

# レポートの書き方

プログラミングB演習担当

令和7年12月15日

## 1 はじめに

### 1.1 何のために書くか

- 他人にとって理解しやすい報告文, 説明文や仕様書などを迅速に,かつ大量に記述する能力を養う。この能力は,情報処理の技術者・研究者に強く要求されている。
- 学生は与えられた課題に対し,どのように,またどこまでその課題を解決したのかを教員に正確に報告すること。教員は,受理したレポートに基づいて,その課題に対する学生の達成度を判定する。未提出であれば達成度を評価できないため,単位を認定できない。
- レポートをまとめる作業を通して,考えを深め,課題についてより深く理解すること。したがって,演習を終えてからレポートの記載内容を考えるのではない。課題に取り組み始めたときから,考えたこと,作業したことや観察したことなどをノートに記録しておき,それらを材料としてレポートにまとめる。

### 1.2 どのように書くか

- 具体的な記述項目は3節で述べるが,それらの事項について必要十分な内容を書くこと。書くべきことが欠けていたり,書く必要がないことをダラダラ書くことは避けるべき。
- 教員に十分理解してもらえるように,わかりやすく書くこと。書いたレポートは必ず自分で読み直し,説明不足の点やわかりにくい点がないか,誤りがないかを十分にチェックすること。日本語としておかしな文を含んだまま提出することは論外である。

る。特に, LATEX では ‘>’ や ‘<’ などの文字が ‘,’ や ‘,’ のように文字化けを引き起こす可能性があるため,誤字脱字を確認しておくこと。

### 1.3 提出期限

病気,その他やむを得ない理由で受理条件を満たせないときは,できるだけ早く担当教員にその旨を知らせること。

## 2 形式

以下の形式にしたがってレポートを書くこと。

- 用紙  
A4
- 様式  
用紙を縦長にして横書き。一段組。(この「レポートの書き方」のように1ページが左右の「段」に分かれているものを2段組と呼ぶ。)
- ページ数  
15ページ以内。ただし,ソースコードなどを記載する付録はカウントしない。必要十分な内容を簡潔に記載する。文量が少ないと良いことだが,必要な内容が欠けているのは減点対象である。
- 用語  
日本語口語文章体。
- フォント  
本文のフォントは明朝体を用いること。ゴシック体や斜体は強調したい部分のみに用いること。

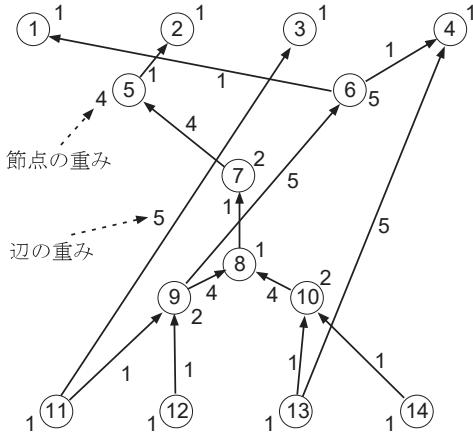


図 1: 図の例

#### • 表紙

レポートの最初のページは表紙とし、本文は書かない。表紙には以下の項目のみを記載する。「提出年月日」は課題の締め切りの日ではなく、自分が提出する日を書く。

タイトル	プログラミングB レポート
課題番号	課題〇〇
提出年月日	令和 年 月 日
教員名	武政 淳二
提出者	学生番号, 氏名

#### • ページ番号

各ページの中央下にページ番号をつける。

#### • 節, 小節

レポートは内容ごとに節に分け、必要ならさらに小節に分け、それぞれに「番号」と「タイトル」をつける。レポートを階層的に構成することにより、おおまかな内容が分かりやすくなる。

#### • 図表

図および表には、通し番号と見出しをつけ、本文中では「図1に示すように、節および辺からなる有向グラフを考える」というように説明する。本文で参照しない図は載せないこと。番号と見出しあり、図の場合は下に書き、表の場合は上に書く。

表 1: 表の例

学年	科目	担当教員	曜日	時限
1	プログラミング B	伊野	月	4
1	プログラミング B	伊野	火	4

## 3 レポート本文の構成

### 1. 課題内容

取り組んだ課題番号を記載。(例:「基本課題および発展課題1,3,4に取り組んだ」)

### 2. アルゴリズムの説明

アルゴリズムとは、計算機上で問題を解く際の算法。アルゴリズム設計は自然言語で記述し、特定のプログラミング言語に依存しない抽象度の高いものである。したがって、この節ではプログラミング言語に依存した説明は避け、この部分を読むことで、任意のプログラミング言語を用いて同じ動作をするプログラムを第三者が記述できるような説明を書くように心がける。

### 3. プログラムの説明

プログラムとは、アルゴリズムをプログラミング言語を用いて実装(具現化)したもの。その実装にあたり、どの部分の処理を関数としてまとめるのか、あるいはそれらがどのように呼び出しあって処理が進むのか、などの設計を記述する。

#### 3.1. 入力形式

プログラムへの入力として、何をどのように与えるのかを記述する。

#### 3.2. 出力形式

プログラムが何をどのように出力するのかを記述する。

#### 3.3. データ構造

プログラムがどのようなデータをどのように格納するのか(型など)を記述する。

#### 3.4. 大域変数

複数の関数から読み書きを行う大域変数を記述する。

#### 3.5. 関数の設計 (main() 関数含む)

各関数が何を引数として必要とし、何を処理するのか（返すのか）、および関数間の呼び出し関係（図示が望ましい）を記述する。

- 3.6. 関数の説明各関数が所望の処理をどのように行うのかを、必要に応じて「2. アルゴリズムの説明」の記述を参照しながら説明する。

#### 4. 工夫した点

誰にとってどのような嬉しさがある工夫なのかが分かるよう、プログラム作成に関して評価してほしいところを書く。レポートにおいて、考察と並んで最も他者と差別化できる箇所であり、客観的に有用な工夫をプログラムに施した者は、積極的に詳述すること（評価対象となる）。

#### 5. 考察

プログラムの設計や作成にあたり考察したこと、よりよいプログラムにするための改善点、あるいは実行結果に対する考察などを客観的に記述する。（「～だと思う」という結びは主観的な感想であるので原則用いない。）

#### 6. 感想

文字通りの感想。この部分は主観的な記述でよい。

#### 7. 作業工程

課題を提出するまでの作業一覧とそれぞれに要した累積時間を列挙する。作業内容の例として以下を挙げる。

- アルゴリズム設計
- プログラム設計
- コーディング
- デバッグ
- レポート作成

#### 8. 参考文献

本や雑誌などを参考にしたときは、文献のリストをつける。本文中では文献番号を[]でくくって引用する（4節参照）。

#### 付録 プログラムリスト

非常に長いプログラムの場合、必要十分な部分のみを掲載すること。必要十分というのは、プログ

ラムを説明するうえで支障がなく、少なすぎでも多すぎてもいいこと。ある部分を省略することで、他の部分を説明できなくなる場合は、省略すべきではない。

## 4 文献引用のしかた

本については、著者名、題名、出版社名、発行年を書く。雑誌については、著者名、記事名、雑誌名、巻号、記事のページ、発行年を書く。下記のうち、[1] が本の例であり、[2] が雑誌の例である。

[1] 粟原徹：“プログラミング入門 C 言語,” 実教出版、2005.

[2] 柴山悦哉：“デザインパターン,” bit, Vol.31, No.5, pp.3-10, 1999.

引用は、「～のアルゴリズム [3] を用いた」あるいは「文献 [4] にあるように」などとする。

## 5 生成 AI の利用について

レポート課題は文章の執筆能力を向上するためのものであるため、生成 AI により作成した文章を利用することは推奨しない。ただし、完全に禁止することができないため、生成 AI により作成した文章をレポートで利用した場合の規則を定める。

具体的には、レポートにおいて、生成 AI により作成した文章を二重引用符で囲み、末尾に “[\*生成 AI\*]” と記述すること。例えば、「本解法では中置記法で記述された数式を逆ポーランド記法に変換し処理する。“逆ポーランド記法とは被演算子の後に演算子を記述する記法である。”[\*生成 AI\*] 2つの被演算子と 1つの演算子の組を読み込み後、被演算子を演算子の入力とし計算する。」とレポート内で記述すると、2文目が生成 AI により作成された文章となる。そして、使用した生成 AI (例えば ChatGPT 等) を参考文献として掲載すること。

また、生成 AI より作成した文章の中で剽窃が見つかった場合、使用した学生に責任があるとし、通常の剽窃と同じ扱いとする。レポートにおける剽窃は、試験のカンニングと同じように扱われるため、本講義だけでなく、今期すべての講義の単位が没収となる。

## 6 コメント集

### • アルゴリズム設計

- L<sup>A</sup>T<sub>E</sub>X の itemize 環境やフロー図などを併用し、以降で記述するプログラムの全体構造を大まかに把握できる。
- × アルゴリズムの入力と出力が分からぬ（拡張課題にどの程度対応しているのか？など）。
- × プログラム変数を参照しながら「プログラムの説明」をしている。

### • プログラムの説明

- アルゴリズム設計で大まかに示した各項目に対し、プログラムリストの行番号を示して実装の説明をトップダウンに展開している。
- プログラムをうまく構造化できている場合、各項目がコンパクトになっているので、プログラムリストの抜粋を掲載できる。
- × プログラムリストのコピペーストで、文章としての説明がない。
- × プログラム文を自然言語に変換しているだけ。例えば、コード中の「a=b;」という行に対し「aにbを代入する」といった説明を、コードの各行に対して行って説明を構成すること。このような説明は、プログラムを見れば分かることであり、本来説明すべき代入の意図（その代入文で何をしたいのか）が分からぬいため適切ではない。

### • 考察

- 客観的な立場からプログラムの完成度について述べている。
  - \* データ構造に関して（主記憶消費量など）
  - \* アルゴリズムに関して（時間計算量など）
  - \* プログラム構造に関して（関数をうまく使っているか）
  - \* 工夫点に関して（本当に使える工夫点か）
  - \* 実行結果に関して（時間計算量と合っているか）
- トピックごとに考えを述べている。

- \* 関数の利用に関して（利点・欠点に関して、どのような場合に使うべきか、回避すべきか）
- \* 拡張性に関して（時刻表などのデータ更新にどのように対処すればよいか）
- \* インタフェースに関して（ユーザにとって使いやすく、適切な情報を提示しているか、またその根拠）
- × 「関数をうまく使っている」などの根拠のない感想を述べている。何がどのようになっているから「うまく」なのかが分からぬ。

### • プログラムリスト

- 行番号とともに、インデント済のリストを掲載している。
  - \* \$ cat -n prog.c > linedprog.c
  - \* L<sup>A</sup>T<sub>E</sub>X の verbatim 環境

### • その他

- × 引数を一つも持たない関数が多く、それらのなかで大域変数の値を変更している（プログラムの構造化ができてない）。
- × 何を意味しているのか分からぬくらい変数名を省略していて、多人数でのプログラム開発時に難がある。

## 7 おわりに

1.1 節で述べたレポートの目的からわかるように、他人のレポートを書き写すのでは意味がない。何よりもレポートは試験の代わりに課しているのであり、書き写しはカンニングと同罪である。発覚した場合、厳罰を受けることになる。レポートは自分で考え、自分の言葉で書くこと。