

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)

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">1. 課題の目的2. 課題の概要3. データ構造<ol style="list-style-type: none">3.1 定数宣言3.2 型宣言3.3 グローバル変数4. アルゴリズムの説明5. ソースリスト6. 実行結果7. 考察課題8. オプション課題9. 参考文献 |
|--|

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。

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

レポートは白地に黒の単色で作成すること。コメントは赤字で入れられる。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。

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。ただし、手書きの図表は受付けない。ワープロソフトの表作成機能や図形描画機能を使用して作成すること。

記載した図、表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、表の番号とタイトルは表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、必ず、本文中でその図表の内容について説明すること。

ソースリストには、ソースの冒頭に通し番号とタイトルをつける。

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 は記載しないでよい。書籍の場合、「著者名」、「書名」、「ページ番号」、「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", pp.20-21, 近代科学社, 2001.

[2] GCC online documentation : <http://gcc.gnu.org/onlinedocs/>.

3. プログラミング演習のレポートの内容について

3.1 データ構造

データ構造の説明では、課題のプログラムで使った、定数、型宣言、グローバル変数の全てについて述べる。変数などの名前、データ型、何を表すものかを説明する。プログラムの宣言部を貼り付けただけでは不十分である。箇条書きにするなどしてわかりやすくなるように工夫すること。

3.2 アルゴリズムの説明

アルゴリズムの説明では、まず、プログラム全体の流れを示し、全体像を明らかにする。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をわかりやすくすること。フローチャートなども用いて説明するとわかりやすい。

関数の説明では、プログラムに含まれるサブルーチンを書き出し、引数や戻り値、処理の内容について説明すること。特に、演習課題で作成した主要なサブルーチンについては、ローカル変数や内部のアルゴリズムについて詳細に説明すること。ソースリストの余白部分に説明を併記してもよい。

フローチャートは通常用いられる形式に従って書く。(詳細は付録 1 を参照すること。) 処理内容の大まかな流れがわかるように、適宜、処理内容を簡単な文章で書くとよい。逆に、一つのフローチャートにプログラムのあまりに詳細な部分までを示すと非常にわかりにくいものになる。ソースコードそのものを書き込んだフローチャートもわかりにくい。

3.3 ソースリスト

プログラムではインデント(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。ループカウンタなどを除いては、原則として意味のある変数名を使用する。グローバル変数には 1, 2 文字の長さの変数名は使わない。適宜、コメントなどを入れ、わ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。担当からの指示が分かりやすくなる。

3.4 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。

3.5 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。

3.6 感想・意見

“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記載しても良い。ただし、記載する場合には別途、感想または意見といったタイトルの節を設けて書くこと。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。

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

など。

4. おわりに

サンプルレポート(別紙)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), “学術論文の書き方, 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.