

### 本日の授業のテーマ

本日の授業のテーマは、以下のとおりです。IF 文と GO TO 文を学習します。教科書の P.25～P.49 です。

- (1) 制御文の役割と種類
- (2) IF 文
- (3) 論理式と関係演算子
- (4) GO TO 文

本日の授業のゴールは、以下のとおり。

- 制御文は、プログラム中の実行文をコントロールしていることが理解できる。
- IF 文の種類がわかる。
  - ブロック IF 文
  - 論理 IF 文
  - 算術 IF 文
- 関係演算子が理解できる
- 文番号と GO TO 文がわかる。

## 1 制御文の役割と種類

今まで学習したプログラムの実行文は、プログラムの記述順序にしたがい実行されます。これでは、不便なことがあります。計算結果に応じて、実行文を使い分けたい場合があります。このようなときに、制御文を使います。

### 1.1 分岐

例えば、教科書の最初の例題のような場合です。作成すべきプログラムは、

- 学生の受験番号と点数を読み込む
- 80 点以上ならば、合格と印刷
- 80 点以下ならば、不合格と印刷

です。80 点以上か以下か判断をして、それに応じた文が実行されます。これは、実行文が途中で分岐しますので、分岐と呼ばれる制御文です。

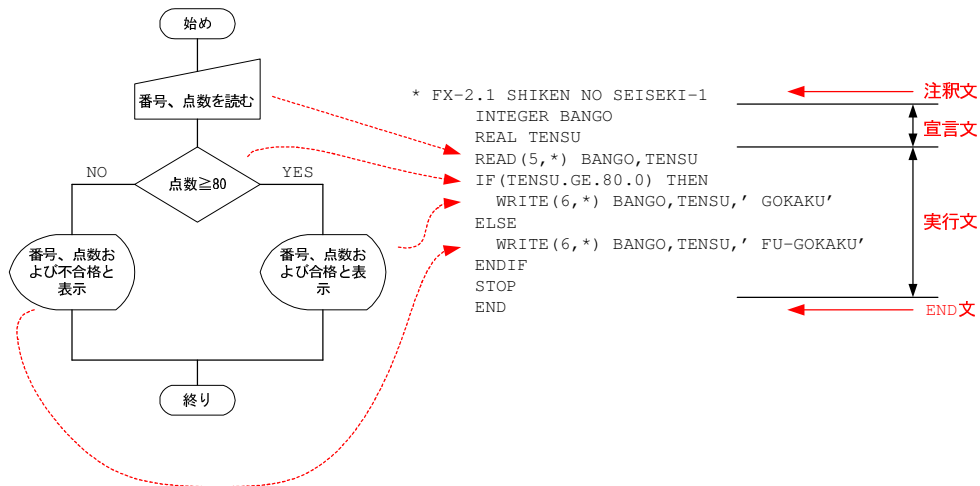


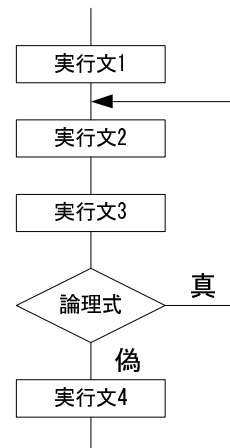
図 1 例題 2.1 の説明

制御文には、もう一種類あります。それは、同じ実行文を所定の回数実行させる場合に使います。この制御文は、通常ループと呼ばれます。

### 1.2 繰り返し(ループ)

ほとんどのプログラム言語には、この分岐とループの制御文があります。これが無い高級言語はないでしょう。この 2 つの制御文を理解すると、他のプログラム言語の習得も直ぐにできます。

ここで、ループまで説明すると混乱しますので、分岐の説明後、詳しく学習します。次の学習の DO 文です。



## 2 IF 文

### 2.1 ブロック IF 文

最もよく使われる IF 文です。通常は、これを使ってください。他の言語も、ほとんどこのブロック IF 文の形です。他の IF 文に比べて、プログラムが読みやすいし、論理 IF 文や算術 IF 文も、ブロック IF 文に置き換えることができます。

条件によって実行される文が複数あり、それが固まり(ブロック)になっているからブロック IF 文と呼ばれます。

ブロック IF 文は、IF で始まり、ENDIF で終わります。通常、IF と ENDIF の間の実行文は字下げ(インデント)をして、文の構造を分かりやすくします。

ブロック IF 文の使われ方のパターンは、次の 3 通りです。すべて覚えてください。

#### 2.1.1 パターン 1

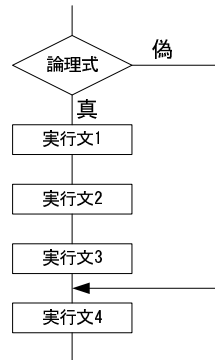
最初のパターンは、論理式が正しい場合のみ、文が実行されます。このパターンでは、論理式が正しいければ、

実行文 1→2→3

で実行されます。IF と ENDIF で囲まれた文がすべて実行された後、ブロック IF 文から抜け出します。そして、実行文 4 を実行します。

論理式が誤りならば、ブロック IF 文から抜け出して、ENDIF 文の次、実行文 4 を実行します。

```
IF (論理式) THEN
  実行文 1
  実行文 2
  実行文 3
ENDIF
実行文 4
```



#### 2.1.2 パターン 2

次のパターンは、論理式が正しい場合と誤りの場合で、異なる文が実行されます。論理式が正しいければ、

実行文 1→2→3

と実行されます。IF と ELSE で囲まれた文がすべて実行された後、ブロック IF 文から抜け出します。そして、実行文 7 を実行します。

論理式が誤りならば、

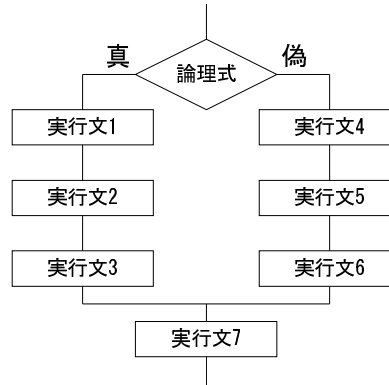
実行文 4→5→6

と実行されます。ELSE と ENDIF で囲まれた文がすべて実行された後、ブロック IF 文から抜け出します。そして、実行文 7 を実行します。

```

IF (論理式) THEN
  実行文 1
  実行文 2
  実行文 3
ELSE
  実行文 4
  実行文 5
  実行文 6
ENDIF
実行文 7

```



### 2.1.3 パターン3

最後のパターンは、論理式が多段階になっています。最初の論理式 1 が正しければ、

実行文 1→2

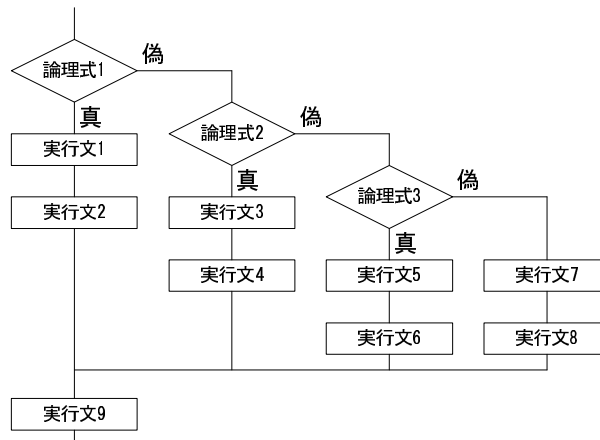
と実行されます。次の論理式を記述する ELSE IF まで囲まれた文がすべて実行された後、ブロック IF 文から抜け出します。そして、実行文 9 を実行します。

最初の論理式 2 が誤りならば、次の論理式 2 の文を実行します。後と、フローチャートの通りです。説明しなくても分かりますよね。

```

IF (論理式 1) THEN
  実行文 1
  実行文 2
ELSE IF (論理式 2) THEN
  実行文 3
  実行文 4
ELSE IF (論理式 3) THEN
  実行文 5
  実行文 6
ELSE
  実行文 7
  実行文 8
ENDIF
実行文 9

```



## 2.2 論理 IF 文

先ほどのブロック IF 文では、論理式が真の場合、複数の文を実行することが可能です。一方、ここで説明する論理 IF 文は、唯一つの文のみ実行します。文の構造が単純なので、1 文で簡潔に実行文が書けるときに使用します。論理 IF 文は、ブロック IF 文に書き換えることもできます。

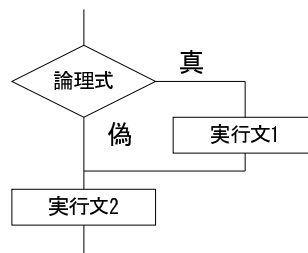
論理 IF 文の書き方とフローチャートは、下に示す通りです。普通、ブロック IF 文のように字下げ(インデント)はしません。

動作は単純で、論理式が正しければ、

実行文 1→2

と実行されます。誤りであれば、実行文 1 は実行されないで、実行文 2 が実行されます。

IF (論理式) 実行文 1  
実行文 2

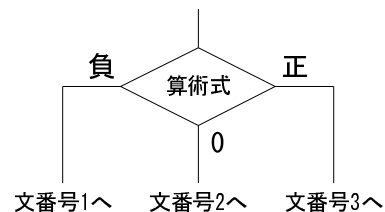


## 2.3 算術 IF 文

教科書に書いてあるとおり、プログラムが分かりにくくなりますので、使用しないほうが良いです。使っているプログラムがありますので、とりあえず説明だけしておきます。ブロック IF 文よりも先に、この算術 IF 文ができたので、仕様として残っています。ブロック IF 文が先に作られると、算術 IF 文は作られなかったかもしれません。ということで、これもブロック IF 文に書き換えることができます。

算術 IF 文の書き方とフローチャートは、下に示す通りです。今までの論理式の変わりに算術式が使われます。算術式の結果により、指定された文番号へ飛びます。文番号については、後で説明します。

IF (計算式) 文番号 1, 文番号 2, 文番号 3



### 3 論理式と関係演算子

IF 文を使った実行の制御は、論理式の結果に依存します。今まで、論理式と言うものを説明しませんでした。ここで詳しく説明します。

算術式の計算結果は、数字です。それに対して、論理式の計算結果は、

- 論理式が正しい場合 真 (TRUE)
- 論理式が誤りの場合 偽 (FALSE)

となります。例えば、以下の論理式は、

1<5	真
6>8	偽

です。このように、計算式が正しいか否かによって、真か偽の判断をするのが論理式です。

ここで示した論理式は、数値の演算ですが、それ以外に論理演算や文字型の演算があります。これらについては、FORTRAN の後半で学習します。皆さんは、数値型の演算を使うことが多いはず。まずは、数値型の論理演算に慣れてください。

数値型の論理演算は、数字の大小の比較だけです。だから、数学の不等号の演算だけです。その不等号が正しいか、否かのみを判断します。困ったことに、FORTRAN77 では、数学の不等号の記号 "<" や ">" が使えません。そこで、以下の表 1 のように文字で表すことになっています。

表 1 論理演算子。A と B との比較

数学	FORTRAN	演算子	語源
$A < B$	A.LT.B	.LT.	Less Than
$A \leq B$	A.LE.B	.LE.	Less than or Equal
$A = B$	A.EQ.B	.EQ.	Equal
$A \neq B$	A.NE.B	.NE.	Not Equal to
$A > B$	A.GT.B	.GT.	Greater Than
$A \geq B$	A.GE.B	.GE.	Greater than or Equal

論理演算子と算術演算子と比較してみましょう。以下のようになります。演算子の意味が良くわかるでしょう。これらは、2 つの変数から 1 つの値 (数値や論理) を計算しますので、2 項演算子とも呼ばれます。

- 算術演算子 2 個の数値から 1 個の数値を計算
- 論理演算子 2 個の数値から 1 個の論理 (真 or 偽) を計算

## 4 文番号と GO TO 文

IF 文は、GO TO 文と共に使われることがしばしばあります。ついでに、GO TO 文も学習しましょう。GO TO 文は実行文です。GO TO 文があれば、それが示す分番号に強制的に飛びます。

### 4.1 文番号

実行文に目印をつけるものです。整数の番号を書きます。書き方は、図 1 の通りです。第 1 桁目から第 5 桁目まで、任意の 1 個の整数を書くことができます。また、プログラムに沿って、順序もなく、全く任意です。ただし、同じ文番号を 2 個書くことはできません。コンパイルエラーを起こします。

算術 IF 文では、制御の移る文を示します。また、次に述べる GO TO 文で制御の移る文を示すときにも使われます。

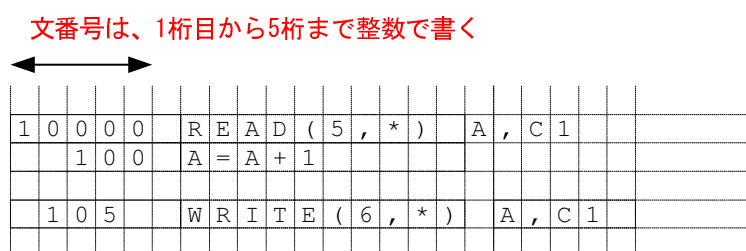


図 1 文番号の書き方

### 4.2 GO TO 文

実行文にこの文があると、それが示している分番号に強制的に移動します。書き方は、以下の通りです。

GO TO 文番号

### 4.3 IF 文と GO TO 文の組み合わせ

動作は説明しません。ブロック IF 文や算術 IF 文での使用例を示します。このプログラムの動作を考えてください。

INTEGER N, I, SUM	INTEGER N, I, SUM
SUM=0	SUM=0
I=0	I=0
READ (5, *) N	READ (5, *) N
100 I=I+1	100 I=I+1
IF (I.LE.N) THEN	SUM=SUM+I
SUM=SUM+I	IF (I.LT.N) GO TO 100
GO TO 100	WRITE (6, *) SUM
ELSE	STOP
WRITE (6, *) SUM	
ENDIF	END
STOP	
END	