

フローチャートの書き方

1. フローチャートの基本

フローチャートの記号は、日本では日本工業規格(JIS)で決められている。表 1は、フローチャートで用いられる主な記号をまとめたものである。

普通、フローチャートは、プログラムのソースを作成する前の段階、すなわち、設計の段階で作成する。フローチャートを作成する理由としては、以下のようなものがあげられる。

- 図で示されているので、文章だけよりもわかりやすく、プログラムにしやすい
- プログラムを改良する際に効率よくできる
- 1つのプログラムを複数人で作るときに、手分けしやすいなど

フローチャートを作成する際の基本事項は、以下の通りである。

- (1) フローの最初と最後を明記する。
- (2) 処理の流れは原則として、上から下へ、左から右へ。それに逆行する際には、矢印をつける。(教科書によってはすべての処理の流れに矢印をつけている場合もある。すべての処理の流れに矢印つけても問題ない。)
- (3) 線が交差しないようにする。

図 1は、日常生活の一場面のアルゴリズムをフローチャートにした例である。

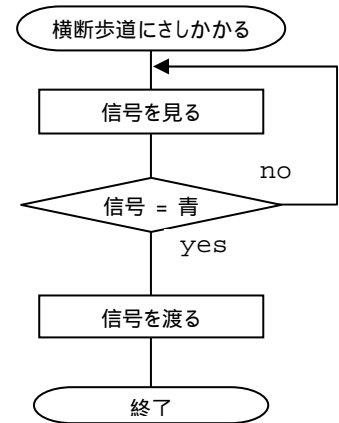


図 1 信号を渡る手順

表 1 フローチャートで用いられる主な記号

	端子 フローチャートの始まり及び終わりを表す		ループの開始
	処理 計算、代入などの処理を表す		ループの終了
	サブルーチン 定義済みの処理を表す		入出力 ファイルへの入出力を表す
	判断 条件による分岐を表す		ページ内結合子 フローチャートが長くなり、ページ内で二列にするとときなどに使う。
	表示 コンソール上への結果の表示を表す		ページ外結合子 フローチャートを次のページに続けるときに使う。

2. 処理の構造

フローチャートでは、基本的に以下の3つの構造を組み合わせてアルゴリズムを表記する。

- 順次構造
- 分岐構造
- 反復構造

順次構造

順次構造とはその名の通り、処理が上から下へ単純に並んでいる構造である。順次構造の例を図 2 に示す。

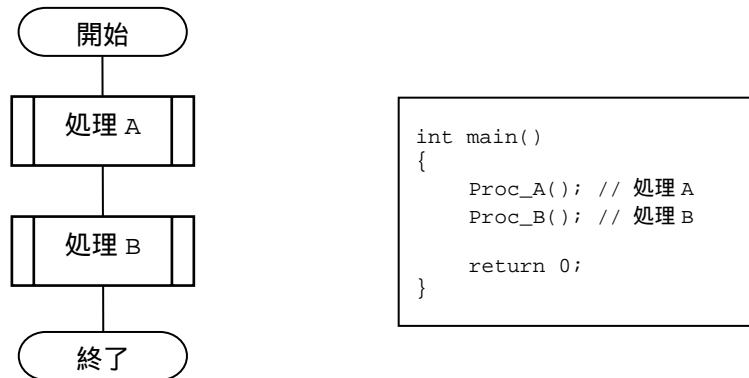


図 2 順次構造

分岐構造

分岐構造とは、ある条件によって処理が別れる構造である。条件は"yes"(または"true","真"など)か"no"(または"false","偽"など)で表される。C では、if 文などで表される。分岐構造の例を図 3 に示す。

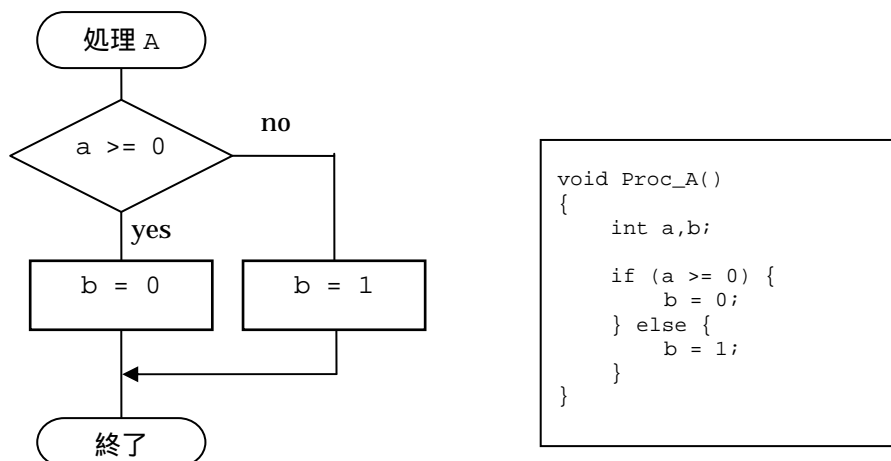


図 3 分岐構造

反復構造

反復構造とは、ある条件を満たしているうちは、処理を実行するという構造である。C での反復構造は、for, while, do while 文などがある。条件をどこで設定するかによって前判定型と後判定型に分けられる。for, while が前判定型、do while が後判定型となる。図 4にそれぞれの場合の例を示す。

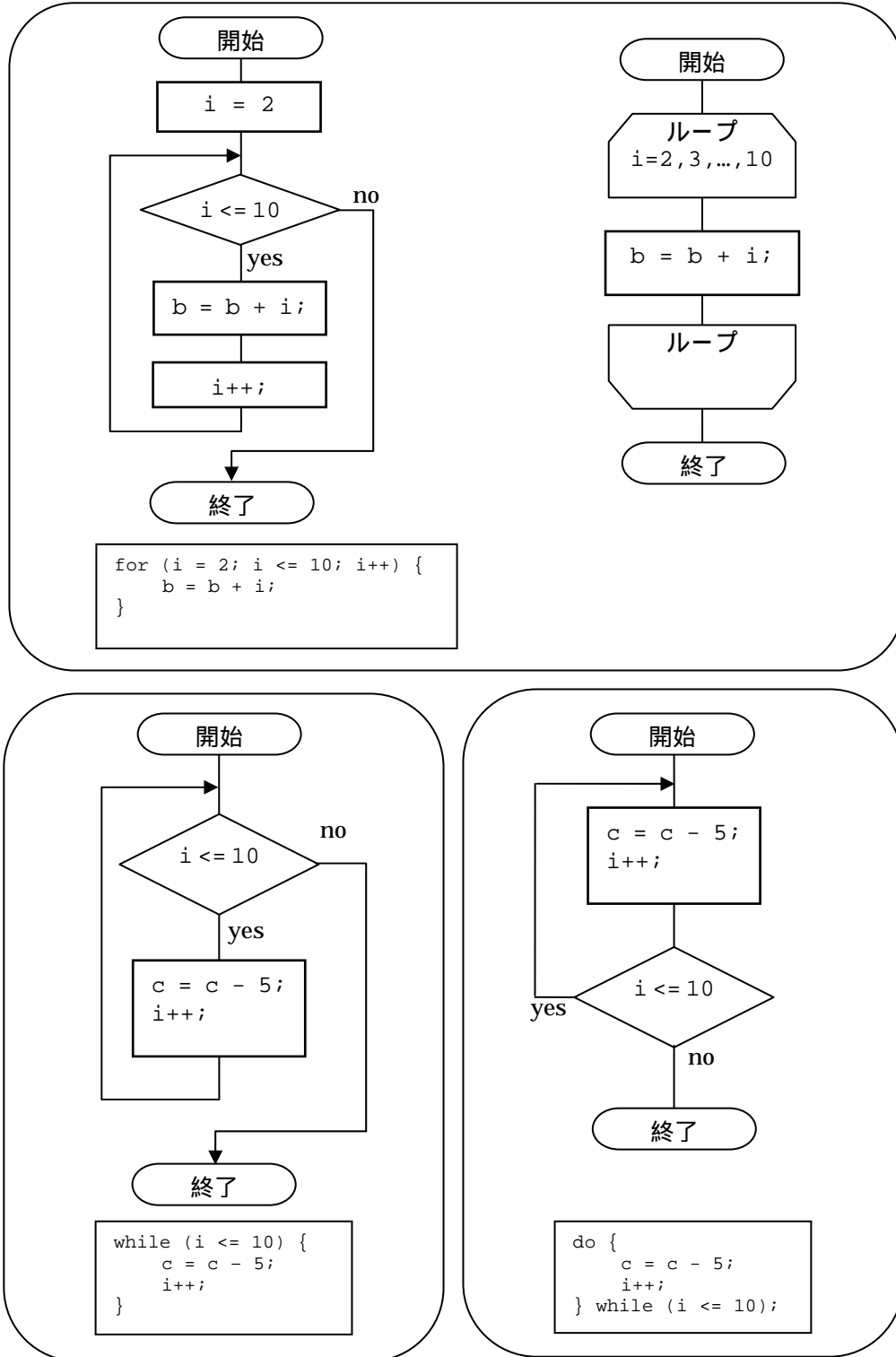


図 4 反復構造

この他にも単一分岐を組み合わせた多重分岐などがある。多重分岐の例を図 5に示す。その他の詳しい部分は各自で調べること。

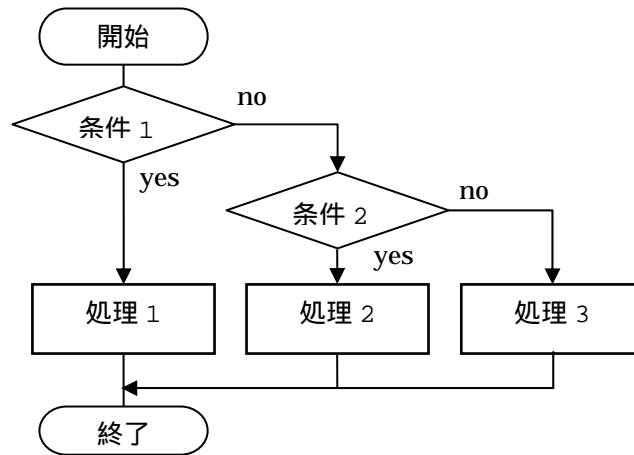


図 5 多重分岐

3. フローチャート作成時の注意点（よく間違える点など）

図 6は、フローチャート作成時に気をつける点をまとめたものである。よく読んで、注意してフローチャートを書くこと。

- 処理の流れを表す線は、箱から離れていないか？
- 分岐の条件が間違っていないか？
- 条件分岐からの出力の線に Yes, Noなどを正しく付記してあるか？逆になっていないか？
- 箱への入力が2本になっていないか？
- 箱の横から入力が入っていないか？
- 白黒で作成したか？網掛けすることで箱の中の文字が読みづらくなっていないか？

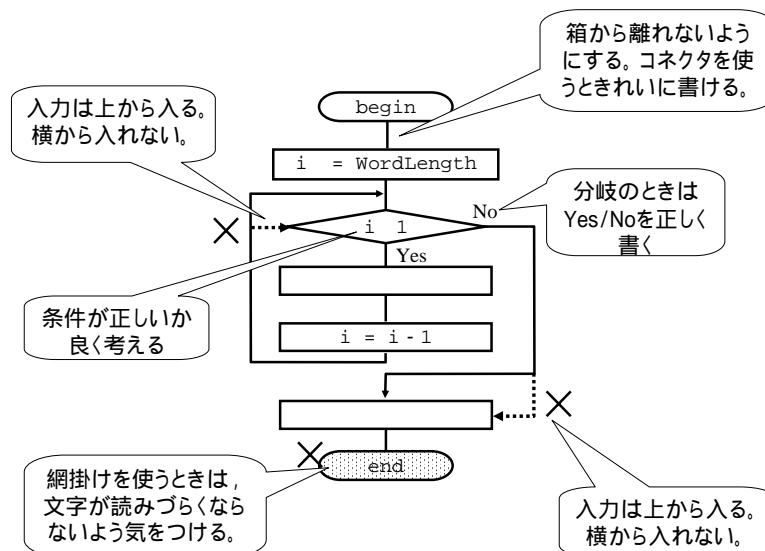


図 6 フローチャート作成時の注意点