

Programming I 0x0a



ループの応用(2010.06.07)

塩澤秀和 <http://vilab.org>

while文 (復習)

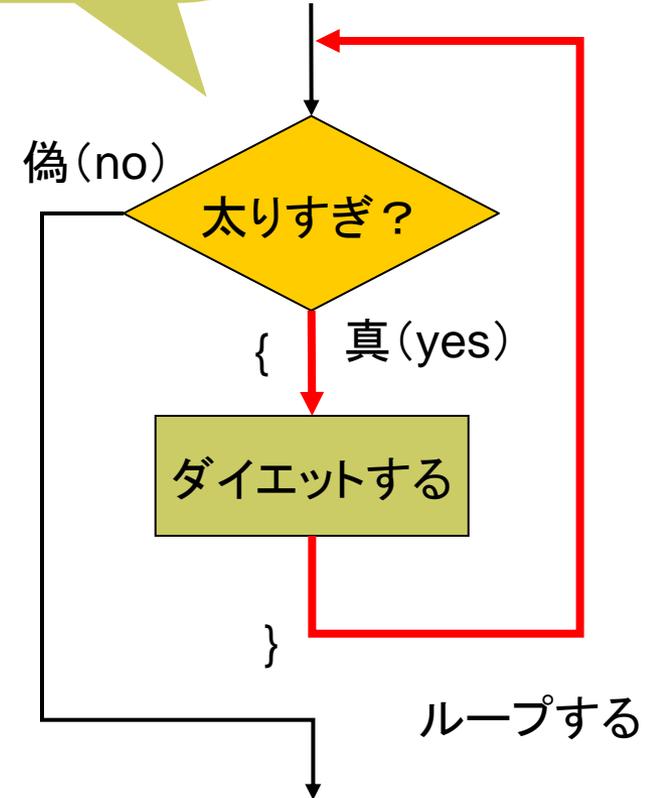
毎回最初に条件
をチェックする

□ 繰り返し (ループ、反復)

- 継続する条件が成り立つ間、決められた処理を何回も繰り返す
- “while” = 「～である間」
- 条件が不成立なら、処理を飛ばす

□ while文の構文

```
while (条件式) {
    条件が“真”の間、繰り返す処理
    ...
}
```



whileのフローチャート

- **注意:** 条件式は、終わる条件でなく、“**続ける条件**”を書く

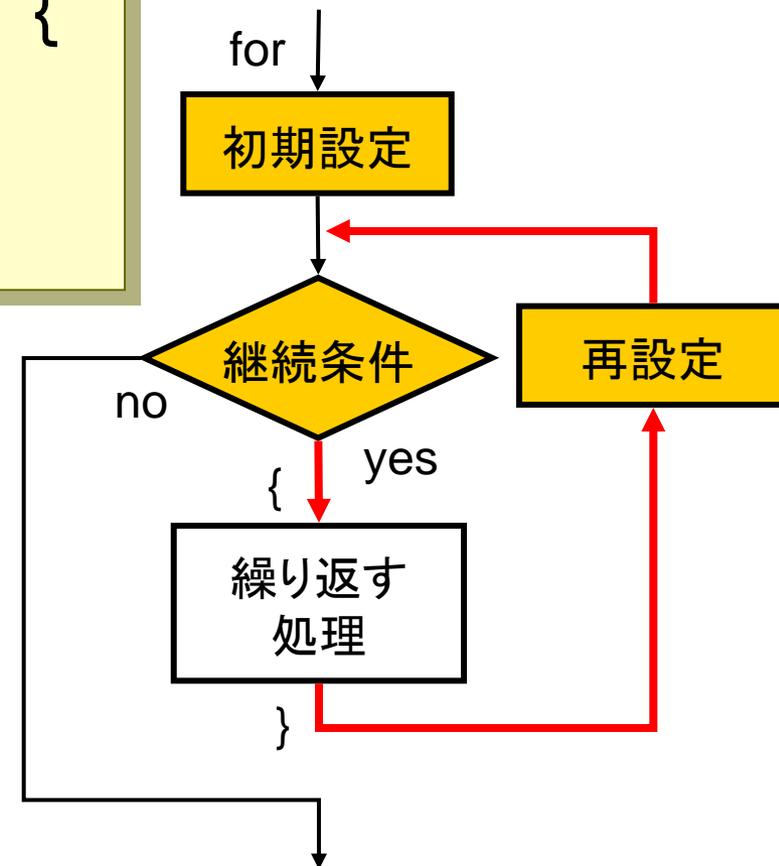
for文(復習)

よくあるループのパターンを
簡潔に書くための構文

□ for文の構文

```
for (初期設定; 継続条件; 再設定) {
    条件が“真”の間、繰り返す処理
    ...
}
```

□ while文で書き換えてみよう



forのフローチャート

for文のパターン

□ 例1

```
for (i = 0; i < 10; i++) {  
    printf("%d ", i);  
}
```

□ 例2

```
for (i = 1; i <= 10; i++) {  
    printf("%d ", i);  
}
```

□ 例3

```
for (i = 0; i < 100; i += 5) {  
    printf("%d ", i);  
}
```

□ 例4

```
for (i = 99; i >= 0; i--) {  
    printf("%d ", i);  
}
```

□ 例5

```
for (i = n; i <= m; i++) {  
    printf("%d ", i);  
}
```

□ 例6 (誤差が出るので推奨しないが)

```
for (x = 0.0; x < 1.0; x += 0.1) {  
    printf("%f ", x);  
}
```

2重ループのプログラム

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int h; /* 時 0~23 */
```

```
    int m; /* 分 0~59 */
```

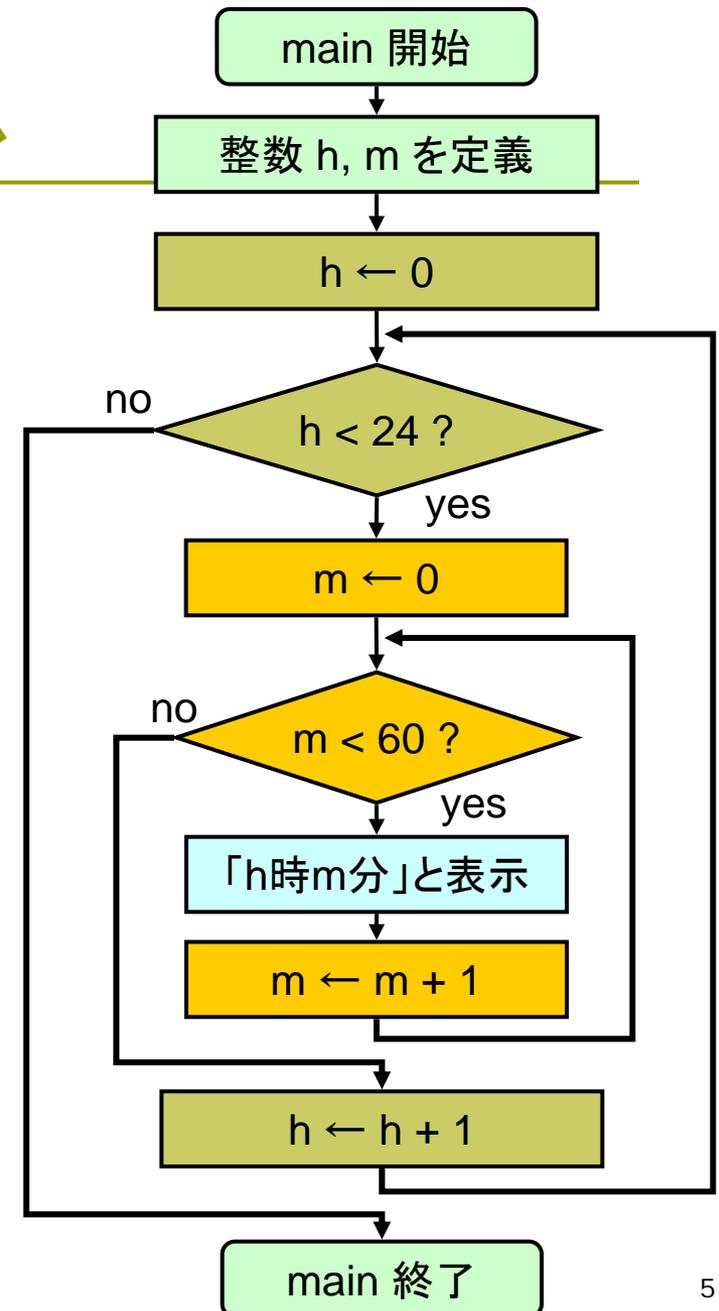
```
    for (h = 0; h < 24; h++) {  
        for (m = 0; m < 60; m++) {  
            printf("%d時%d分¥n", h, m);  
        }  
    }
```

```
    getchar(); /* [Enter]キーを待つ */
```

```
}
```

```
return 0;
```

```
}
```

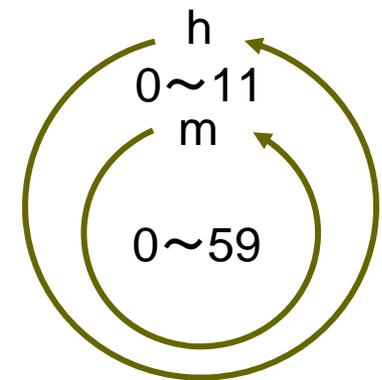


多重ループ

□ ループのなかでループする

```
for (h = 0; h < 24; h++) {
    for (m = 0; m < 60; m++) {
        printf("%02d時%02d分¥n", h, m);
    }
}
```

h が1増えるごとに
m は60回ループする



順々に全部表示

	h=0	h=1	h=2	...	h=22	h=23
m=0	00時00分	01時00分	02時00分	...	22時00分	23時00分
m=1	00時01分	01時01分	02時01分	...	22時01分	23時01分
m=2	00時02分	01時01分	02時02分	...	22時02分	23時02分
...
m=59	00時59分	01時59分	02時02分	...	22時59分	23時59分

フローチャートを書いてみよう

九九の表示

```
#include <stdio.h>

int main(void)
{
    int i, j;

    i = 1;
    while (i <= 9) {
        printf("%dの段\n", i);
        j = 1;
        while (j <= 9) {
            printf("%d×%d = %d\n",
                i, j, i * j);
            j++;
        }
        printf("\n");
        i++;
    }
    return 0;
}
```

無限ループ (p.57,62)

□ 無限 (永久) ループ

- わざと無限ループを作る方法

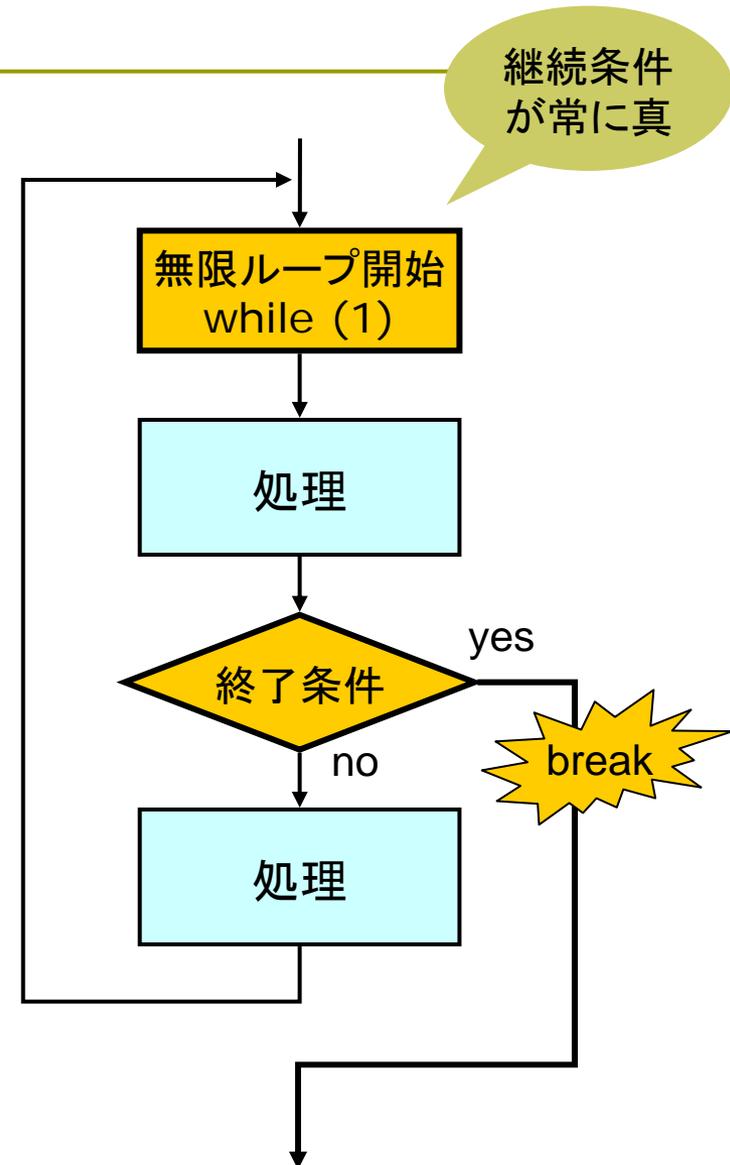
```
while (1) {  
    ...  
}
```

```
for (;;) {  
    ...  
}
```

- 本当に終わらないループを作ってしまうとまずいので...

□ break文 (p.58)

- ループを中止して外に抜ける
if (終了条件) break;
- ループの途中で終了 (継続) を判定したいときに使う



フローチャートを書いてみよう

無限ループの例

```
/* 入力値によってループを終わらせる */
#include <stdio.h>

int main(void)
{
    int n;
    for (;;) {
        printf("続けるなら1を入力:");
        scanf("%d", &n);

        if (n != 1)
            break;

        printf("%dなので継続\n", n);
    }
    printf("%dなので終了\n", n);
    return 0;
}
```

無限ループ

ループ脱出

その他の制御構造

□ do-while文 (p.59)

- ループの処理の最後に、継続条件を判定する
- 判定が最後なので、最低1回はループの中を通る
- うまく使えば分かりやすいが、必要ないときには使わない

□ switch-case文 (p.65)

- 整数値での多重分岐を“見やすく”書くための構文
- if ~ else if ~...~ else で書いても本質的な違いはない

□ goto文 (p.68)

- **使わない!** (適切な理由がある場合を除く)
- 1968年の有名な論文『**GO TO文は有害だと考えられる**』

エドガー・
ダイクストラ

do-whileとswitch-case

```
/* do-whileの例(10aの問題) */
#include <stdio.h>

int main(void)
{
    double pi = 3.1416;
    double r, S;

    do {
        printf("半径 r=");
        scanf("%lf", &r);
    } while (r < 0.0);

    S = pi * r * r;
    printf("面積 S=%f\n", S);
    return 0;
}
```

r が負の間
やり直す

```
#include <stdio.h>
int main(void)
{
    int n;
    printf("番号? ");
    scanf("%d", &n);
    switch (n) {
        case 1:
            printf("1番を選択\n");
            break;
        case 2:
            printf("2番を選択\n");
            break;
        default:
            printf("1,2以外を選択\n");
            break;
    }
    return 0;
}
```

演習問題

- 10a. 「do-whileの例」のフローチャートを描きなさい。
- 10b. 「九九の表示」のプログラムをfor文を使うように書き換え、さらに九九の“表”の形式で表示するようにしなさい。
- 桁数をそろえるために、printfの書式指定でたとえば「%2d」とする。
- 10c. $2+3+5=10$ のように、合計が10になる自然数3つの組み合わせをすべて調べて表示するプログラムを作成しなさい。
- 3重ループ(または2重ループ)を使い、その中のif文で判定する。
- 10d. キーボードから次々に自然数を読み込んで合計に加え、負の数を読み込んだら、自然数だけの合計を表示して終了するプログラムを作成しなさい。
- 無限ループを用いればよい。最後の負の数は結果に加えないこと。
- 次回までの課題: リスト5-1, 5-2を入力して教科書を予習