

図解を入力とした設計書作成とプログラミングによる 文書作成教育

藤田 悠*1

*1 長野工業高等専門学校

Technical Writing Education by Programing and Documentation using Diagram as Documentation Input

Yutaka FUJITA*1

*1 National Institute of Technology, Nagano College

システム開発に携わる学生を対象に、文書作成能力を育成するための教材を作成し、実践してきた。対象者を2つのグループに分け、異なる演習課題で設計書を作成させ、異なる設計書を作成したペアで設計書を交換してプログラミングさせ、ディスカッションさせる教材である。これにより、ソフトウェア開発における文書の役割や、文書作成能力の自覚などに関する気づきを与えることができた。従来、教材の演習課題では、設計書を作成させるための入力情報として、ソースコードを用いていた。しかし、ソースコードから設計書を作成する手順では、通常ソフトウェア開発プロセスと逆方向であるので、プロセスを把握している場合には特に、違和感を与える教材になっていた。本稿では、入力情報を図解として与え、図解のアルゴリズムについて設計書を作成する教材とした。改善を適用した教材を作成し、その教材を用いて教育を実施した。その結果、目的に合致する教育効果が確認できたことを報告する。

キーワード: テクニカルライティング, ドキュメンテーション, プログラミング, ソフトウェア開発文書

1. はじめに

ソフトウェア開発やシステム開発に携わる技術者にとって、ソフトウェアの実装技術だけでなく、文書作成技術も必要である。

開発プロセスにしたがって進められるプロジェクトなどでは、要求定義、アーキテクチャ設計、詳細設計などの工程の成果物として開発文書を作成して、次の工程に開発をつなげる。この開発文書による情報伝達によって開発が行われるため、文書の品質は最終的な成果物であるソフトウェアの品質に影響を与える。

開発文書は、UMLやDFDなどといった図による方法や、自然言語によって書かれる。工程によって、図や表を多く用いる場面が多いこともあるが、上流工程では自然言語を用いて表現することが多い。また、開発にかかる時間の内、実装工程以外の部分に多くの開

発時間が費やされており、それらの工程で開発文書を作成する能力が必要である。しかし、開発現場では、ライティングの教育機会は限られている。そこで、学校教育において、基礎能力としてライティング力を育成することが求められる。

このような、開発現場からの要望があるが、学校教育においては、実装に必要とするプログラミングを中心に学習しており、ライティングを学習する時間は限られている。本校では、ライティングを専門で学習する科目はなく、PBL(Project Based Learning)の形式をとって行われる総合的な活動の一部として開発文書を作成するが、そのほかに設計書などの開発文書を作成する機会はない。

ほかに文書を作成する機会として、レポートを作成する実験科目はあるが、ライティングの指導は教員によるアドバイスなどによって行われており、文書作成

に特化した時間は設けられていない。

そこで、ソフトウェア開発者に文書作成能力が必要であることに学生自身が気づき、学生それぞれが高めべき文書能力を認識することを目的とした教材を検討した。この教材で学生が自覚したことを、以降で取り組むレポート作成をはじめとした文書作成の場面で育成していくことにつなげたいと考えた。

作成した教材では、ペアで異なる演習課題を与え、その演習課題の設計書を作成させる。作成した設計書をペアで交換して、設計書をもとにC言語でプログラミングさせる。プログラミングした結果をもとにディスカッションさせる。

従来法では、演習課題の与え方が課題であった。従来、ソースコードを入力情報として与えて設計書を作成させた。この入力情報には利点もあるが、開発プロセスの方向とは逆にあたる。そこで、アルゴリズムを表現する図解を与えて、設計書作成をさせた。

本稿では、設計書作成の入力情報をソースコードから図解に変えることによって、開発プロセスに違和のない演習課題の提供方法に改善する。改善した教材にて教育を実施し、図解による入力情報にてソフトウェア作成に十分な情報が設計書に表わされているか確認する。さらに、従来法と変わることなく、学生による気づきが得られることを確認する。

2. 教育方法

従来実施してきた教育方法および、教材について述べる。

2.1 教育方法の概要

これまで、ソフトウェア開発技術者にとって文書教育の必要性と、学生自身が文書作成能力に気づくための教育として、次のステップからなる教育を検討し、実施してきた。

- (1) ソフトウェア開発における文書作成の位置づけに関する座学
 - (2) ペアで異なる演習課題について設計書作成
 - (3) ペアで設計書を交換してプログラミング
 - (4) 作成プログラムや設計書を基にペアでディスカッション
- (1)では、ソフトウェア開発におけるプロセスの知識

を踏まえて、要求分析や設計などの工程の成果物として文書を作成することや、作成した文書を次の工程の入力として開発をすすめることを説明した。文書が存在することにより、文書の品質が、ソフトウェアの品質やコスト、納期に影響を与えることを認識させた。

(2)では、演習課題として演習課題1と演習課題2の2題用意した。受講する学生を半分に分け、半分の学生には演習課題1、もう半分の学生には演習課題2に関する設計書を作成させた。

(3)では、演習課題1に取り組んだ学生と、演習課題2に取り組んだ学生をペアにした。作成した設計書をペアの相手と交換させ、受け取った設計書をもとに、C言語にてプログラミングするように指示した。

(4)では、設計書を交換したペアで、それぞれがプログラミングしたプログラムを見せあわせた。さらに、設計書作成、プログラミング、作成プログラムの確認を通して気づいた点をペアでディスカッションさせた。

2.2 教育方法を構成する教材

教育の各過程で用いた教材について詳しく説明する。

2.2.1 演習課題

2題の演習課題は次のような内容である。

【演習課題1】指定された金額でポテトチップスとせんべいの2種類の菓子を買うときに、おつりを最も少なくする菓子の組み合わせ決定する。

【演習課題2】2科目の成績点数リストに、総合点による順位をつける。ただし、同一点数の場合には、ある1科目の順位に従う。

この演習課題を学生に提示する方法として、従来法では、印刷したソースコードを渡し、このソースコードのプログラムを相手に作成してもらえるような設計書を作成するように指示した。ソースコードを設計書の入力にすることで、実装すべきプログラムのアルゴリズムを理解したうえで、そのアルゴリズムどのように文書化して伝えればよいか考えられると想定した。

2.2.2 ディスカッションシート

ディスカッションをするときには、ディスカッションしたことを記録するために、ディスカッションシートを配布して、記入させた。

ディスカッションのステップまでに、設計者として

表1 ディスカッションシートフォーマット

立場	対象	内容
プログラマ	プログラム	(プログラマの立場でプログラムに対する議論を記入する.)
	設計書	(プログラマの立場で設計書に対する議論を記入する.)
設計者	プログラム	(設計者の立場でプログラムに対する議論を記入する.)
	設計書	(設計者の立場で設計書に対する議論を記入する.)
その他		(上記以外の議論を記入する)

設計書を作成し、プログラマとしてプログラミングしたので、これら2種類の役割を体験している。そのため、得られた気づきが、いずれの立場のときに生じたものであるのかを踏まえてほしいと考えた。また、各作業での成果物として、設計書とプログラムがある。そのため、気づきがいずれの対象物に関するものであるのか区別してほしいと考えた。

そこで、立場と対象をわけて議論をメモするためのディスカッションシートを配布した。ディスカッションシートのフォーマットを表1に示す。

このディスカッションシートの「内容」以下の欄に、立場と対象を区別して記入するように指示した。

2.3 教育結果

従来法での教育により、目標に設定していた(1)自分の文書を客観的に見ること、(2)文書作成力が必要であることに気づくこと、(3)設計を考えること、に関する気づきを与えることができた⁽¹⁾。

実施したアンケート結果で、受講者は設計書作成とディスカッションが役立ったと回答していた。設計書を作成することで、実際に自分の文書を作成し、それが、プログラミングを通して、どのように伝達できたを相手から実際に得られる意見を交わすことで、気づきが得られることにつながったと考えられる。

2.4 問題点

従来法における問題点は、設計書を作成させるための入力情報が、ソースコードであることである。

入力がソースコードであることによるメリットも 2

点考えられる。メリットの1点目は成果物との比較が容易であることである。ソースコードを入力とし、設計書を媒介して相手がプログラミングする。その出力としてソースコードが作成されるので、入力と出力を比較して、設計書がどのようにソースコードに影響を与えるかをとらえやすいと考えられる。メリットの2点目として、アルゴリズムを伝達するために便利であることがある。設計書に書いてほしい内容を設計者に伝えるために、演習課題の内容のみをあたえると、その課題を解く方法すなわちアルゴリズムを考案しなければならない。そのため、文書作成よりも、発想に時間がとられてしまい、演習の目的が変化する⁽²⁾。ソースコードを用いることで、文書化してほしいアルゴリズムを理解して、それを「どのように文書で表現すればよいか」を考えることに絞ることができる。

他方、入力がソースコードであることの問題点は、実際の開発プロセスに適合しない入力情報であることである。ソフトウェアの開発プロセスでは、要求定義、アーキテクチャ設計、詳細設計、実装の順番で進める。実装よりも前の段階では、ソースコードはできておらず、自然言語や図式などを用いた内容が示された文書が各工程の成果物である。ソースコードから設計書を作成する行為は、リバースエンジニアリングに該当することになり、開発プロセスの順番に逆行することになる。そのため特に、本教材を社会人の技術者に適用する場合などでは、強い違和感を与えることになる⁽⁴⁾。このような違和感を与えないために、入力情報を適切な形式に変えることが必要である。

3. 入力情報の改善

従来法の問題点に対して、設計書を作成するための入力情報として、ソースコードではなく、アルゴリズムを表わした図解にした。

3.1 概要

入力情報として設計者に伝えたいことは、設計書を受け取った相手に作成してもらいたいプログラムのアルゴリズムである。これまでは、そのアルゴリズムをC言語で表わしたソースコードを入力としていた。この代替として、アルゴリズムを図や表などによる図解を用いることとした。

	ポテトチップス	→	せんべい	→	おつり
(1)	0個 代金 0円		残金 8個 代金 400円		残金 20円
↓					
(2)	1個 代金 130円		残金 5個 代金 250円		残金 40円
↓					
(3)	2個 代金 260円		残金 3個 代金 150円		残金 10円
↓					
(4)	3個 代金 390円		残金 0個 代金 0円		残金 30円

図 1 演習課題 1 の図解

演習課題の内容は、主に繰り返しと条件判断による処理である。そこで、図解では、処理手順での繰り返しによって値がどのように変化していくのか、具体的な数値を用いた例を示した。文書化の際には、図解で例示している内容を一般化し、設計書としてアルゴリズムを文書で表現することを期待した。

3.2 演習問題

今回の演習課題として用いた 2 題の図解を示す。

3.2.1 演習問題 1

演習課題 1 については、入力されたお金で、おつりが最も少なくなるように、2 種類の菓子を購入する個数を決めるものである。

アルゴリズムとしては、購入し得るパターンをすべて導出して、その中で最善のパターンを見つけ出す。そのアルゴリズムの図解を図 1 に示す。導出するパターンを少なくするために、価格が高い菓子の個数を増やしていき、その残金で価格が低い菓子を出来得る限り購入し、その残金をおつりとする手順を繰り返す。

配布した資料には、第 1 章に「目的」として、設計書を作成するプログラムで解決すべき内容を文章にて簡潔に述べている。この文章では、解決すべき内容を導出するアルゴリズムには言及していない。第 2 章には、2 種類の菓子であるポテトチップスとアメの価格を定義した。第 3 章に、図解を示した。

3.2.2 演習課題 2

演習課題 2 については、2 科目の点数が与えられた 5 名の成績について、順位付けするものである。ただし、2 科目の合計が同じ時には国語の点数が高い方の順位を上にとすることとする。

アルゴリズムとしては、初期状態にて全員の順位を

初期状態					
科目	1 人目	2 人目	3 人目	4 人目	5 人目
国語	68	81	83	90	65
数学	92	76	51	85	69
合計	160	157	134	175	134
順位	1	1	1	1	1

1 人目を基準に、順位を下げる。

科目	1 人目	2 人目	3 人目	4 人目	5 人目
国語	68	81	83	90	65
数学	92	76	51	85	69
合計	(基準) 160	(小) 157	(小) 134	(大) 175	(小) 134
順位	1	1+1	1+1	1	1+1

2 人目を基準に、順位を下げる。

科目	1 人目	2 人目	3 人目	4 人目	5 人目
国語	68	81	83	90	65
数学	92	76	51	85	69
合計	(大) 160	(基準) 157	(小) 134	(大) 175	(小) 134
順位	1	2	2+1	1	2+1

3 人目を基準に、順位を下げる。

科目	1 人目	2 人目	3 人目	4 人目	5 人目
国語	68	81	(基準) 83	90	(小) 65
数学	92	76	51	85	69
合計	(大) 160	(大) 157	(基準) 134	(大) 175	(同じ) 134
順位	1	2	3	1	3+1

4 人目を基準に、順位を下げる。

科目	1 人目	2 人目	3 人目	4 人目	5 人目
国語	68	81	83	90	65
数学	92	76	51	85	69
合計	(小) 160	(小) 157	(小) 134	(基準) 175	(小) 134
順位	1+1	2+1	3+1	1	4+1

5 人目を基準に、順位を下げる。

科目	1 人目	2 人目	3 人目	4 人目	5 人目
国語	68	81	83	90	65
数学	92	76	51	85	69
合計	(大) 160	(大) 157	(大) 134	(大) 175	(基準) 134
順位	2	3	4	1	5

図 2 演習課題 2 の図解

1 にしておき、基準とする成績と他の成績を比較して、比較対象の点数が低ければ比較対象の順位を 1 位分下げる。この処理を、基準を変えて、全員分が基準になるまで繰り返すと、順位が決定する。点数を比較するとき、総合点数で比較して同じ時には、比較相手の国語の点数が低ければ、相手の順位を下げて、国語の点数も同じであれば、順位操作は行わない。このアルゴリズムを表した図解を図 2 に示す。

配布した資料には、第 1 章に「目的」として、設計書を作成するプログラムで解決すべき内容を文章にて簡潔に説明した。この文章には、解決すべき内容を導出するアルゴリズムには言及していない。第 2 章に、図解を示した。

4. 教育実施結果

設計書作成のための入力情報を、図解に変更した教育方法にて教育を実施した結果を示す。

作成した設計書、プログラム、ディスカッションで記述したディスカッションシート、アンケートを実施した結果を示す。注意として、欠席や未提出などの都合から、全員の成果が含まれていないため、提出された結果のみを評価対象としている。

4.1 対象者

高等専門学校電子情報工学科 3 年生 41 名を対象に実施した。

文書作成に関する教育としては、設計書の作成についての教育は授業などでは行われていない。一般的なライティングに関する教育については、実験レポート作成などで、助言する形にとどまっている。

プログラミングについては、2 年次に通年で C 言語を授業にて学習し、一通りの知識を習得している。3 年次では、実験やアルゴリズムに関する授業などで、C 言語によるプログラミングをしている。

4.2 設計書

第 1 回の授業にて、クラスを半分に分けて、半分は演習課題 1、もう半分は演習課題 2 の設計書を、Word にて作成するように指示した。作成の際には、「1. プログラムの目的」「2. 入力」「3. 出力」「4. 定義」「5. 処理内容」「6. 注意事項」を設けたテンプレートファイルを配布した。「4. 定義」「5. 処理内容」以外は一通りの内容が記載されている。残された 2 か所の章を中心に入力することが主な実施内容である。以降にて作成した設計書の記述状態を紹介する。

4.2.1 箇条書き・順序リスト

「処理内容」の章では、繰り返し処理を含むアルゴリズムを表現するために、箇条書きや順序リストを用いる方法が考えられる。

学生が作成した設計書を確認すると、番号がつかない箇条書きの表現を用いた設計書は 1 件、番号を付けた順序リストを用いた設計書は 4 件であった。これら以外の 23 件の文書では、通常の文章を用いた表現であった。

通常の文章の中では、「はじめに」、「次に」、「最後に」など、順番を表す表現で、前後の文章と関連させる方法を用いている様子が見られた。一方で、図解で用いた具体的な数値をそのまま用いた文章も見られ、アルゴリズムを抽象化することや、順番で整理することなどがまだ十分でない文章である設計書が見られた。

4.2.2 導入文章

処理内容の手順を述べる前に、処理の概要が示されていると、詳しい内容を理解するための事前理解を促すことにつながる。特に箇条書きを用いる場合には、

表 2 文書のアルゴリズム評価

演習課題	提出件数	一致している設計書
演習課題 1	16 件	13 件 (81%)
演習課題 2	11 件	9 件 (82%)

導入のための文章があることが好ましい。そのような概要が十分かかれていた設計書は 2 件、十分ではないが導入の文章があった設計書は 3 件であった。

導入文章を書くことや、概略をまとめることが行われていない様子が見受けられる。

4.2.3 アルゴリズム

図解で示した内容のアルゴリズムを文書化して設計書を作成するので、文書には図解と一致したアルゴリズムが示されていることが期待される。

学生が作成した設計書の処理手順部分に書かれた文章のアルゴリズムが、図解のアルゴリズムと一致しているかどうか確認すると、アルゴリズムが一致した設計書は 22 件であった。その内訳を表 2 に示す。

アルゴリズムが一致していない設計書の様子として、演習課題 1 では、アルゴリズムに触れられていないものの、内容が混乱しているものがあつた。演習問題 2 では、分量として少ない文章であるものや、ソートのアルゴリズムを記載している設計書が見られた。

4.3 プログラム

学生が作成したプログラムの状態として、実行ができるか、正しい解を表示するか、により評価した。提出のあつたプログラムを分類した結果を表 3 に示す。

提出されたプログラムの様子を見ると課題内容にかかわらず、実行可能な状態であつたが、正しい結果を表示するプログラムであつたものは、演習課題 1 で 5 件 (50%)、演習課題 2 で 10 件 (83.3%) と大きな差がみられた。演習課題 1 で菓子の個数を求める課題は、内容としては難しくないのだが、求める値が 3 種類あることから、ループを複雑に考えている様子や、導出した個数の結果をすべて保持しようとして複雑な処理を実装している様子が見られた。

他方、演習課題 2 の順位付けの課題は、順位付けが

表 3 プログラムの状況

課題内容	提出件数	実行可能	正しい
演習課題 1	10 件	10 件	5 件
演習課題 2	12 件	12 件	10 件

決まる原理を理解するには難しさはあるが、示された順位付けの処理方法が分かれば、それをプログラムにすることは難しくないと考えられる。そのため、正しいプログラムが完成したものが多いと考えられる。

4.4 アンケート

ディスカッション後の授業終了時に実施したアンケートでの結果を示す。

4.4.1 設計書を書くときに難しかったこと

「設計書を書くときに難しかったことはどのようなことですか」との質問に対して、選択肢から択一させた。その集計結果を図3に示す。「何を書くべきか考えること」が最も多かった。アルゴリズムを文書で伝えるには何を書けばよいか考えている様子が見られた。さらに、3番目に多く「文章による表現方法」が選ばれており、何をどのように書けばよいかを考えている様子であることが伺える。2番目に、「アルゴリズムを考えること」が多かった。図解からどのようなアルゴリズムであるか理解することに、一定の難しさがあるといえる。

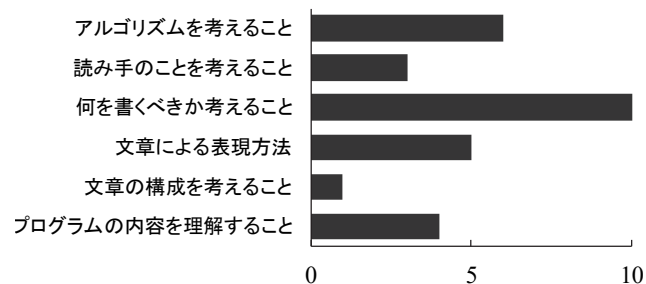


図3 設計書を書くときに難しかったこと

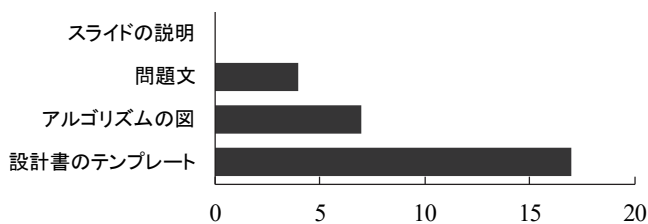


図4 設計書作成時に役立ったもの

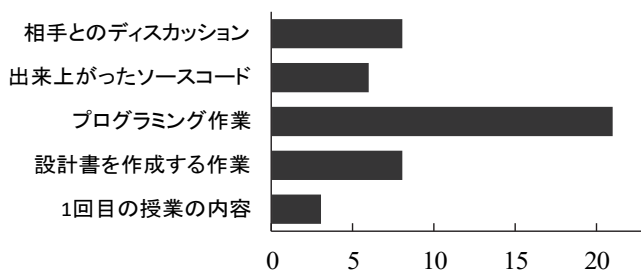


図5 ソフトウェア開発における文書の役割の理解

4.4.2 設計書を作成するときに役にたったもの

「設計書を作成するときに役に立ったものは何ですか」という問いに対して、選択肢「設計書のテンプレート」、「アルゴリズムの図」、「問題文」、「スライドの説明」から一択する質問の結果を図4に示す。

「設計書のテンプレート」が最も多かった(58.6%)。「アルゴリズムの図」が多く選ばれることを想定していたが、24.1%のみであった。

このような結果になった理由として、設計書のテンプレートには、「データ構造」と「処理手順」以外の部分が入力されていたため、全体の文書を作成するうえでは、すでに入力されていることが「役立った」と判断されたのではないかと考えられる。

文書作成のための入力情報としては、図解のみが与えられているので、図解とソースコードによる入力情報を比較して、どちらが良いかどうかを本結果から判断するのは困難である。

4.4.3 文書の役割を理解するきっかけ

「ソフトウェア開発における文書の役割を理解するきっかけになったことはどれでしょうか」という質問に対して、複数選択可能な選択肢を用意した。図5に集計結果を示す。

その結果、「プログラミング作業」が21件であり、「設計書を作成する作業」、「相手とのディスカッション」がそれぞれ8件であった。

設計書を読んで、プログラミングするという行為を通して、どのような内容や記述が必要であるか、実感できていることが確認できた。さらに、実際に文書を作成する作業や、相手から自分の文書について聞くことや文書について討議することが役立っている。文書やプログラムにまつわる作業を通して文書の役割を理解している様子が見える。

4.4.4 この授業で得られたこと

「この授業で得られたことは何でしょうか」という自由記述の問いに対して、目標とした気づきに関連する回答があった。以降に、目標に該当する具体的な記載を示す。自由記述を分類した結果を表4に示す。

「コミュニケーション」、「理解」に分類した、文書を客観的に見ることに関する気づきとして、「相手に伝える文章を書くことが必要である」、「自分の理解と人の理解に差がある」、「相手に自分が思った通りのこと

表4 自由記述の分類

カテゴリ	件数
コミュニケーション	8
設計書	8
日本語の書き方	6
プログラミング	4
理解	3

を伝えるのが難しい」などの記載があった。

「日本語の書き方」に分類した，日本語による文書作成力に関する気づきとして，「日本語の難しさ」，「簡潔な文章を書くこと」，「何が必要か，書く順番（をどうすればよいか）」などの記載があった。

「設計書」，「プログラミング」に分類した，ソフトウェアの設計，開発における文書の役割に関する気づきとして，「プログラムを書く前に設計書を書くべきだと思った」，「設計書からプログラミングする方法」などの記載があった。

4.5 ディスカッションシート

ディスカッションシートには，「立場」として「プログラマ」と「設計者」を設け，それぞれに「対象」として「プログラム」と「設計書」を設けた。

ディスカッションシートに記載されていた文の数を「立場」と「対象」ごとに計数した結果を表5に示す。プログラマの立場から，設計書に対する議論が多いことが分かる。これは，プログラマは，設計書を読んで，プログラミングという作業を行う立場であることが関係していると考えられる。プログラマは，設計書を手がかりにプログラミングするので，実際のプログラミングに十分か，設計書を精査することになる。そのため，不十分な内容やあいまいな表現などに気づきやすいと考えられる。

次に，ディスカッションシートに記載されていた内容を検討する。

(1) プログラマ-プログラム

プログラマの立場から，プログラムに関する記述で多く用いられていた名詞として，「設計書」，「実装」，

表5 ディスカッションシートにおける立場と対象ごとの記述件数

立場\対象	設計書	プログラム
設計者	33	27
プログラマ	44	31

表6 プログラマ-プログラムディスカッション語句

出現回数	語句
4	実装，入力，設計書
3	通り，変数，プログラム

表7 プログラマ-設計書ディスカッション語句

出現回数	語句
7	処理内容
6	設計書，プログラム
5	手順
4	自分

表8 設計者-プログラム ディスカッション語句

出現回数	語句
5	自分
4	順位，プログラム，想定
3	部分，実装，設計書

表9 設計者-設計書 ディスカッション語句

出現回数	語句
4	ポテトチップ
3	金額，定義，処理内容

「入力」，「通り」，「変数」，「プログラム」がある。出現回数の多い語句を表6に示す。

設計書通りに実装できたかどうか，入力の処理や変数名や変数の定義などに関してプログラミング時に検討した様子が見られる。

(2) プログラマ-設計書

プログラマの立場から，設計書に対する記述に多く含まれている名詞として「処理内容」，「設計書」，「プログラム」などがある。出現回数の多い語句を表7に示す。

設計書にある「処理内容」の章に対して，プログラムにすることを考えたときに，その手順が分かりやすかったかどうか，議論している。自分で情報を補う様子や，自分の理解力について振り返る様子が見られた。

(3) 設計者-プログラム

設計者の立場からプログラムに関する記述として，「自分」，「順位」，「プログラム」，「想定」などのキーワードの出現頻度が高かった。出現回数の多い語句を表8に示す。

自分の意図した設計内容がプログラムに反映されているかどうか，設計書の記述と合わせて議論している様子がうかがえる。

(4) 設計者-設計書

設計者の立場から設計書に関する議論として、「ポテトチップ」、「金額」、「定義」、「処理内容」などの語句の出現頻度が高かった。出現回数の高い語句を表 9 に示す。

ポテトチップスや金額など、具体的な処理内容の表現方法が適切であったか、定義や処理内容が相手にプログラミングしてもらうために十分かどうか、議論している様子が見える。

5. 図解への変更による影響

本稿の目的とした、入力情報を図解にしたことによる影響を考察する。

5.1 入力情報としての役割

プログラムが想定したアルゴリズムで作られている割合は、演習課題 2 題の平均値は 81.5%であった。この割合から、設計書に書くべきアルゴリズムが図解を通して伝達できていると判断できる。また、アルゴリズムの一致状態は、演習課題 1 では 81.0%であり、演習課題 2 では 82.0%であることから、演習課題による差はほとんどないと見なることができると考えられる。

したがって、アルゴリズムの伝達を目的として、入力情報を図解に変更したが、変更したことによる悪影響は見受けられない。問題による差異も小さいと考えられるため、文書化するうえでは、難易度に違いはないといえる。

5.2 学習目標への影響

本教材の目標は、ソフトウェア開発における文書の役割を理解することと、文書作成能力が必要であることに気づくこととした。

アンケートでの「授業で得られたこと」の回答と、ディスカッションでの議論内容から、学習目標に該当する記載が確認できた。学習目標を達成するために十分であるといえる。

6. まとめ

ソフトウェア開発などの技術者に必要とする文書作成能力を高めることを目的として、設計書作成とプログラミングをする教材を考案し、ソフトウェア開発に

おける文書の位置づけや文書によるコミュニケーション、日本語表現に関する気づきを目標として実施した。

従来は、設計書を作成するための入力情報として、ソースコードを用いていたので、ソフトウェア開発プロセスに沿った進め方と逆行していた。そこで、アルゴリズムを表した図解を入力情報とし、そのアルゴリズムを設計書にあらわす教材とした。

この変更した教材を用いて教育を実施した結果、目標に見合う気づきがアンケートや活動の成果物などから得られた。図解による表現による影響や、問題の差異も小さい。

図解を設計書作成の入力情報としたことで、プロセスに従って開発する技術者を対象に実施しても、違和感を与えない教材になったといえる。

今後の展開として、本教育方法での気づきをもとに、具体的な文書作成能力を高めるための演習問題を作成して実施することを検討している。さらに、今回のようにプログラミングさせることで効果はあるが、コンピュータを用意する手間や、プログラミング言語の習熟などが関与するので、コンピュータを必要とせず、同様のコミュニケーションや作業を介した教材を検討したい。

謝辞

本研究は JSPS 科研費 JP 17K12937 の助成を受けたものである。

参考文献

- (1) 藤田悠, 山本樹, "プログラミングを介した技術文書作成教育の改善と実践", 教育システム情報学会研究報告, vol.30, no.6, pp83-90, (2016)
- (2) 藤田悠, 山本樹, "プログラミングを利用した技術文書作成教育の実践と検討", 教育システム情報学会研究報告, 28-6, pp.51-58 (2014)
- (3) 藤田悠, "プログラミングを用いた文書作成技術育成のための導入教育教材の改善", 日本教育工学会 第 33 回全国大会, pp331-332, (2017)
- (4) 藤田悠, 山本樹, "プログラミングを用いた社会人技術者向け文書教育の実施", 日本教育工学会 第 32 回全国大会, pp.211-212, (2016)