

プログラミング演習

セメスタ課題報告書の書き方

1 一般的注意

実社会におけるシステム開発において、システムのさまざまな側面について記述された文書の作成作業は、システムの構築作業そのものと同様か、それ以上に重要な作業である。システム開発プロジェクトにおいては、システム開発に関わる重要事項についてできる限り詳細に文書化して関係者に配り、プロジェクトに参加する全ての開発者がつねに同一の目的意識を持っているようにすることが望ましい [1]。

さらに、プロジェクトに参加する開発者の間だけでなく、開発者と外部の関係者との情報交換、情報共有にも文書は重要な役割を果たす。例えば、要求仕様書やシステム取扱いマニュアルは、開発者と顧客との間の情報交換に利用される重要な文書である。

ソフトウェアシステムの開発に関しても、主にソフトウェア工学の分野で、システム開発の過程でどのような文書を作成すべきか、それらをどのように作成すべきか、作成された文書を開発工程にどう活用すべきか、などについての議論が行われている [2, 3, 4]。

本演習においては、ソフトウェアシステム開発に関する各種の方法論を踏まえ、セメスタ課題で作成するプログラム作成の各工程に関する記述を含む報告書を作成する。

報告書の作成にあたっては、以下の点に留意すること。

- C 言語をまったく知らない第三者が報告書を読んだ場合、プログラムの処理内容と利用方法について理解できるようにすること。
- C 言語あるいは他のプログラミング言語を習得している第三者が報告書を読んだ場合、提出されたプログラムと同等のプログラムを作成できるような内容であること。

2 報告書の形式

本演習における報告書の作成形式は、基本的には物理学実験で指示されている報告書の形式を踏襲する。ただし、物理学実験の報告書が実験結果の報告書であるのに対し、本演習の報告書はソフトウェア開発に関する報告書であるため、報告書に記述すべき内容は物理学実験の場合とは異なることに注意せよ。

セメスタ課題への解答として作成したプログラムについて、以下の内容をワープロまたは手書きで記述し報告書を作成すること。用紙は A4 とし、左上一箇所をホチキスで止めて提出すること。ページ数は自由であるが、各ページにページ番号を振ること。

表紙 報告書の表紙には、報告書自身に関する情報、報告書の作成者に関する情報などを記述する。
セメスタ課題の報告書に対しては、

- 科目名、履修クラス
- 課題番号、問題番号、課題名
- 学籍番号、氏名
- 報告書作成日（提出日）
- 作成したプログラムのソースファイル名、実行ファイル名、プログラム作成日（提出日）

を記述すること。

プログラムの概要・目的 作成したプログラムの用途や主たる機能について簡潔にまとめる。すなわち、作成したプログラムが「何を行うプログラムなのか」、「どのような用途に利用できるプログラムなのか」がわかるように記述すると良い。

報告書のこの部分を読むだけで、プログラムの全体像をつかめるように、記述する。ただし、プログラムの具体的な使用法についての説明は次の章にまとめて記述すること。

プログラムの使用法 作成したプログラムをどのように使うのか、特に入出力に着目して記述する。プログラムとユーザとの典型的なやりとりの様子を示した実行例などを交えながら、既存のソフトウェア製品の取扱いマニュアルのように記述すると良い。

特に、プログラムの入出力に関しては、リダイレクションを利用する際に、入力ファイルの作成や、出力ファイルの解釈に役立つような内容を記述することが望ましい。入力ファイルの例や、出力ファイルの例を示し、それらに対する説明を記述しても良い。

入出力に関しては、最低でも以下の内容について触れられているようにすること（必ずしも下記の順番で記述されている必要はない）。

入力

- 何個の値をどのような順番で入力する必要があるか
- それらはそれぞれ何を意味する値か（入力された値はどのように解釈されるのか）
- それぞれについて、どのような条件をみたす値（型や範囲など）が許されているのか
- どのような形式で入力する必要があるか
- どこへ入力するのか（本演習の場合は常に標準入力）

出力

- 何個の値がどのような順番で出力されるのか
- それらはそれぞれ何を意味する値か
- それぞれについて、どのような条件をみたす値（型や範囲など）が出力される可能性があるのか

- どのような形式で出力されるのか
- どこへ出力されるのか (本演習の場合は常に標準出力)
- エラーや特別な入力に対してどのような出力を行うのか

原理 作成したプログラムのアルゴリズムを考えるのに利用した、基本原理について説明する。与えられた課題を解決する手法の理解に必要となる、理論的基礎について記述すると良い。また、基本原理の説明やプログラムの設計についての記述中で利用される各種の概念についても整理し、まとめておく。

与えられた課題が数学の問題を解くプログラムであった場合には、問題の解決に必要となる数学的な性質に関する説明が必要である。同様に、与えられた課題が物理の問題を解決するプログラムであった場合には、物理学上の様々な法則等に関する説明が必要であろう。

重要な式や定理には番号を付け、本文中の必要とされる箇所から参照する。

設計・実装 作成したプログラムのアルゴリズムと、その実装であるソースコードについて説明する。

アルゴリズムの説明にはフローチャートなどを利用すると良い。先に説明した基本原理と照し合せながら、なぜそのアルゴリズムで要求された処理を正しく行うことができるのか、なぜ正しい答えを導き出すことができるのかについて論理的に説明する。フローチャートなどは、適宜まとまった処理に分割して記述し、説明を加えると理解しやすい。

プログラムを(関数定義などの手法を利用して)複数のモジュールに分割して作成している場合には、どのような考え方に基づいてモジュール分割を行ったかについての説明や、それぞれのモジュールの機能、入出力の説明を記述すると良い。例えば、関数定義によって main 以外の関数を作成した場合には、それらの引数や戻り値の意味、型などに関する記述を行う。

フローチャート等で記述したアルゴリズムと、ソースコードの対応についても、自明でない箇所に関しては説明を加えると良い。特に、プログラミング言語の制約などのために特殊なプログラミングテクニックを利用している箇所がある場合には、その説明を行う。

評価 作成したプログラムの機能、性能に対する評価を行う。

いくつかの入力例に対して、プログラムの実行結果を示し、作成したプログラムが正しく動作していることの説明に利用すると良い。

要求された課題に対して、一部実現できなかった機能がある場合には、どのような機能を実現できなかったか、その機能を実現するためにはどのような手法・考え方が必要となるかについて議論すると良い。例えば、入力の値によっては、課題で要求された解とは異なる出力を得るようなプログラムを作成した場合には、どのような条件をみたす入力値が与えられた場合には正しく動作するのか、なぜそれ以外の場合には正しい解を導出できないのか、などについて議論する。

また、作成したプログラムで保証される解の精度はどの程度であるか、プログラムが解を求めるために必要とする計算時間(計算量)や、使用するメモリの量などについて議論するとさらに良い。

考察・まとめ 課題として与えられた問題そのものに関して考察を行う。

通常、どのような分野においても、ある1つの問題に対して、複数の解決法が存在する。作成したプログラムで採用した解法以外の解法をいくつか考え、どの解法がどのような点で有利であるか、また、どのような点で不利であるか、考察する。特に、与えられた問題を解決するための理想

的なアルゴリズムについて、必要とされる計算時間やメモリの量を調べ、作成したプログラムで要する計算時間やメモリの量と比較検討すると良い。

また、与えられた課題と同様の基本原理に基づいた手法で解決可能であるような別の問題や、与えられた課題の発展形となる問題を考え、それらの解決法について議論しても良い。

さらに、アルゴリズムやプログラムを考える過程において判明した問題そのものに関する知見（数学的性質、物理的性質など）がある場合には、それらについても簡潔にまとめると良い。

参考文献 アルゴリズムやプログラムを作成するにあたり、参考にした資料があれば記載する。また、報告書の内容を第三者が理解するために必要と考えられる資料があれば、それらについても記載する。参考文献には番号を付け、報告書本文中で必要と考えられる箇所から表番号や図番号と同じように参照すること。参考文献リストの記述法や参照の方法は、参考文献 [5]などを参考にする と良い。

表・図 表や図には必ず番号とタイトルを付け（表の番号・タイトルは表の上に、図の番号・タイトルは図の下に記載すること）、表番号や図番号は、報告書本文中で説明のある箇所から必ず参照すること。表や図は、本文と同じページ内に記述しても良いが、ページの上部または下部にまとめて掲載すること。

参考文献

- [1] 石川博章. システム工学, series 電気・電子・情報系 (1), 共立出版, 1998 年.
- [2] 鈴木正人. ソフトウェア工学 — プロセス・開発方法論・UML — , Information Science and Engineering-S8, サイエンス社, 2003 年.
- [3] 河村一樹. ソフトウェア工学入門, 近代科学社, 1995 年.
- [4] 伊藤潔, 他. ソフトウェア工学演習, IT Text, オーム社, 2001 年.
- [5] 杉原厚吉. どう書くか — 理科系のための論文作法, 共立出版, 2001 年.