

# プログラムの書き方

柳本 豪一

今までプログラムの書き方については説明してきませんでしたが、問題点 (バグ) の少ないプログラムを作成する上で有効なテクニックを説明します。今後プログラムを作成していく時には、以下の点を注意してプログラムを作成してください。

グローバル変数にはわかりやすい名前、ローカル変数には短い名前 グローバル変数とは関数の外で定義される変数です。ローカル変数とは関数内で定義される変数です。グローバル変数は、プログラム中のどこからでもアクセスされるので、わかりやすい名前を付けるように気をつけましょう。少しくらい長くなっても構いません。一方、ローカル変数は、関数内で使われるので、短くて意味が分かる名前であれば問題ありません。ただし、短くてもいいといっても、変数の意味が分かるものにすべきです。また、プログラム言語ごとの命名規則がある場合には、それに則るようにしましょう。

(例)

C 言語:変数は小文字、定数は大文字など

C++言語:単語の先頭文字は大文字など

プログラムの構造が分かるようにインデント プログラムを書くときには、読みやすくするため字下げを適宜行うようにしましょう。{}で囲まれるひとかたまりの処理は、字下げすることを進めます。そうすることで、括弧の閉じ忘れやループする処理の範囲が一目で分かるようになります。

かっこを使って曖昧さの解消 プログラム中でかっこがなくても期待通りの動作をする場合がありますが、まとまった処理をかっこで括弧することで、見た目に処理のまとまりがわかりやすくなります。プログラムは動作するだけでなく、可読性もあげるように注意しましょう。

複雑な式は分割 複雑な処理を行うときに、複雑な条件式を書いてしまいがちです。このような条件式は一目で分からないため、問題が含まれていても気づかなくなります。同様に、あまり一行に複雑な処理を詰め込みすぎないように気をつけましょう。

簡潔な記述より分かりやすい記述 プログラムの本など見ると巧みな書き方がたくさん載っています。ところが、あまり乱用すると何をしているのかすぐに分からないプログラムになってしまいます。このようなプログラムは後でメンテナンスをするときに大きな問題となります。

動作に注意 ++は変数の前に付けるのか後ろに付けるのかによって動作が異なります。このような演算子は思わぬ副作用をもたらすことがあります。使うときには注意しましょう。

インデントとプレースのスタイルの統一 プログラムの字下げ (インデント) や演算子の前後に挿入する空白 (プレース) などは、自分のスタイルで統一するようにしましょう。さまざまなスタイルが混在するとプログラムが読みにくくなる原因になります。

慣用句の使用 プログラム言語によって慣用句と呼べるようなものがあります。このような慣用句を使うようにしましょう。例えば、for 文では以下のように書きます。

```
for(i = 0; i < n; i++){  
    .....  
}
```

同じような処理は、以下のようにも書けますが、条件式に等号を含むものは一般的ではないので使わないようにしましょう。

```
for(i = 0; i <= n-1; i++){  
    .....  
}
```

```
i = 0;  
while(i <= n-1){  
    .....  
    i++;  
}
```

他にも無限ループは、以下のような書き方だけを用いるようにしましょう。

```
while(1){  
    .....  
}
```

```
for(;;){  
    .....  
}
```

このように特定の書式を使うことによって、異なる部分が目立つようになり、問題点を探するときに見つけやすくなります。もちろん意図して条件式に等号を利用することは問題ありません。

多分岐には else-if 多分岐をするプログラムを書くときには、else-if を使うようにしましょう。つまり if が多重になっているプログラムを書かないようにしましょう。else を付け加えることで、予想外の入力データに関して条件に当てはまらない場合を捕まえることができるので、問題を発見しやすくなります。

マジックナンバーには名前 プログラムの中に数字をそのまま書かないようにしましょう。プログラムを作成してすぐはその数字の意味が分かっているとしても、時間が経つと分からなくなります。名前をきちんと付けて管理するようにしましょう。

当たり前のコメントは削除 プログラムを見て分かるような処理にはコメントを付けないようにしましょう。コメントがあるために、読みにくくなってしまう可能性があります。

関数とグローバルデータにはコメント 関数はあるひとまとまりの動作をするため、関数の名前からすぐに動作内容が分からなくなることがあります。したがって、コメントを付けるようにしましょう。グローバルデータについても、様々なところからアクセスされるので、きちんとコメントを付けておきましょう。

コメントに真実を記述 複雑なプログラムに修正を加えていくとコメントと実際のプログラムにずれが生じることがあります。このようなコメントは、プログラムの理解の邪魔をすることができますように修正しましょう。

悪いプログラムにコメントは不要 コメントが複雑になりすぎるような場合には、プログラム自体を書き直すようにしましょう。苦労して書いたコメントほど、後で理解できなくなることが多いです。

以上の注意点を守って綺麗なプログラムを書きましょう。