

プログラミング序論演習資料

～レポートの書き方:プログラミング序論演習を例にして～

2001-5-29 加藤直樹

1. はじめに

この資料ではレポートの書き方について説明します。レポートは書く内容や書き手の個性によって色々な書き方があり、このような書き方が絶対というものはありませんが、ここでは目安(ガイドライン)を示します。他の授業でもレポートの書き方の指導があると思いますが、それらと共に、レポートを書くときの参考資料として利用してください。

2. レポートの内容構成

2.1 全体構成

プログラミング序論演習のレポートの内容は、作成したプログラムやプログラムを使った実験、検証の報告が主だと思います。したがって、レポートの全体構成は次表のようになります。プログラムを二つ作成したり、二種類の実験や検証を行ったりした場合は“本文”の部分を二回繰り返す形にするのがよいでしょう。

表1 レポートの全体構成

プログラムの場合		実験・検証の場合	
要旨			
はじめに			
概要	本文	概要	
設計		実験環境	
実現		実験手法	
実行結果		実験結果	
考察		考察	
おわりに			
謝辞			
参考文献			
付録			

2.2 構成詳細

次にレポートを構成する各部分でどのようなことを述べればよいのかを示します。

(1) 要旨

“要旨”ではレポートの内容を簡潔に述べます。“おわりに”に書いたことをより簡潔にまとめればよいでしょう。要旨は書かなくてもかまいません。また、要旨には見出し番号をつけません。

(2) はじめに

“はじめに”では、どうしてこのレポートに示すプログラムを作ろうとしたのか、どうして実験をしようとしたのかといった背景や目的を述べます。また、本文の構成を簡単に示すと読者が読みやすくなります。“はじめに”は読者が最初に読む部分です。最後まで読んでみたいと思わせることも“はじめに”の重要な役割です。

(3) 本文

“本文”ではレポートの本体、プログラミング序論演習のレポートでは作ったプログラムに関する事、または、行った実験や検証のことを述べます。本文は表 1 にも示したようにいくつかの部分から構成されます。決して“本文”という見出しをつけないようにしましょう。

(a) 概要

“概要”ではどのようなプログラムを作るのか、または、どのような実験や検証を行ったのかを簡潔に述べます。また、実験や検証のレポートの場合、その目的や必要性を述べます。この点を述べる場合、見出しは“目的と必要性”とした方がよいでしょう。一つのプログラムや実験だけしか述べていない場合、“はじめに”の内容がほとんどここと同じになってしまうと思います。そのような場合は“概要”は省略してもかまわないでしょう。

(b) 設計

“設計”では、プログラムにどのような機能をもたせるのか（機能設計）、プログラムを作る上で使用した理論やアルゴリズム、そして、プログラムの流れなどを書きます。また、設計するにあたって前提となることがあれば、それを“設計方針”として最初に述べます。

(c) 実現

“実現”では作成したプログラムソースのリストを示し、そのプログラムを構成する関数や変数の説明、設計で書いた機能やアルゴリズムなどとの対応を述べます。これらの説明は、このような構造でプログラムを作るという文脈（構造設計）で“設計”で述べることもできます。ただし、この場合もソースリストは実現結果なので“実現”で示します。プログラムソースが大きい場合は、付録として巻末に添付し、ここでは巻末に添付したことを明記するだけにした方が読みやすくなるでしょう。

また、作成したプログラムの規模（行数など）、開発に使用した計算機や言語ソフト（開発環境）、実行するために必要な計算機環境（実行環境）を述べます。言語ソフト標準のもの以外でそのソースをレポート内に示していないライブラリやプログラムソースがコンパイルするときに必要な場合は、必ず開発環境として示します。

(d) 実行結果

“実行結果”ではプログラムを実行した結果を示します。実行結果が画面へ出力される場合はハードコピー（画面イメージそのまま）が最適です。printf 文などを使った出力の場合はリダイレクト機能を用いて出力をファイルへ格納し、その内容を印刷するのもよい

でしょう。重要なのは、出力された結果をそのまま示すことが大切です。そこで、示した実行結果をどのように得たのかを述べるのも大切なことです。実行結果も分割が難しく、大きくなってしまふ場合は付録としてかまいません。

入力するパラメータによって出力結果が変わるプログラムの場合、パラメータを変化させて、どのようなパラメータでも動作することを示すのが好ましいです。また、実行時間を計測した場合は、それを示すのも良いでしょう。

(e) 実験環境

“実験環境”には実験に使用した機材や、被験者がいる場合には被験者の情報を示します。また、実験に使用したプログラムがある場合は、(b)設計、(c)実現にならって、プログラムの説明を述べます。

(f) 実験手法

“実験手法”にはどのような手順で実験を行ったのかを述べます。実験環境と共に、ここに書いた情報を元にすれば、同じ実験を再現できるようにします。

(g) 実験結果

“実験結果”では得られた実験結果を示します。実験結果がプログラムの実行結果である場合は(d)実行結果にならってください。実験結果では事実を示し、その結果の議論や議論のためのデータ分析結果は“考察”にまわします。

(h) 考察

“考察”では、実行結果に基づいてプログラムがきちんと動いているかどうかの説明や、実行時間を示した場合はその妥当性など、プログラムに対して考えたことを述べます。また、どうしても正常に動作しない部分ができてしまった場合は、その原因の予想や解決案を述べます。

実験の場合はその実験結果を元に考えたことを述べます。何か疑問や目的があつて実験をしたのでしょうから、その疑問や目的にからめて、議論を展開します。

(4) おわりに

“おわりに”には、レポートのまとめとして、ここまで書いたことを簡潔にまとめます。“はじめに”に目的を書いた場合、その目的に呼応して目的の達成度を述べます。また、プログラム作成や実験にあたっての感想等があればここで述べるのがよいでしょう。

(5) 謝辞

“謝辞”にはレポートを作成する上で協力してくれた人や団体などがある場合は、その人への謝辞を書きます。

(6) 参考文献

“参考文献”ではレポートを作成する上で参考にした文献等があればそれを書きます。ここに書いた情報を見れば読者がその文献を見つけられるように、雑誌なら著者、標題、雑誌名、巻、号、ページ、発行年、単行本なら著者、署名、発行所、発行年などを書きます。参考文献の一つ一つに番号をつけて、本文から参照する場合はこの番号を[1]、¹⁾の

ように示します（下図参照）。なお，“参考文献”には見出し番号はふりません。

本文：

C 言語における 2 次元配列は 1 次元配列で表現されている[1]。

参考文献：

[1] 柴田望洋，秘伝 C 言語問答 ポインタ編，ソフトバンク(1991)。

図 1 参考文献の書き方と参照の仕方

(7) 付録

プログラムソースリストや実行結果が大きく，本文に入れると文章が読みにくい場合は，付録として添付します。添付するものが複数の場合，付録 1，付録 2，... というように番号をつけて，本文から参照しやすいようにします。なお，付録は必ず本文から参照してください。本文でなにも説明せずに付録だけをつけても何も意味がありません。

3. レポートの外部体裁

3.1 表紙

レポートには必ず表紙を付けます。表紙には授業名（プログラミング序論演習）と自分でつけた標題，担当教官名，提出日，所属，学籍番号，氏名を書きます。学科掲示板のプログラミング序論演習の場所から，情報コミュニケーション工学科規定の表紙がダウンロードできますので，それを利用してください。

3.2 紙の綴じ方

レポートを構成する紙は左上をホッチキスなどで綴じます。ページ数が多く，左上だけでは不安定な場合は上部二箇所ほどを留めます。ゼムクリップなど簡単に外れてしまうものは絶対に使わないようにしましょう。

レポート全体では紙を縦に使っているときに，付録などでどうしても紙を横に使いたいページがある場合は，横向き使用の紙の上部が縦向き使用の紙の左側にくるようにします（下図参照）。

また，レポートを再提出するときは，前回のレポートを新しいレポートの後ろにつけてください。

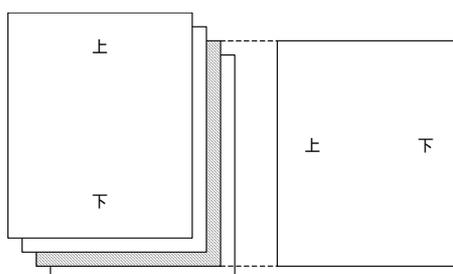


図 2 横向き使用の紙の入れ方

3.3 切り貼りするとき

本からのコピーをレポートに載せたいときなど別の紙をレポートに貼る場合は、必ず原稿をコピーしたものを提出してください。この理由は糊のつけすぎが原因でレポートがよごれてしまっていたり、貼ったものがはがれてしまったりするかもしれないからです。

4. 文書の体裁

4.1 文字

本文の文字フォントは明朝系、10.5 ポイントを使います。見出しはゴシック系を使用し、若干フォントや行間を大きくして見やすくする工夫をしてください。

また、英数字は原則として半角文字を使用します。逆に日本語は必ず全角文字を使用します。

4.2 文章構造

文章構造について注意すべき点を次に示します。

- ・ 段落の最初は必ず 1 字あけます。余計な改段落、改行は読みにくくなるのでやめましょう。また、長すぎる段落も読みづらいので、きちんと意味で改段落を行います。
- ・ 段落の前後には余計な空行を入れないようにしましょう。空行を入れないと見にくというのは、段落が長すぎたり、短すぎたりという、文章の構造がおかしいことが多いです。
- ・ ある文の下に書いたものを示すときには、「以下に」ではなく「次に」を使用します。「以下に」は、その部分以降すべてを指すことになります。
- ・ 図や表は文ではありません。見出しの後にいきなりプログラムや図、表を書かないようにしましょう。まず、まず文章が必要です。箇条書きの前も同じです。

4.3 見出し

見出しには書いてある内容を具体的に簡潔に説明するものをつけましょう。

章節見出し番号の付け方の例を図 3 に示します。一般にレベルは 3 段階、つまり、章、節、項までにしましょう。それ以降は見出しつきの箇条書きを使います。見出し番号の付け方は自分でルールを決めればそれでよいのですが、上のレベルで使った番号の付け方を箇条書きも含めて下のレベルで使用しないことと、丸数字（ など）は使用しないことを守ってください。また、ページの最終行に章節見出しを配置することになってしまった場

1. レベル 1
- 1.1 レベル 2
- 1.1.1 レベル 3

- (1) 箇条書きレベル 1
- (A) 箇条書きレベル 2
- (a) 箇条書きレベル 6

図 3 見出し番号のつけかた

合は、改ページなどを施して、見出しを次のページの先頭に配置してください。

図の見出しは図の下に、表の見出しは表の上に書きます。図表の見出しも図や表と違うページになってしまわないようにレイアウトを整えてください。

4.4 表や図

変数の説明などで、変数名とその役割だけを簡単に書くのであれば、表にした方がよいでしょう。図や表は文章を補助するものです。図表で表現しているからといってまったく文章を書かないようなことをしてはいけません。きちんと文章でも説明しましょう。ただし、図表は文章がなくてもその意味がわかるように配慮することも忘れてはいけません。

5. おわりに

この資料ではプログラミング序論演習のレポートを題材としたレポートの書き方について説明しました。レポートは他人に自分の行ったことや考えたことを理解してもらうために書くものです。したがって、書くときにはいつも読み手のことを考えて、わかりやすいレポートを書くように心がけましょう。