

■プログラミング課題におけるソースファイルの審査視点について

作成: 2011 年 7 月 16 日(土)

審査員 榊 政彦

今大会のプログラミング課題では、先の全国大会に倣いソースファイルの審査を含めました。下表は、課題の概要を示します。一文での説明を加えてはありますが、説明が不十分であると思われることから、審査員内で協議し、コンテスト直前での公開となりますが、ソースファイルの審査視点について述べさせていただきます。

表: 課題概要

番号	課題(概要)
5	構造: ソースファイルのプログラムの書き方が、C 言語が推奨するプログラミング構造になっている。
	書式: ソースファイルのプログラムの書き方が、競技者の一定のルールに従って書かれている。
	読み易さ: 一見して、ソースファイルのプログラムが課題の流れを追跡しやすくなっている。

審査員内でかなり議論した焦点は、参加校のみなさんが、これまでの大会準備の上で、プログラムを作成する際、上記の様な三つの視点で、どの様に意識してこられたかということに尽きました。狭義的には、プログラムは、課題が動けば、それで十分に使命を果たしたことになりますが、実社会、つまり、プログラミングの職場、現場においては、ただ動けばよいということだけではなく、他者への説明責任、後人への引継ぎの為に工夫が求められます。全国大会の設定意図がその為に設定された視点ではなからうかという理解で、今回の審査視点を考えてみました。

これらの視点を評価するに利用できる世界標準、国内標準、業界標準や規格のような明確なものはありません。これまでのプログラミングというソフトウェアを扱う業界、識者の間で、これまでの時代経過の中から慣用として、この方がよりよいとされてきたものが存在するにすぎません(それらの内容は、書籍として、或いは、現在であればインターネットで公開された資料などが多数あります)。一人で完結するものであれば、それは我流でもよいでしょう。しかし、組織人としてプログラミングをすれば、その組織で求められる標準や規格に沿って、ソースコードを書く必要があります。

しかし、C 言語の授業や、各県大会、また、今大会へ向けた現実的なプログラミングの指導現場で、この視点が十分に理解され、意識されたものであったのかということでは、かなり厳しい状況ではないとも考えています。今年度の全国大会に、これらの視点が同じ様に踏襲されるのかも定かではありませんし、昨年の全国大会で実際の審査の視点(基準)がどうなっていたのかも不明であることは事実です。しかし、プログラミングを行う上で、ソースコードをきれいに書き上げる技術も必要であることは、確かなことでもありますので、今大会では、次の様な視点から審査を行います。

今大会の競技を明日に控える今日という時点で、ただ、気をつけるべきは、このことで競技者のみなさんに不要な負担を強いることがないようにということだと思います。これまでの大会対策の中で、競技者のみなさんが身に付けられているプログラミングをベースに、これらの視点をどれだけ加えていけるのかということになりますし、我々審査員側としましても、ソースコードの(一言でいえば)見栄えを、かなりアナログ的な観点(主観)で審査をしなければならぬところもあって、これらの項目については、慎重な審査を心がけなければなりません。審査員としては、自身のこれまでのプログラミングの経験や知識をもとにしながら、しかし、個人的な主義、嗜好をできるだけ排除することも必要だろうと思っています。その為にも評価の視点を事前に公開しておくべきとの考えから、本日提示させていただきます。

>構造

C 言語の一番の特徴が、この構造的に記述することができることです。下記仕組みを使い、プログラムを効率的に組み立てることができる様になっています。これらのいくつかを使いプログラムが組み立てられていることが求められます。

関数*

ソースファイルの分割

外部変数

ヘッダ・ファイル*

静的変数(static)**

ブロック構造*

初期化*

再帰(recursion)

プリプロセッサ

ファイルの取り込み(#include)*

マクロの置換(#define)*

条件付き取り込み

注*: 課題で一番使う頻度が多いもの。

注**: 変数の定義と初期値の設定を同時にする時に、使うことが多い。

>書式

伝統的な記述方法には、次の様なものがあります。

・変数名ー

小文字で書きます。

使用目的に関係した名前にするのが賢明です。

局所変数やループのインデックスには、短い名前を使い、外部変数には長い名前を使います。

自動変数は、同じ名前の外部変数や関数を隠す働きがあるので、その様な変数名は避けるのが賢明です。

・記号定数ーすべて大文字書きます。

・関数名ー使用目的に関係した名前にするのが賢明です。

そして、これらの記述方法が一貫して使われている必要があり、一部だけで摘要されてはいけません。

>読み易さ

プログラムが見やすく書かれていることは、自分にとっても他人にとっても非常に重要なことです。その為に使う方法には、次のものがあります。

・コメントを書き、プログラムでやることを簡潔に説明します。

・字下げ(インデント: タブ)をして、プログラムの論理構造を強調するのに使います。

・1 行には、一つの文だけを書きます。

・(普通は)演算子のまわりに空白を残します。

・処理のブロックのネストは、深くなりすぎないことが肝要です。

一般的にエディタ画面から、はみ出さないようにしたり、

印刷の際、折り返して印刷されないように深さをあまり深くないようにすることが奨められます。

・goto 文は、あまり使いません。

第 10 回高校生ものづくりコンテスト九州大会 電子回路組立部門
プログラミング課題

下図にプログラミングの書き方の一例を示します。この書き方は、審査員が20年前に携わったプロジェクト(社内での検査装置の自動制御システムの開発、エンジニアx6名、凡そ12ヶ月)の時に、プログラミングの書き方について取り決めた共通の書式で記述したものです。

```

kaday01.c - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
/* ===== */
/* 熊本県高校生ものづくりコンテスト大会 第7回 事前講習会 2009年11月28日(土)          タイトル          */
/* 電気系 電子回路組立部門                                                            概要(説明)        */
/* 日立マイコン用基板(RY3048F-ONE TypeH)制御のマイコン 【課題1】                    作成・改版日付(履歴) */
/* ケーブル接続: 基板.CN1~H8側J2、基板.CN2~H8側J3                                     */
/* LEDを全点灯(或いは、全消灯)する。                                                */
/* ===== */

/** ヘッダファイルの定義 **/
#include "3048f.h"                                /* ヘッダファイル          */

/** 外部参照関数の定義 **/

/** 内部参照関数の定義 **/
void InitializeIO(void);                          /* マイコン初期設定      */

/** 記号文字定数の定義 **/
#define LED_AL_OFF    0xff                        /* LED表示: 全消灯        */
#define LED_AL_ON     0x00                        /* LED表示: 全点灯        */

#define OE_ENABLE     0                           /* 許可(負論理)          */
#define OE_DISABLE    1                           /* 非許可(負論理)        */

#define LED7SEG_ALL   PA.DR.BYTE                  /* LED/7SegLED出力        */
#define OE_7SEG       PB.DR.BIT.B7               /* OE:Output Enable/7SEGMENT LED */
#define OE_LED        PB.DR.BIT.B6               /* OE:Output Enable/LED   */
#define LED7SEG_2     PB.DR.BIT.B5               /* 7Segment LED 2        */
#define LED7SEG_1     PB.DR.BIT.B4               /* 7Segment LED 1        */

/** グローバル変数の定義 **/

/* ===== */
/* 関数名: main                                関数の説明          */
/* 引数: なし(void)                             */
/* 戻り値: なし(void)                           */
/* 内容: マイコンのマイコン実行                */
/* ===== */
void main(void)
{
    /* 0-変数 */
    /* 処理 */
    InitializeIO();
    OE_7SEG = OE_DISABLE;
    OE_LED = OE_ENABLE;
    for(;;){
        LED7SEG_ALL = LED_AL_ON;
        /*LED7SEG_ALL = LED_AL_OFF;*/
    }
}

/* ===== */
/* 関数名: InitializeIO                        関数の説明          */
/* 引数: なし(void)                             */
/* 戻り値: なし(void)                           */
/* 内容: マイコン初期値設定                    */
/* ===== */
void InitializeIO(void)
{
    /* 出力端子の宣言 */
    PA.DR.BYTE = 0;
    PA.DDR     = 0xff;
    PA.DR.BYTE = 0xff;
    PB.DR.BYTE = 0;
    PB.DDR     = 0xf0;
}

/* ===== */
/* ===== EndOfProgram ===== */
/* ===== */

```


■プログラミング競技における注意事項について

作成:2011 年 7 月 16 日(土)
審査員 榊 政彦

今大会のプログラミング競技について、下記の通り注意事項を申し送ります。

- ・競技における事務局からの配布物
プログラミング課題
- ・持ち込み可能な資料
今大会において持ち込み可能な資料はありません。
- ・開発環境の条件
今大会の為に準備した持ち込み用のヘッダファイル、自作関数などのソースファイル以外のものは、競技に関係のないものとしてすべて削除してください。
>削除対象例:プログラムのソースファイル、ヘッダファイル、文書ファイル、など。
- ・持ち込み可能なソースファイル
 - >ヘッダファイル
例 CPU 入出力ポートの定義、7 セグメント LED 点灯データ、記号定数など
 - >プログラムファイル
例 CPU 初期化関数、入出力関数、タイマー処理、割り込み処理、LED 表示制御関数など

——
(質問 5:2011 年 6 月 26 日(日)受付)

持ち込み可能なヘッダファイル、ライブラリ、コンフィグレーション用の関数は前日に申請とありますが、どういふものは良くて、どういふものが悪いかわせて下さい。

(回答:2011 年 6 月 28 日(火))

下記の様な定義は、持ち込み可としています。

CPU の入出力ポートの定義

例 スイッチ、CPU の入出力端子状態のオン、オフ

LED の点灯制御

例 7 セグメント LED の点灯文字の設定などを定義した記号文字定数あるいは初期化付き変数

CPU 入出力などの基本的な関数

例 タイマー、入力取込、割り込みなど、

持ち込み不可なものは、次のようなものです。

練習に使ったソースファイル

これまで参考にしてきたソースファイル

お持ち込みになるプログラム作成用のパソコンには不要なソースファイルは、存在しないと考えておりますので、ご理解とご協力の方、よろしくお願ひいたします。

・課題の動作確認

挙手により、審査員へ課題の動作確認依頼を表明してください。

課題の動作確認は、提示されている課題毎に行っても構いませんし、幾つかの課題をまとめて動作確認をしても構いません。このことは、競技者の方がご選択ください。

競技終了直前での挙手は可能です。競技終了後、動作確認をします。

挙手(動作確認依頼)されないと動作確認はできません。

挙手が多く、審査員の動作確認を待つことが考えられます。CPU には、動作確認用のプログラムをロードして審査員を待ちます。

プログラミング作業は、進めてください。

・ソースファイルの提示

今大会では、ソースファイルの書き方についても審査対象となっています。

競技終了時、パソコンの画面に作成した課題のソースファイルを表示しておいてください。

また、その際、今大会に持ち込んだヘッダファイル、自作関数のソースファイルがあれば、それらも表示しておいてください。それらも審査対象です。但し、持ち込むヘッダファイルのうち、CPU メーカー提供のものは審査対象とはしません。