



# *OpenSESSAME Seminar*

組込みソフトウェア技術者・管理者向けセミナー

初級者向けテキスト

組込みソフトウェア管理者・技術者育成研究会  
- SESSAME -

(<http://www.sesame.jp/>)

\*\*\*\*\* 目 次 \*\*\*\*\*

1 . SESSAME の紹介およびコースの概要	1
2 . 開発課題と失敗事例の解説	4
3 . 組込み向け構造化分析の例・設計の概要(1)	21
4 . 組込み向け構造化分析の例・設計の概要(2) 実習/回答と補足説明	44
5 . 組込み向け構造化設計(1)	51
6 . 組込み向け構造化設計(2) 実習/回答と補足説明	70
7 . プログラミング 組込み用語基礎知識	73
8 . ソフトウェアテストの概要	109
9 . プログラミング実習への説明	139
10 . プログラミング 実習	149
11 . プログラミング 実習の回答と補足説明	161
12 . ソフトウェアテスト 実習	164
13 . ソフトウェアテスト 実習/回答と補足説明	166
付録 . 話題沸騰ポットのシミュレーション	174

\*\*\*\*\*

## プログラミング実習への説明

上原 慶子 / 三宅 貴章

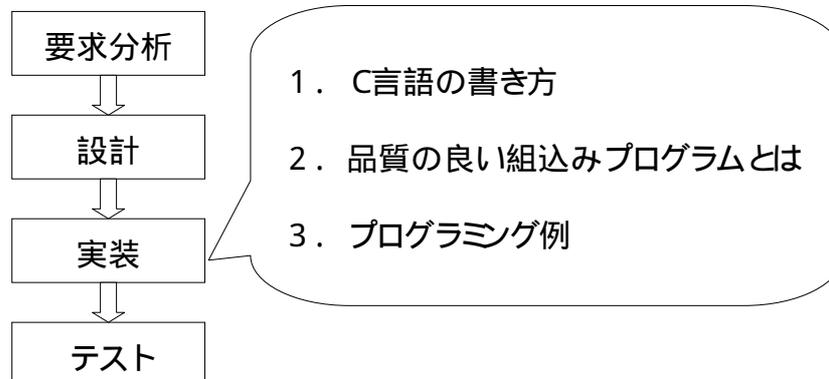


1. SESSAMEの紹介およびコースの概要
2. 開発課題と失敗事例の解説
3. 組込み向け構造化分析の例・設計の概要(1)
4. 組込み向け構造化分析の例・設計の概要(2)  
実習/回答と補足説明
5. 組込み向け構造化設計(1)
6. 組込み向け構造化設計(2) 実習/回答と補足説明
7. プログラミング 組込み用語基礎知識
8. ソフトウェアテストの概要
9. プログラミング実習への説明
10. プログラミング 実習
11. プログラミング 実習の回答と補足説明
12. ソフトウェアテスト 実習
13. ソフトウェアテスト 実習/回答と補足説明

付録：話題沸騰ボットのシミュレーション

277

## アジェンダ



278

## 1. C言語の書き方

---

- 1.1 C言語の特徴
- 1.2 演算子の優先順位
- 1.3 if文に注意！
- 1.4 わかりやすいプログラムの文体

279

SESSAME CONTENTS 2004

## 1.1 C言語の特徴

---

### 1. 関数を基本とした構造化プログラミング

main()関数がトップ。プログラムは関数の集合体  
if else文、while文、for文・・・構造化に適した制御文

### 2. 自由度と簡潔性

変数名などの単語の綴り内以外であれば、空白・改行は自由  
豊富な演算子により、さまざまな書き方が可能

例)

```
i = i + 1;    i = 1;    i++;
```

280

SESSAME CONTENTS 2004

## 1.1 C言語の特徴

---

### 3. 不定 (コンパイラ依存)

同じソースコードであってもターゲットによって動作が異なる場合有。  
コンパイラによってコード生成も異なる。  
コーディング規約が必要。

### 4. 自動変数と静的変数

自動変数 :スタック上に領域を確保。宣言した関数内で一時的に有効  
静的変数 :メモリ上に領域を確保。関数実行後も値を保持

### 5. アセンブリ言語に近い

ビット演算子やポインタなど、アセンブリ言語に近い記述が可能  
オブジェクトコードは他の高級言語に比べコンパクト

281

SESSAME CONTENTS 2004

## 1.2 演算子の優先順位

---

演算子には、優先順位が決められている。  
括弧を活用して誤解を防ごう

例) `if (byte0 & 0x08 == 0 ){...`

演算子“&”と“==”では、“==”の優先順位が高い。  
この記述の場合、`0x08 == 0`の比較結果 (偽 0)と`byte0`を  
AND演算するため、条件式は常に (偽 0)となってしまう

正しくは、...

`if ( (byte0 & 0x08 ) == 0 ){...`

282

SESSAME CONTENTS 2004

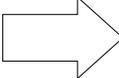
## 1.3 文に注意

### 文は障害を起こしやすい

よくある誤り

1. “==” と “!=” の間違い
2. “&&” , “||” , “=” と “&” , “|” , “=” の間違い
3. 演算子の優先順位の勘違い (前ページで紹介)
4. ifとelseの対応

```
if(k1 == 0 )
    if(k2 == 0 )
        i = i + 1;
else
    i = i + 2;
```



```
if(k1 == 0 )
    if(k2 == 0 )
        i = i + 1;
else
    i = i + 2;
```

283

SESSAME CONTENTS 2004

## 1.4 わかりやすいプログラムの文体

### 1. わかりやすい変数名・関数名

外部変数と内部変数の違いが変数名からわかる  
外部変数は名前から使用目的がわかる / 内部変数は短く  
関数名は、動詞 + 名詞のスタイルで!

### 2. 構造がわかるインデント(繰り返しや判断など)

```
for(i = 0 ; i < 10 ; i+ );
if(k2 == 0 )
    i = i + 1;
```



```
for(i = 0 ; i < 10 ; i+ )
;
if(k2 == 0 )
    i = i + 1;
```

284

SESSAME CONTENTS 2004

## 1.4 わかりやすいプログラムの文体

---

### 3. “括弧の活用” と“空白”

括弧を使って、優先順位による障害を防ぐ  
( )や演算子の前後に空白を入れて読みやすく

### 4. 式は自然な書き方で (は判りにくい)

`if( (k > 10) )`            `if(k <= 10 )`

### 5. 一貫性をもたせる

気まぐれな記述は他人にわかりにくい = バグのもと  
慣用表現で記述の統一を図る

`for(i=0; i<n; )`  
`data[i+]=0;`            `for(i=0; i<n; i+ )`  
`data[i]=0;`

285

SESSAME CONTENTS 2004

## 1.4 わかりやすいプログラムの文体

---

### 6. 数字の記述は避ける

その意味・根拠がわかる名前をつける  
`sizeof` 演算子の活用 … 配列サイズの取得時など

### 7. コメントのつけ方

コードを見てわかるコメントは不要  
変数や定数、関数などには積極的にコメントをつける  
変数・定数 意味、とる値など  
関数 処理の概要、引数、戻り値など

ソースとの相違が無いように …

286

SESSAME CONTENTS 2004

## 2. よい組込みプログラムとは

---

### 2.1 品質のよいプログラムとは

### 2.2 よいプログラムを作る注意点

### 2.3 McCabeの複雑度

287

SESSAME CONTENTS 2004

## 2.1 品質のよいプログラム

---

- ISO/IEC9126の品質特性  
がプログラミング時の注意点

機能性

効率性・実行時間が短く、資源の使用が少ない

信頼性・プログラムの障害発生率が低い

保守性・障害修正や機能追加変更が容易

使用性

移植性・他の環境でも動作可能

288

SESSAME CONTENTS 2004

## 2.2 品質のよいプログラムにするためには

### ・おもな注意点

効率性・書き方によるオブジェクトサイズの違い  
を習得する。(コンパイラにも依存)

信頼性・コーディングミスを引き起こしにくい文体

保守性・わかりやすいプログラム

移植性・標準的な書き方とコンパイラ依存を極力  
排除した記述

289

SESSAME CONTENTS 2004

## 2.3 McCabeの複雑度

- ・ 判定文が多いとプログラムの理解に時間がかかる。
- ・ プログラムの複雑度とはプログラムがどの程度複雑でわかりにくいかを数値で表す。
- ・ プログラムの複雑度の計量法でよく使われているのが、McCabeの閉路複雑度

McCabeの複雑度=判定数 + 1



McCabeの閉路複雑度  $\leq 20$  わかりやすい

McCabeの閉路複雑度  $> 20$  理解に時間がかかる

McCabeの閉路複雑度  $> 50$  作成者以外にはほとんど理解不可能

複雑度を下げするために関数分割をするのはよく考えてから

290

SESSAME CONTENTS 2004

### 3. プログラミング例

---

#### 3.1 状態遷移マトリクス例

#### 3.2 プログラム例その1

#### 3.3 プログラム例その2

#### 3.4 プログラム例の特徴

291

### 3.1 状態遷移マトリクス例

---

状態番号	状態 / 事象	保温設定ボタン	沸騰ボタン	初期沸騰検出	沸騰終了
1	水位不足	変化なし	変化なし	変化なし	x
2	加熱途中	7	変化なし	3	x
3	初期沸騰	7	4	x	4
4	高温保温	7	5	x	x
5	高温沸騰前	8	変化なし	6	x
6	高温沸騰中	9	4	x	4
7	節約保温	10	8	x	x
8	節約沸騰前	11	変化なし	9	x
9	節約沸騰中	12	7	x	7
10	ミルク保温	4	11	x	x
11	ミルク沸騰前	5	変化なし	12	x
12	ミルク沸騰中	6	10	x	10

292

## 3.2 プログラム例その1(switch文)

各状態と事象をそれぞれ、判断文で判定し、関数を呼び出す

```
switch (status){
  case RETAINING : /* 高温保温中 */
    if((cur_switch & BOIL_SWITCH) != 0) /* 沸騰*/
      status = BOILING; /* 沸騰モードへ遷移 */
      prboil(); /* 沸騰処理呼び出し*/
    }
    else if((cur_switch & RETAIN_SWITCH) != 0) /* 保温 */
      status = ECONOMY; /* 節約モードへ遷移 */
      preco(); /* 節約処理呼び出し*/
    }
  ...
}
```

293

SESSAME CONTENTS 2004

## 3.3 プログラム例その2(関数ポインタの定義)

各状態と事象をそれぞれ、二次元の配列で考え、関数ポインタを使って呼び出す

```
struct functbl {
  void (*fp)(void); /* 呼び出す処理関数のアドレス */
  mode_type status; /* 遷移する状態 */
};
mode_type status;
mode_type nextstatus;
#define MAXSTATUS 12
#define MAXEVENT 5
struct functbl datatbl[MAXSTATUS][MAXEVENT] = {
  { {prboil, BOILING}, {preco, ECONOMY}, ... }
};
```

294

SESSAME CONTENTS 2004

### 3.3 プログラム例その2(関数ポインタによる呼び出し)

---

各状態と事象をそれぞれ、二次元の配列で考え、関数ポインタを使って呼び出す

```
void (*callp)(void) {
    for (;) { /* 無限ループ */
        /* イベント待ち処理およびevent領域へのイベント番号の設定 省略 */
        ;
        callp = functbl[status][event].fp; /* 呼び出す関数の設定 */
        nextstatus = datatbl[status][event].status; /* 次の状態の設定 */
        (*callp)(); /* 状態とイベントに対応した処理の呼び出し */
        status = nextstatus; /* 状態の変更 */
    }
}
```

295

SESSAME CONTENTS 2004

### 3.4 プログラム例の特徴

---

#### 1. switch-case文

- 小さなマトリクス向き
- 可読性高いが、モレ・ヌケのリスクあり。

#### 2. 関数ポインタ

- 大規模なマトリクスに有効
- 状態 条件にかかわらず処理時間が一定
- 状態・イベントの追加や遷移仕様の変更が容易
- 変数によってアドレッシングするため、暴走のリスクあり

296

SESSAME CONTENTS 2004

## 本ドキュメントのご利用に際して

- 本著作物の著作権は作成者または作成者の所属する組織が所有し、著作権法によって保護されています
- SESSAME は本著作物に関して著作者から著作物の利用 を許諾されています
- 本著作物は SESSAME が利用者個人に対して使用許諾を与え、使用を認めています
- SESSAME から使用許諾を与えられた個人以外の方で本著作物を使用したい場合は [query@sessame.jp](mailto:query@sessame.jp) までお問い合わせください

SESSAME が著作者から許諾されている権利

著作物の複製・上演・演奏・公衆送信及び送信可能化・口述・展示・上映及び頒布・貸与・翻訳・翻案・二次的著作物の利用

- ドキュメント中には Microsoft 社, Adobe 社等が著作権を所有しているクリップアートが含まれています

---

### OpenSESSAME Seminar

#### 組込みソフトウェア技術者・管理者向けセミナー 初級者向けテキスト

---

2002年10月15日 初版 第1刷発行

2003年10月29日 初版 第2刷発行

2004年3月19日 第2版 第1刷発行

2004年4月30日 第3版 第1刷発行

2004年6月17日 第4版 第1刷発行

**著者** 上原慶子、大野晋、坂本直史、鈴木圭一、須田泉、西康晴、  
二上貴夫、三浦元、三宅貴章、森孝夫、山田大介、山崎辰雄

**編集・発行** 組込みソフトウェア管理者・技術者育成研究会  
(SESSAME)

<http://www.sessame.jp>

---

無断転載・複写、使用を禁ず

Printed in Japan