

アルゴリズム
第4回授業
“基本制御構造 繰り返し”
(教科書 Page 24-33)

山口雅樹 (CISSP)

本日の進め方

- ・ 基本制御構造その2 繰り返し

1-5 基本制御構造 繰り返し (WIKIより)

ループとは、特定の条件下において特定の処理を繰り返すこと、あるいはそのように作られた制御構造のことを言う。日本語の名詞として「繰り返し」とも。特定の条件が成立している限り、特定の処理を繰り返し何度でも実行する。逆に言えば、条件が成立しなくなったときに、処理を中止する。

次のページのサンプルプログラムでは、CNT(カウント) が10未満の場合に、繰り返し処理を行うプログラムを動かしてみます。

プログラムの宣言部と処理部、フローチャート

○プログラム名: Loop10Times() /* 教科書 25ページ 繰り返し処理 */

○整数型: CNT

●CNT ← 0 **初期化**

■ CNT < 10

●表示処理 (CNT)

●CNT ← CNT + 1

□

●表示処理(CNT)

デバッグメッセージ / 出力:

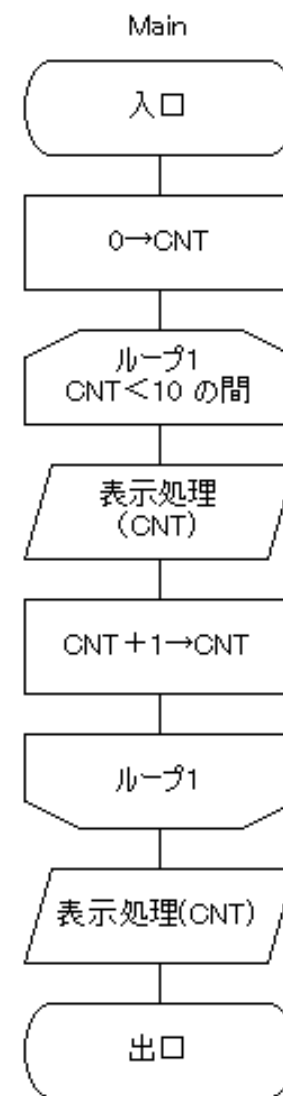
実行..

0
1
2
3
4
5
6
7
8
9
10

ループ処理

(CNTが10未満の場合に繰り返す)

前判定となる



プログラムの宣言部と処理部 変数の流れ（前判定）

	判定	処理	CNTの値
1回目判定	$CNT=0 < 10$	表示処理	$CNT=0$
2回目判定	$1 < 10$	表示処理	$CNT=1$
3回目判定	$2 < 10$	表示処理	$CNT=2$
4回目判定	$3 < 10$	表示処理	$CNT=3$
5回目判定	$4 < 10$	表示処理	$CNT=4$
6回目判定	$5 < 10$	表示処理	$CNT=5$
7回目判定	$6 < 10$	表示処理	$CNT=6$
8回目判定	$7 < 10$	表示処理	$CNT=7$
9回目判定	$8 < 10$	表示処理	$CNT=8$
10回目判定	$9 < 10$	表示処理	$CNT=9$
11回目判定	$10 < 10$ ★ここでループを抜ける	表示処理	$CNT=10$

前判定と後判定

○プログラム名: Loop10Times() /* 教科書 27ページ 繰り返し処理 */
○整数型: CNT

●CNT ← 10

■ CNT < 10

●表示処理 (CNT)

●CNT ← CNT + 1

□

●表示処理(CNT)

前判定

デバッグメッセージ / 出力:

実行..

10

○プログラム名: Loop10Times() /* 教科書 27ページ 繰り返し処理 */
○整数型: CNT

●CNT ← 10

■

●表示処理 (CNT)

●CNT ← CNT + 1

□ CNT < 10

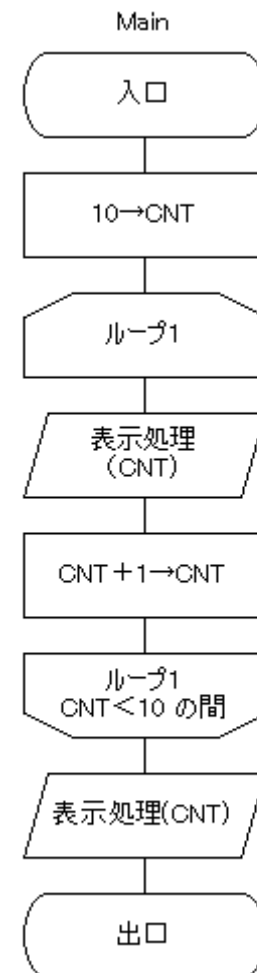
●表示処理(CNT) 後判定

デバッグメッセージ / 出力:

実行..

10

11



前判定だと 一度も実行されないケースがあるが、
後判定だと、必ず一度は実行される。

For型ループ

```
○プログラム名: Loop10Times() /* 教科書 28ページ 繰り返し処理 */  
○整数型: LoopCount  
    初期値    継続条件    増分  
■ LoopCount : 0, LoopCount < 10, 1  
| ●表示処理 (LoopCount) 処理について  
□
```

デバッグメッセージ / 出力:

実行..

0
1
2
3
4
5
6
7
8
9

★継続条件 LoopCountが10になったらループを抜けます

ここでTraining
(コピー利用)

教科書 29ページの解答例

```
○プログラム名: Loop10Times() /* 教科書 29ページ 繰り返し処理 */  
○整数型: Cnt, Total  
● Total ← 0  
  
■ Cnt: 0, Cnt < 101, 1  
| ● Total ← Total + Cnt  
□  
  
●表示処理(Total)
```

デバッグメッセージ / 出力:

実行..

5050

1 から 10 までの整数の合計を求める

○プログラム名: Loop10Times() /* 教科書 30ページ 合計 */

○整数型: Cnt, Total

● Total ← 0

● Total ← 1 + Total

● Total ← 2 + Total

● Total ← 3 + Total

● Total ← 4 + Total

● Total ← 5 + Total

● Total ← 6 + Total

● Total ← 7 + Total

● Total ← 8 + Total

● Total ← 9 + Total

● Total ← 10 + Total |

●表示処理(Total)

←同じ命令を繰り返して記述しなければならない。

デバッグメッセージ / 出力:

実行..

55

1 から 10 までの整数の合計を求める

(効率よくかけると、Cntを100とか1000に簡単に変更できる)

○プログラム名: Loop10Times() /* 教科書 31ページ 繰り返し */

○整数型: Cnt, Total

● Cnt ← 0

● Total ← 0

■ Cnt ≤ 10

| ● Total ← Total + Cnt

| ● Cnt ← Cnt + 1

□

● 表示処理(Total)

デバッグメッセージ / 出力:

実行..

プログラムの宣言部と処理部 変数の流れ

	判定	処理	CNTの値	TOTALの値
1回目判定	$CNT=0 \leq 10$	Total=Total+Cnt	CNT=0	0
2回目判定	$1 \leq 10$	Total=Total+Cnt	CNT=1	1
3回目判定	$2 \leq 10$	Total=Total+Cnt	CNT=2	3
4回目判定	$3 \leq 10$	Total=Total+Cnt	CNT=3	6
5回目判定	$4 \leq 10$	Total=Total+Cnt	CNT=4	10
6回目判定	$5 \leq 10$	Total=Total+Cnt	CNT=5	15
7回目判定	$6 \leq 10$	Total=Total+Cnt	CNT=6	21
8回目判定	$7 \leq 10$	Total=Total+Cnt	CNT=7	28
9回目判定	$8 \leq 10$	Total=Total+Cnt	CNT=8	36
10回目判定	$9 \leq 10$	Total=Total+Cnt	CNT=9	45
11回目判定	$10 \leq 10$	Total=Total+Cnt	CNT=10	55
12回目判定	$11 \leq 10$ ★ここでループを抜ける	なし		55

○プログラム名: Loop10Times() /* 教科書 31ページ 繰り返し */
/* こうすると処理を 1 回減らせる */

○整数型: Cnt, Total

● Cnt ← 2

● Total ← 1

■ Cnt ≤ 10

| ● Total ← Total + Cnt

| ● Cnt ← Cnt + 1

□

●表示処理(Total)

デバッグメッセージ / 出力:

実行..

1 から100のうちの偶数の合計を求める

○プログラム名: EvenGokei() /* 教科書 32ページ 繰り返し */

○整数型: Cnt, Total

● Cnt ← 2

● Total ← 0

■ Cnt ≤ 100

| ● Total ← Total + Cnt

| ● Cnt ← Cnt + 2

□

●表示処理(Total)

デバッグメッセージ / 出力:

実行..

2550

引き算の繰り返しで余りを求める (負の数は考えない)

○プログラム名: Divide() /* 教科書 33ページ 繰り返し */

○整数型: A, B, Syo, Amari

● A ← 7

● B ← 3

● Syo ← 0

■ A ≥ B

| ● A ← A - B

| ● Syo ← Syo + 1

□

● Amari ← A

● 表示処理(Amari)

デバッグメッセージ / 出力:

実行..

初期化について

変数の内容は不定（定まっていない）ことが多いので、
初期化しておくことが必要
特に、ループ処理の回数（Count）などは初期化することが重要