

繰り返し (不定回反復)

情報処理演習
(テキスト: 第8章8.3節)

今日の内容

1. 不定回反復とその必要性
2. while 文
3. do-while 文
4. ループの途中での実行の制御
5. 今日の練習問題

1. 不定回反復とその必要性

■不定回反復の必要性

- 繰り返しの回数がまえもって分からない場合は, for 文で記述するのは不適
- 論理式が正しい間, 繰り返す.

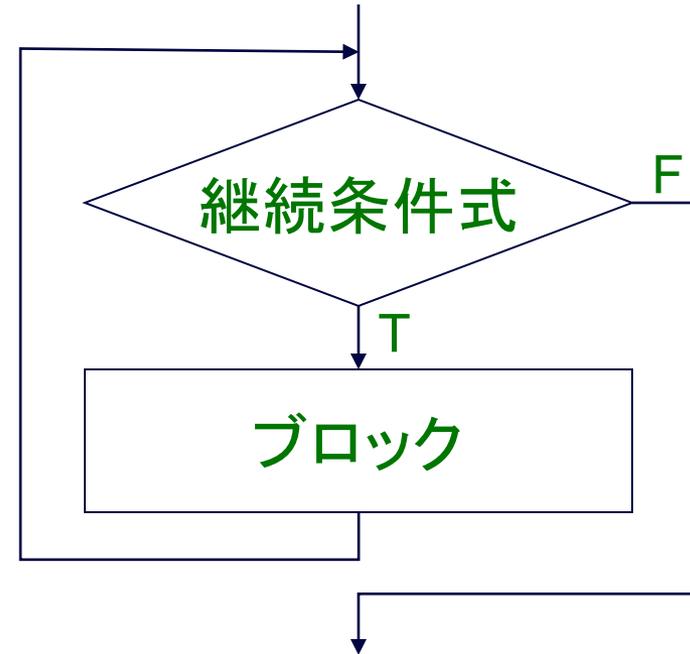
```
while( 論理式 ){  
    文1  
    文2  
    ⋮  
}
```

```
do{  
    文1  
    文2  
    ⋮  
} while( 論理式 );
```

2. while 文

(テキスト94ページ)

```
while (継続条件式) {  
    ブロック  
}
```



- 継続条件式が成り立っている間、ブロックを実行する
- 条件が成り立たなければ、ループは終了する。
- ブロックの実行前に条件を評価するので、ブロックが1回も実行されない場合もある。

while 文の使用例

```
#include <stdio.h>
/* 0が入力されるまで入力値を加算していく. */
int main(int argc, const char * argv[]) {
    int sum, x;
    sum = 0;
    printf("data? ");
    scanf("%d", &x);
    while(x != 0){ /* 0が入力されるまで繰り返される. */
        sum += x;
        printf("data?");
        scanf("%d", &x);
    }
    printf("sum = %d", sum);
    return 0;
}
```

sum = sum + x を意味する

sum += i とは

(テキスト54ページ)

□ は sum という変数の中身を表す。

ループに入る前 $sum = 0$

□
0

i が1で実行 $sum = sum + 1$ ← $sum_1 = sum_0 + 1$

□ $0+1$ ← □ 0 + 1

i が2で実行 $sum = sum + 2$ ← $sum_2 = sum_1 + 2$

□ $0+1+2$ ← □ $0+1$ + 2

⋮

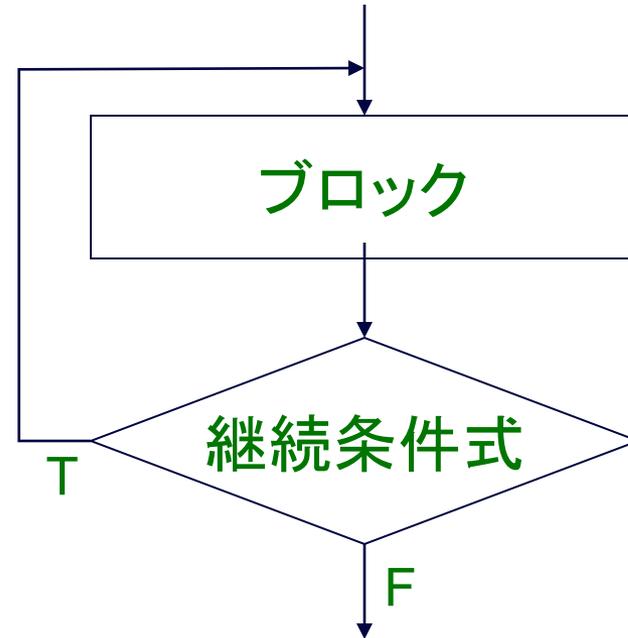
i がnで実行 $sum = sum + n$ ← $sum_n = sum_{n-1} + n$

□ $0+1+2+ \dots +n$ ← □ $0+1+2+ \dots +(n-1)$ + n

3. do - while 文

(テキスト94ページ)

```
do {  
    ブロック  
} while (継続条件式);
```



- 継続条件式が成り立っている間, ブロックを実行する.
- 条件が成り立たなければ, ループは終了する.
- ブロックを実行した後に条件を評価するので, ブロックを必ず1回は実行する.

do - while 文の使用例

```
#include <stdio.h>
/* 0が入力されるまで入力値を加算していく. */
int main(int argc, const char * argv[]) {
    int sum, x;
    sum = 0;
    do{
        printf("data?");
        scanf("%d", &x);
        sum += x;
    } while(x != 0);
    printf("sum = %d", sum);
    return 0;
}
```

1から10までの合計

- 3種類のループはどれを使っても構わない。

for ループ

```
int sum = 0;
for (int k = 1; k <= 10; k++) {
    sum += k;
}
```

変数 k はループ変数
(ループの回数を数えるための変数)

while ループ

```
int sum = 0;
int k = 1;
while (k <= 10) {
    sum += k;
    k++;
}
```

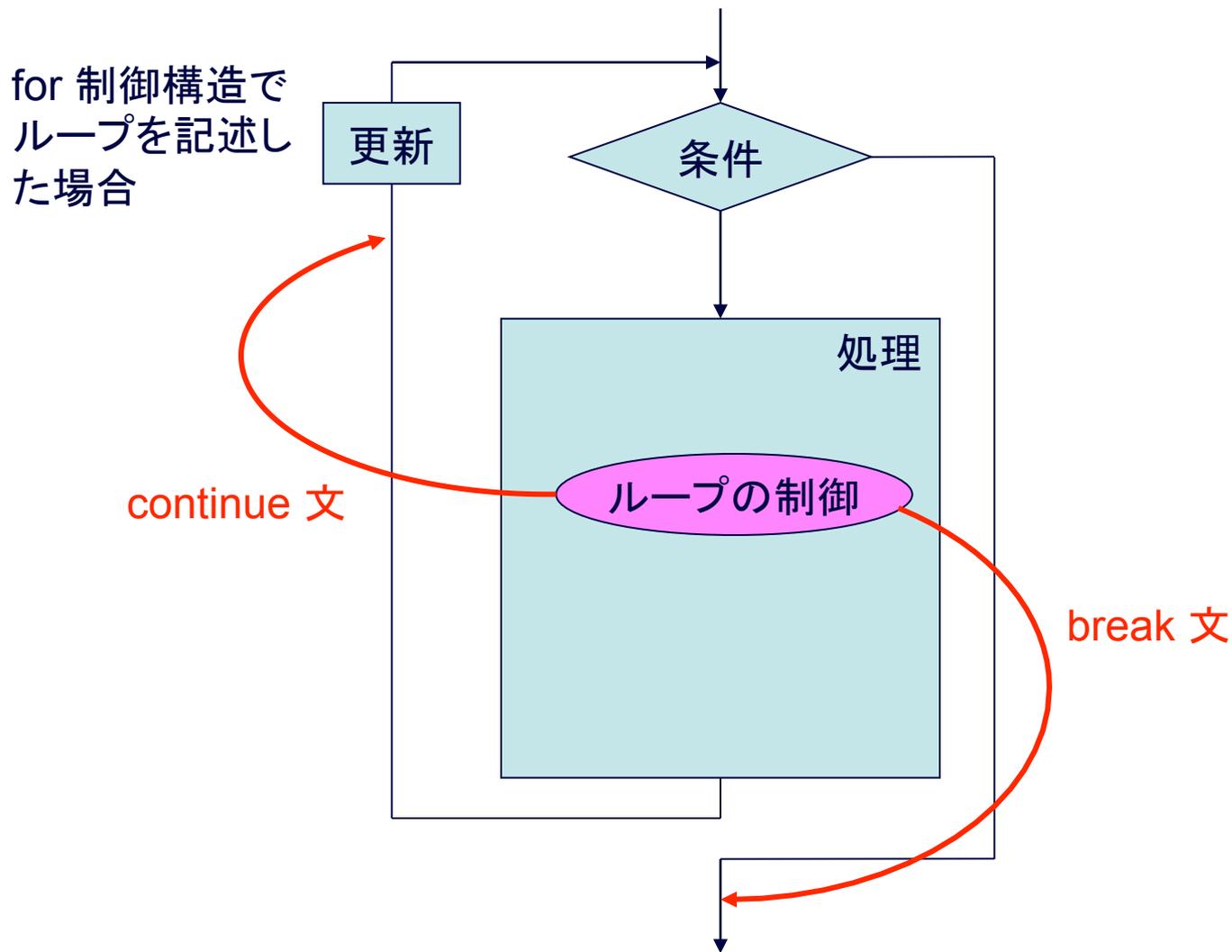
do-while ループ

```
int sum = 0;
int k = 1;
do {
    sum += k;
    k++;
} while (k <= 10)
```

4. ループ途中での実行の制御

- ループの途中で、繰り返している処理を...
 - 終了させたい.
(ループから抜ける.)
 - ◆ `break` 文
 - ◆ ループのブロック, `switch` 制御構造のブロックで使用可能
 - ループ内の残りの処理を省きたい.
(ループは続行)
 - ◆ `continue` 文
 - ◆ ループのブロックで使用可能

2つの制御文の流れ



5. 今日の練習問題

Level	問題
B	ニュートン法による平方根の近似プログラムを作成せよ(例8.3.2参照).
B+	ニュートン法による三乗根の近似プログラムを作成せよ.
A	8.5練習問題10「ひとりロシアンルーレット」を作成せよ.