

## レポートの書き方について

平成 16 年 4 月 1 日作成

平成 16 年 9 月 1 日改訂（参考文献の書式例を追加）

平成 28 年 8 月 4 日改訂（図 1 章立ての例を修正）

令和 8 年 4 月 1 日改訂（全面的に文言を修正）

### 1 はじめに

一般に、報告書やレポートは、自分の行った仕事や研究の内容を第三者に伝えるために作成するものであり、「内容を知らない人が読んで理解できる」、「時間が経ったあとに読み返しても何をやったかが理解できる」ことが重要である。

さらに、大学で課せられるレポートには、「自分が課題の内容を理解していることを伝える」という役割もある。指導書や参考文献を単に丸写ししただけのレポートは、自分が課題を理解していることを読み手（教員）に伝えることができず、評価が下がることになる。自分の文章で、わかりやすく簡明に書くことが重要である。

以下では、プログラミング演習のレポートに要求される書式と内容のポイントについて説明する。

### 2 レポートの書式について

#### 2.1 表紙

レポートの最初のページには必ず表紙をつける。表紙には、課題名、提出者の学籍番号、氏名、提出年月日を明記すること。

#### 2.2 章立て

レポートの本文は章や節に分け、章番号とその章の内容にふさわしい見出しを付ける。本文の冒頭にはその章の内容について簡単な記述を付けるのが望ましい。図 1 に、章立ての例を示す。なお、章立てを指示された場合には、それに従って作成すること。

- |   |
|---|
| <ol style="list-style-type: none"><li>1. レポート課題の概要</li><li>2. データ構造とアルゴリズム</li><li>3. プログラム</li><li>4. 実行結果</li><li>5. 考察</li></ol> <p>参考文献</p> |
|---|

図 1 章立ての例

#### 2.3 文章の書き方

レポートは第三者に読んでもらうことを前提としたものである。課題出題者以外の人が

読んでもその内容が正しく伝わるように書かなければならない。簡潔に正しい日本語を意識し、「ですます」調ではなく「である」調で書くこと。課題のホームページ等に記載されている「～せよ」「～を示せ」といった表現は、出題の文章であり、報告書では使わない。「～を行った」「～を示す」といった表現を用いること。

文章の話題が変わる場合は改行して段落を変え、段落の開始行は字下げ（行頭に全角空白を1文字挿入）をすること。句点（。）ごとの改行や、段落間への空行の挿入といった、不必要な改行は避けること。

また、記入漏れやミスタイプがないか、提出前によく確認すること。

## 2.4 フォントと色について

基本的に、レポートの背景は白地とし、文章は黒の単色で作成すること。フローチャートなどのカラーを使用する必要性のない図は、基本的に白地に黒の単色で作成すること。本文は10 または 10.5 ポイントの明朝体フォントで記述すること。章や節のタイトルは、ゴシック体（または太字のゴシック体）で、本文より1～2 ポイント大きいフォントを使用すること。

## 2.5 ページ番号

レポートのページ下部中央に、必ずページ番号を入れること。表紙のページ番号は不要であり、次のページから1 ページ目が開始される。

## 2.6 図、表、ソースリストの番号とタイトル

レポートに図や表を用いて、わかりやすいものにすることは望ましい。基本的には手書きの図表は受け取らない。記載した図、表、ソースリストには必ず番号とタイトルをつける。図の番号とタイトルは図の下に、表またはソースリストの番号とタイトルはそれらの上に書くのが一般的である。また、単に図表を記載するだけでなく、本文中でその図表の番号を参照し、図表の内容について説明すること。

## 2.7 参考文献

演習課題や考察課題を解く際に参考にした書籍や URL を、レポートの最後にまとめて記載する。授業の URL は記載しないでもよい。書籍の場合、「著者名」、「書名」、「出版社」、「発行年」の順に記載するのが一般的である。以下に参考文献の書き方の例を示す。

[1] 熊谷毅, 玉城眞吉, 白川洋充, “例題で学ぶ C 言語,” 近代科学社, 2001.

[2] GCC online documentation, <http://gcc.gnu.org/onlinedocs/>, 2026 年 4 月 1 日参照.

### 3 プログラミング演習のレポートの各章における注意事項について

#### 3.1 データ構造とアルゴリズム

データ構造を説明する場合は、その課題において重要な意味を持つ定数、構造体、グローバル変数、ポインタなどについて、宣言部（名前およびデータ型）を示すとともに、その役割について説明することが望ましい。

アルゴリズムを説明する場合は、全体像から先に説明し、次に個々の関数のアルゴリズムを説明する（段階的に詳細を記述していく）ことでレポートがわかりやすくなる。必要に応じてフローチャートなども用いて説明するとわかりやすい。

関数について説明する場合は、関数の宣言部を書き出し、関数の引数、戻り値、処理の内容についてそれぞれ説明することが望ましい。特に、主要なサブルーチンについては、ローカル変数の役割や内部のアルゴリズムについて詳細に説明するとよい。

フローチャートを記載する場合は、「フローチャートの書き方」(flow\_guide.pdf)を参照し、記号の形に注意しつつ、処理内容の大まかな流れがわかるように記載すること。一つの記号内に記載する処理の量（コードの行数）や、記号内の言語（自然言語、疑似言語、C言語）は規格で規定されていないものの、読み手がアルゴリズムを理解できるよう、適度な情報量を持つフローチャートにすることを心がけること。

データ構造、アルゴリズム、関数、フローチャートなどは、課題によって求められる記載事項が異なる場合があるため、詳細は各課題担当者の指示に従うこと。

#### 3.2 プログラム

プログラムでは適切なインデント（字下げ）を行うと共に、空行や空白を適宜入れて見やすい形に整えること。変数名に関しては、ループカウンタなどを除き、なるべく意味のある変数名を使用する。特にグローバル変数に1, 2文字程度の短い変数名をつけると別の変数と衝突しやすいため気を付けること。コメントを適宜入れ、分かりやすいプログラムを書く習慣をつける。最低でもサブルーチン単位でのコメントをつけること。

ソースリストは、等幅フォント（Courier New, MS ゴシック, など）で記述し、見やすくなるようにインデント幅、フォントサイズ、改行幅などを工夫すること。

また、ソースリストに行番号をつけておくと良い。

#### 3.3 実行結果

プログラムの実行結果を記載するとともに、その結果が何をテストしたものであるか、また、その結果から何がわかるかを文章で説明すること。入力値によって出力が異なるプログラムの場合、境界値を入力するなどしてテストを行い、意図したとおりに動作していることを確認し、示すことが望ましい。

### 3.4 考察

課題作成に当たって創意工夫した点や、自分のプログラムの問題点、改良の余地について述べること。自分が採用した設計やアルゴリズムが適切なものであったかなどについて考察すること。例えば、入出力データの制限や精度、および、実行時間やメモリの使用量などについて考察できると良い。

また、他の方法は考えられないかについても検討し、他の方法がある場合は、自分の方法と比較検討を行うと良い。

課題によっては、考察課題が提示される場合がある。この場合は、考察課題に対する解答を記載すること。

### 3.5 感想（任意）

“苦労した”、“やっと理解できた”などは感想であり、考察ではない。また、感想は、本来の報告書には記載すべき内容ではない。しかし、今後の演習の参考になるため、感想あるいは意見を本章に記載しても良い。感想に記載する内容としては以下のような点が挙げられる。

- 課題に対する達成度（自己評価）
- 課題の難易度に関する感想
- 自分が教員なら出題する（同レベルの）課題テーマ

## 4 おわりに

別紙のサンプルレポート（sample.pdf）には、具体例をあげながら、レポート作成時に注意すべき点を示してあるので、そちらも熟読すること。なお、ここで説明したレポートの書き方は、プログラミング演習のレポートに共通する一般的な事項であり、課題によって、別途、記載内容について指示がある場合があるので、指示に従って記載漏れが無いよう注意してレポートを作成すること。

## 参考文献

- [1] 電子情報通信学会（編），“学术论文の書き方，発表の仕方，” コロナ社，1985.
- [2] 木下是雄，“理科系の作文技術，” 中公新書，1981.