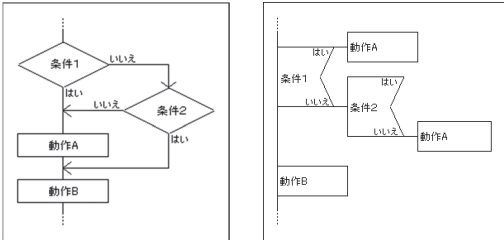


比較



<http://www.happy2-island.com/beginner/invitation02/capter99900.shtml>

プログラムの構成と構成単位

- 基本的な文 – 命令型言語では、代入文
- 接続 – 文を順につなげる
- 分岐 – ある条件が成立するときのみ実行される文を作る
- くり返し – 同じ文を繰り返させる

接続 – 順番に実行する

命令型言語においては、順番に実行すべき文は、(一般に、左から右に、上から下に)順に並べて記す。Fortran では上から下に、Algol 60 では左から右、上から下に、セミコロンで区切って並べる

(例) Algol60では
 $y := 2 * x + 1;$ $z := 3 * y + 2;$
 $x := y + z;$

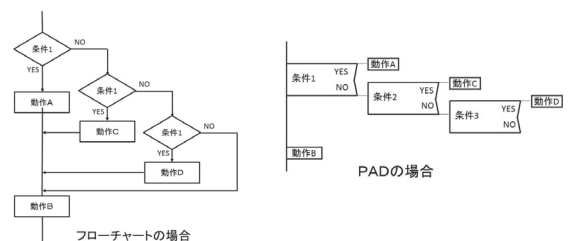
複合文, ブロックともいう

- Algol60では、文の連鎖を begin と end で区切ると複合文(ブロック)となる
- (注) 厳密には宣言文があるブロックとないブロックは分けて考えるが、本スライドでは区別しないことにする
- (例) `begin w := x; x := y; y := w end`
- Algol60 では、文が現れるべきところには、どこでも、複合文(ブロック)が現れて良い。複合文(ブロック)は、1入力1出力である
- (注) 実行の効率性のため(無駄な実行を行わないため)、ブロックを途中脱出する方法(goto文以外)が、現在の言語では提供されている
- (注) 例外という、実行が中断される全く別の枠組みもある

条件文

- 条件「文」というより条件「構造」の方が正しいだろう
 - 文と式から「条件文」を作り上げているから
- ある条件が成立したときに限りある文を実行する。従って、if ... then ... が基本。
 - 勿論、これだけでは使いにくいので、if ... then ... else ... が使われる。
 - さらに、if 構造の末尾を明確にするために、現在では、if ... then ... else ... end とすることが多い

条件文

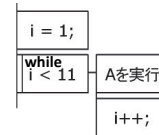
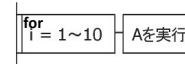


http://www.ced.is.utsunomiya-u.ac.jp/lecture/2013/prog/p1/kadai4/kadai4_2.html

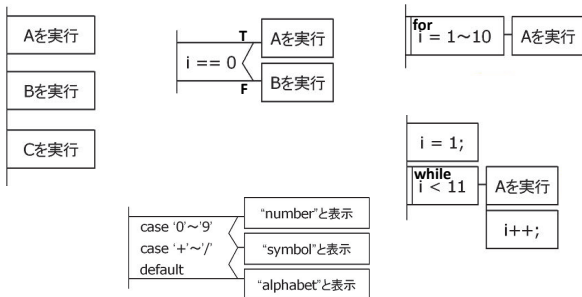
くり返し

- くり返しは、(命令型言語の場合)繰り返す文とくり返しを継続するか否かを判断する部分からなっている
 - コンピュータのプログラムと日常生活のプログラム(繰り返しが無い)との大きな乖離点である
- 機械語には、対応する(原始的な)機能はない
- Fortranでは、DOループという形で for ループが導入された
- Algol60では、while ... do ... が導入された

くり返し



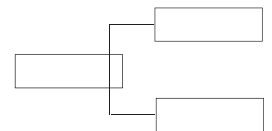
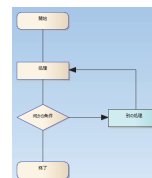
まとめ: PAD図の基本



<http://web.cc.yamaguchi-u.ac.jp/~fukuyo/prog2/302.html>

Break と n+1/2

- 配列中にある要素を検索するプログラムでは、発見したら、くり返しを終了したい
 - while の条件中に「未発見」という条件を入れる
 - 発見したら、ループを終了する → break
- n+1/2 と呼ばれるループがある



PAD図と設計

- PAD図は(流れ図と同様に)下流工程(プログラミング)の設計に向いている
 - 構造化プログラミングそのものであるため、流れ図より遥かによい。
 - 教育にもよい



<http://ipro.nikkeibp.co.jp/article/MAG/20100517/348034/751-develop8P-2>

- 上流設計では、UMLに準拠したアクティビティ図かステートチャートあたりがよい

