラムを書けるように  
なることである。

この中で、三つ目の読みやすいプログラムの書き方  
については、現状、学生は十分な指導を受けていない。  
指導されていない理由としては、読みやすいプログラ  
ムの書き方について指導するには、講師が学生の書い  
たプログラムを添削する必要があり、講師の負担が大  
きいことが挙げられる。しかし、読みやすいプログラ  
ムを書くことは、他人がプログラムを読んで理解する  
時間を節約し、保守や移植も行いやすくなり、プログ  
ラムの価値を高めることにつながる。それゆえ、字下  
げや改行、また変数名やコメントなど、正しい知識さ  
え習得すればすぐにでも実践できることなどは、しっ  
かりと指導を行うべきである。

本研究では、学生が読みやすいプログラムを書ける  
よう支援する演習システムの開発を目標としている。  
そこでまずは、プログラムの読みやすさに最も大きく  
影響を与える字下げについて、学生に指導を行うシ  
ステムの開発を行った。

字下げはルールが簡単であり、プログラミング初心  
者でも実践できる。しかし、現状では、字下げにつ  
いてしっかりと書かれた教材が少なく、また指導を行  
う場合には添削指導が望ましく、講師の負担が大き  
いため現状では指導が十分に行われていない。この  
ため、字下げを正しく行えない学生が約半分以上  
いるのが現状である。本システムにより、学生が  
字下げを正しく行えるようになることが期待され  
る。

## 2. プログラミングスタイル

### 2.1 表面上の改善

本章では、読みやすいプログラムを学生が書ける  
ようになるためには、何を教えれば良いかを検討す  
る。読みやすいプログラムとは、言い換えれば、他  
人が最短時間で理解できるプログラムであり、この  
ようなプログラムを書くためには様々な方法がある。  
例えば、最も基本的なものとして、ソースコードの  
表面上の改

善が挙げられる。これは、その名の通り、プログラ  
ムの書式を整え、ソースコードを見やすくすること  
で、プログラムの読みやすさを向上させる方法であ  
る。具体的には、字下げや改行、変数名の付け方、  
コメントの付与などが表面上の改善にあたる。

ソースコードの表面上の改善は、プログラミング初  
心者でも知識さえあれば実践でき、プログラムの可  
読性を大きく向上させる。そこで、本研究では、ま  
ず字下げについて学生に指導を行い、学生が正しく  
字下げを行えるよう支援するシステムの開発を目標  
とする。

### 2.2 字下げ

字下げとはプログラムを右にずらして見やすくす  
ることで、具体的には行の先頭に空白を挿入して  
プログラムを右にずらすことである。基本的には、  
ブロックの中({}で囲まれた範囲)を1段字下げす  
る。1回の字下げに最もよく使われるスペースの  
数は4つであり、本システムでも1回の字下げに  
用いるスペースの数は4つとする。

```
001 /*
002 読み込んだ整数値の符号を判定
003 */
004
005 #include <stdio.h>
006
007 int main(void)
008 {
009     int no;
010
011     printf(" 整数を入力してください: ");
012     scanf("%d", &no);
013
014     if(no == 0){
015         puts(" その数は0です。");
016     }
017     else if(no > 0){
018         puts(" その数は正です。");
019     }
020     else{
021         puts(" その数は負です。");
022     }
023
024     return(0);
025 }
```

図1 字下げの位置

例えば、図1のようなプログラムの場合、8行目で、開き括弧がある、これにより次の行から1段字下げがされている。この字下げは、4行目の開き括弧と対応している25行目の閉じ括弧まで続く。次に14行目にif文がある、よってif文の有効範囲である16行目までは、さらに1段字下げがされている。同様に18、21行目も制御文の有効範囲であるために、さらに、1段字下げがされている。

### 2.3 字下げの指導状況

前述の通り、字下げのルールは簡単であり、シンプルな考えた方である。しかし、現状では学生に対して十分な指導が行われていない。その理由として以下の2つがあげられる。

一つ目に、プログラムの読みやすさは、プログラムの文法や構文などに比べて、あまり重要視されていないことが挙げられる。このことはプログラミング言語の教科書からも明白である。例えば、読みやすいプログラムを書くための方法として、最も基本となる字下げや改行といった知識さえあれば実践できることさえもしっかりと教えている教科書は稀である<sup>(46)</sup>。

また、二つ目の理由として、読みやすいプログラムの書き方を指導するには、講師の負担が大き過ぎることが挙げられる。というのも、プログラムの読みやすさについて指導を行うには、講師が学生の書いたプログラムを見て添削指導を行う必要がある。しかし、プログラミング演習の講義では、講師1人に対して学生の数が40人程度いるのが一般的であるため講師の負担が大き過ぎる。

### 2.4 学生の字下げ状況

このことを表すように、学生の約半分以上は字下げが適切に行えていない。学生の字下げ状況を示めた表を以下に示す。また、字下げが適切に行われていないプログラムでは、図2に示すようなプログラムが多く見受けられた。

表1 調査結果

	字下げの状況		合計
	適切	不適切	
件数	151	182	333
割合	45%	55%	

```

001 #include <stdio.h>
002
003 int main(void)
004 {
005     int i;
006     int vc[5];
007     for(i=0;i<5;i++)
008         vc[i]=5-i;
009     for(i=0;i<5;i++)
010     printf("vc[%d]=%d\n",i,vc[i]);
011
012     return(0);
013 }

```

図2 学生のプログラムの例

さらに、C言語を用いたソフトウェアの開発経験が1年以上あるプログラミング中級者15名に対して、以下の2つの質問を行った。

- 字下げを行っていますか。
- 字下げのルールを説明できますか。

その結果、ほぼ全員が字下げを行っているものの、9人が字下げのルールを説明できなかった。これらの学生に話を聞いたところ、彼らの多くは、字下げをエディタのオートインデント機能に頼っており特に意識していないことが分かった。

### 2.5 演習システムの提案

今では、高機能なエディタが数多く存在し、字下げの知識がなくとも、これらのエディタを使えば自動で字下げが行える。しかし、高機能なエディタがなければ、字下げができない状態は好ましくない、また、字下げの正しい知識がなければ、プログラムの構造を把握するのに手間がかかってしまいます。

以上から、字下げについて指導を行う必要があると言える。しかし、講師が指導を行うには負荷が大き過ぎるため現実的ではないことは、既に述べた。そこで、次章で講師に代わり、字下げについて指導をおこなう演習システムを提案する。

## 3. プログラミングスタイル指導システム

前章で述べた字下げにおける指導状況を受け、本章では講師の力を必要としない字下げ指導システムを提案する。

### 3.1 PROPEL の利用

本研究室では、ブラウザ上で稼働するプログラミング演習支援システム（以降PROPELと呼ぶ）を開発しており、このシステムを利用することで学生が字下げを習得できるような環境を構築することにする。

PROPELは学生に対してプログラムの作成やデバッグ、また実行といったC言語での開発環境を提供しており、これにより学生はブラウザ上でプログラムの作成から提出までを一貫して行うことができる。

PROPELに字下げをチェックする機能を実装することにより、学生がプログラムを提出したそのときに、字下げについてチェックを行い、学生に結果を表示することができるシステムの開発を行う。図3に本システムの構成図を示す。

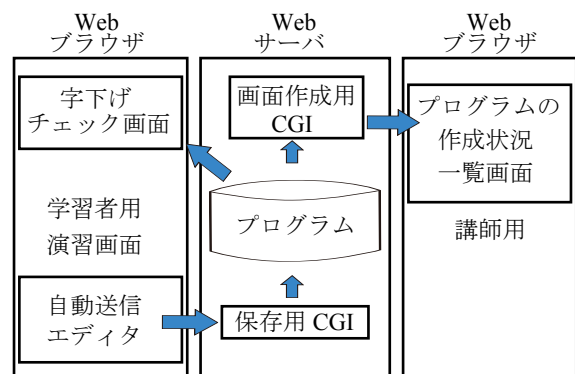


図3 システム構成図

### 3.2 提案する字下げの学習方法

本研究では、以下のような流れで字下げの仕方をも身につける方法を提案する。

- ① 学生が字下げの教材を読む
  - ② 学生が字下げの例題を解く。
  - ③ 学生が演習を行う。
  - ④ システムが字下げのチェック結果を学生に示す。
- これらの流れについて、図を用いて詳しく説明する。

#### (1) 学生が字下げの教材を読む

まず、教材を学生によんでもらい、字下げの基礎知識について理解してもらう。図4に教材の一部を示す。

##### □ 字下げとは

行の先頭に空白（スペース）を挿入することです。字下げを行うことによりプログラムが読みやすくなります。

##### 字下げをする位置

ブロックの中（{ } で囲まれた範囲）で字下げを行います。

```
int main(void)
{
    int i = 0;
    :
}
```

##### 字下げに用いるスペースの数

字下げを行う際のスペースの数は4つを基準単位としてしてください。

```
int main(void)
{ /* ここから一段字下げ
   スペース4つを挿入 */
    int num;

    printf("整数を入力してください");
    scanf("%d", &num);

    if(num % 2){ /* ここから二段字下げ
                 スペース8つを挿入 */
        puts("入力した値は奇数です。");
    }else{
        puts("入力した値は偶数です。");
    }

    return(0);
}
```

図4 字下げの教材

#### (2) 学生が字下げの問題を解く

学生教材を読んでもらった後に字下げについて修正を行う問題を解いてもらう。問題を図5に示す。

□ 以下のプログラムにおいて適切に字下げを行いなさい。

保存	...エディタで書いたプログラムを保存します。
コンパイル	...保存したプログラムをコンパイルします。
実行	...コンパイルしたプログラムを実行します。
字下げチェック	...提出したプログラムの字下げをチェックします。
提出	...書いたプログラムを提出します。提出済みの場合は上書きされます。
ギブアップ	...この時間でのプログラム作成を諦めて提出します。

```
/*
 * 読み込んだ整数値は奇数であるか偶数であるか
 */
#include <stdio.h>
int main(void)
{
    int no;

    printf("整数を入力してください");
    scanf("%d", &no);

    if ( no % 2 ) {
        puts("その数は奇数です。");
    } else {
        puts("その数は偶数です。");
    }

    return(0);
}
```

図5 例題画面

#### (3) 学生が演習を行う

一般にプログラミング演習の講義では、講師は前半でプログラミング言語の文法などについて説明し、学生は後半で与えられた演習問題に取り組む。

プログラミング演習システム PROPEL に字下げチェック機能を実装することにより、学生が演習を通して正しい字下げを身につけることができるようにした。図6に演習画面を示す。

□教科書p.217 演習9-4

文字列strの中に、文字cが含まれている個数（含まれていなければ0とする）を返す関数  
int str\_chnum(const char str[], int c) (\*/\*...\*/)を作成せよ。

問：文字列strの中に、2文字の並び（cdが連続している）が含まれている個数（含まれていなければ0とする）を返す関数 int str\_ch2num(const char str[], int c, intd)を作成し、実行結果を表示しなさい。

保存	...エディタで書いたプログラムを保存します。
コンパイル	...保存したプログラムをコンパイルします。
実行	...コンパイルしたプログラムを実行します。
字下げチェック	...提出したプログラムの字下げをチェックします。
提出	...書いたプログラムを提出します。提出済みの場合は上書きされます。
ギブアップ	...この時間でのプログラム作成を諦めて提出します。

図6 演習画面

#### (4) システムが字下げのチェック結果を学生に示す。

学生が作成したプログラムをシステムがサーバに送り、字下げチェックを行う。そして、チェックした結果を学生に示す。

図7に字下げチェックの流れを示す。

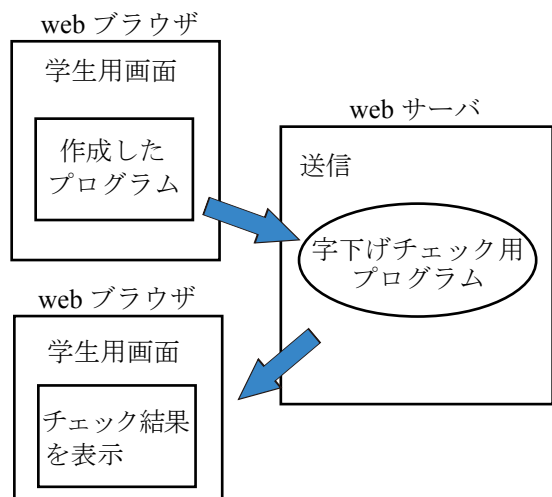


図7 字下げチェックの流れ

以下で、字下げチェックに用いたアルゴリズムについて述べる。あるプログラムの行で字下げを行うべき回数をレベルと呼ぶことにする。また、1回の字下げに用いたスペースの量を深さと呼ぶ。本演習システムでは、字下げ1回に用いるスペースの量は4つと定めて

いるため、レベル0ではスペース0個、レベル1ではスペース4つ、レベル2ではスペース8個を挿入するのが正しい字下げとなる。

例えば、図8の左側に示すようなプログラムのレベルと深さを調べると図8の右側に示すようになる。この場合、7,10,12行目がレベルに対して、字下げ量が適切でないため、字下げが適切でない行として判断する。

	深さ	レベル
001 #include <stdio.h>	0	0
002		
003 int main(void)	0	0
004 {	0	0
005     int num;	4	1
006     printf(“入力”);	4	1
007     scanf(“%d”,&num);	0	1
008		
009     if(num/2==0)	4	1
010     printf(“偶数”);	4	2
011     else	4	1
012     printf(“奇数”);	4	2
013		
014     return(0);	4	1
015 }	0	0

図8 チェック対象のプログラム

学生への指導画面を図9に示す。まず、字下げが適切に行えているかどうかを示し、字下げが適切に行われていない場合には、字下げが適切でない行を示す。また、学生が提出したプログラムで適切でない行の色を変えて表示することにする。

字下げチェック結果

結果：字下げが適切でない箇所があります。  
該当する行：13 15 17 19 行目  
適切に修正してください。

---

提出物

```

001 #include <stdio.h>
002
003 #define G00 0
004 #define CHOKI 1
005 #define PA 2
006
007 int main(void)
008 {
009     int hand;
010     puts(“数字を入力してください”);
011     scanf(“%d”,&hand);
012     if(hand == G00)
013         puts(“じゃんけんの手 グー”);
014     else if(hand == CHOKI)
015         puts(“じゃんけんの手 チョキ”);
016     else if(hand == PA)
017         puts(“じゃんけんの手 パー”);
018     else
019         puts(“無効な数字”);
020     return(0);
021 }

```

図9 指導画面

#### 4. 実験の予定

本システムを実際にプログラミング演習で使用してもらい、システムの有効性を検証する必要がある。そこで以下の条件でシステムの有効性を検証することを予定している。

講義：2013年度前期 プログラミング演習Ⅱ  
対象：電気電子工学科2年生 80名

#### 5. 今後の課題

本論文では、字下げを学生に指導するシステムの提案を行った。今後は、本システムを用いて実験を行いシステムの有効性を検証する必要がある。

また、可読性の高いプログラムを書けるようになるには、字下げの他にも変数名や改行、またコメントなど学ぶべきことは多い。今後はこれらについても、学生が学ぶことができる環境を提案していく必要がある。

#### 6. まとめ

プログラミング演習において、現状、可読性の高いプログラムの書き方については十分な指導が行われていない。指導が行われていない理由としては講師の負担が大きいことをあげた。

こうした現状を受け、本研究では、学生が可読性の高いプログラムを書けるよう支援する演習システムを開発している。本論文では、プログラムの可読性に大きく影響を与える字下げについて、学生に指導を行うシステムを提案した。

#### 参考文献

- (1) Dustin Boswell, Trevor Foucher：“リーダブルコード”，O'REILLY (2012) .
- (2) Steve McConnell：“CODE COMPLETE 下”，pp.323-428, Microsoft Press (2012) .
- (3) 縣 俊貴：“良いコードを書く技術”，技術評論社 (2011) .
- (4) 柴田望洋：“明解C言語”，SoftBank Creative (2009) .
- (5) 柴田亡洋：“解きながら学ぶC言語”，SoftBank Creative (2005)
- (6) 林晴比古：“改訂新C言語入門”，SoftBank Creative (2002) .
- (7) 福岡知的クラスター（第一期）組み込みソフト開発プロジェクト：“組み込み現場の「C」プログラミング 標準コーディングガイドライン”，技術評論社 (2008) .
- (8) 間座秀幸：“プログラミングスタイルを指導するための初心者向け演習システムに関する研究”，平成17年度修士論文 .
- (9) 伊富晶幸, 小島佑介, 北英彦：“プログラムの作成状況を把握する機能を持つプログラミング演習システム”，2010PCカンファレンス (2010) .