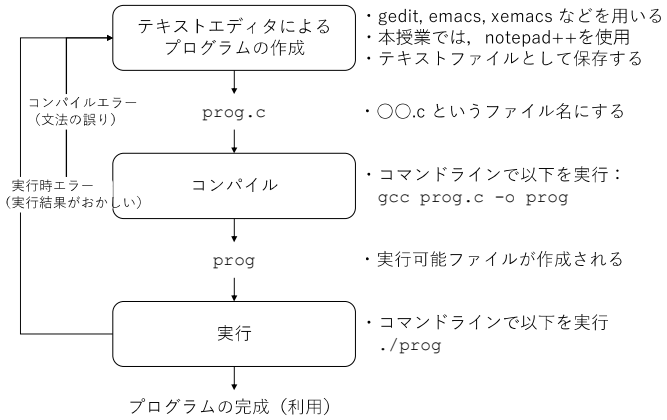
情報処理演習
(2)条件分岐

知能システム学　准教授
万 偉偉（ワン ウェイウェイ）

復習▪プログラム作成▪検証の流れ



復習▪文法とプログラムの構造

```
/* これはプログラムの一例ですよ */
#include <stdio.h>
int main(void) {
    int seisu;

    seisu = 5;
    printf("seisuの値は%dです¥n",seisu);
    return 0;
}
```

行頭が#はプリプロセッサ（特別扱い）

コメント
文字列
識別子
予約語

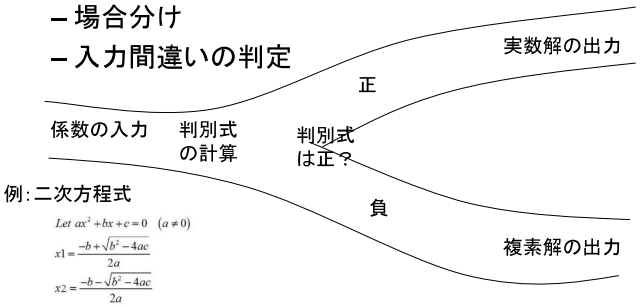
識別子は，アルファベットか数字で作る．
（ただし先頭はアルファベットのみ）

復習▪前回学んだ内容のまとめ

- C言語のプログラムの基本構造
  - #include <stdio.h>, int main(void), return
- 変数
  - int, double, char
- 標準出力関数 printfの使い方
  - printf("hoge%d", a);
- C言語の文法
  - 文，識別子，文字列，コメント

条件分岐とは

- 数値の条件（大小，等しい，など）によってプログラムの流れを変える
  - 場合分け
  - 入力間違いの判定



プログラムで分岐するには

```
double a, b, c, d, ans1, ans2;
d = b * b – 4 * a * c;
```

```
ans1 = (-b + sqrt(d)) / (2 * a);
ans2 = (-b – sqrt(d)) / (2 * a);
```

複素解の計算

- プログラムは小説のように一本の流れを持つので，並列に書けない

if 文

( )	括弧（かっこ）、丸括弧、小括弧、挿入符、パーレン（paren、、parenthesis）
[ ]	ブラケット（bracket）、角括弧、大括弧、箱パーレン、スクエアブラケット
{ }	ブレース（brace）、ブレイス、中括弧、波括弧、こうもり、カーリーブラケット

```
if(d > 0) {
    ans1 = (-b + sqrt(d)) / (2 * a);
    ans2 = (-b - sqrt(d)) / (2 * a);
    printf(….);                          分岐先1
}
else {
    /* 複素解の計算▪表示 */              分岐先2
}
```

if文のパターン

① if 単独

```
if( 条件式 ) {
    条件式が真の時、実行する処理
}
抜けた後の処理
```

② if else

```
if( 条件式 ){
    条件式が真の時、実行する処理
}
else{
    条件式が偽の時、実行する処理
}
抜けた後の処理
```

③ if-else if

```
if( 条件式1 ){
    条件式1 が真の時、実行する処理
}
else if( 条件式2 ){
    条件式2 が真の時、実行する処理
}
else{
    いずれでもない時、実行する処理
}
抜けた後の処理
```

比較演算子について

- “=“ は等号ではない！！
  “=“ は代入演算子（右辺を計算して左辺に代入）
  比較演算子は “==“ である.
- “<=“, “>=“ について
  – 小さいか等しい、大きいか等しい
  “=<“, “=>” はダメ
- 「等しくない」を判定するには
  “!=“ を使う（”≠” の意味）

ブロックについて

- { … } のことをブロックと呼ぶ
  – { … } は全体として「1行」（単文）として扱われる
  – { … } の最初では，変数の定義が出来る
  （ここで定義した変数はブロックの外では無効）
- if (…) { … } の 部分もブロック
  If (a == 0)
  printf(“%d¥n”, a);
  という風にブロックを使わない書き方も許される
  （1行であれば）
- ブロックは何重にでも出来る（入れ子構造）

「かつ」「または」

- 「かつ」(and)は “&&” アンパサンド二つ
  – if (a == 0 && b == 0) のように
- 「または」(or)は “||” パイプ二つ
  – if (a < 0 || a > 10) のように
- 括弧も使えます
  – if((a == 0 || b == 0) && c == 0) のように
- if (! (a == 0)) は if (a != 0) と同じ

本日学んだ内容

- 条件分岐 if , else, else if
- 比較演算子
  – 等しい？ 以下？以上？ 等しくない？
- 論理演算子
  – かつ または ノット
  – ブロック { … }

プログラムを綺麗に書きましょう

- 綺麗：他人に見たら気持ちいい，読みやすい
- インデントと括弧を使うこと
  – 階層構造を持つプログラムに対して，字下げをして見やすくする
  （方法）
  - 同じ階層の命令は字下げの位置をそろえる
  - 階層が深くなるごとに字下げの量を増やす
  （効果）
  - エラーが発見しやすくなる
  - 他人にも理解してもらいやすくなる

プログラム例

```
int main( void )
{
1つ分 ←→ if (a == 0){
2つ分 ←→←→ printf("解は%lfです。", -c/b);
    }
    else{
        d = b*b - 4*a*c;
        if (d > 0) {                    /*判別式が正*/
            …
        }
        else if (d == 0) {              /*判別式が0*/
            …
        }
        else{                           /*判別式が負*/
            …
        }
    }
}
```

同じ階層のものはインデント位置を揃える

インデントの空け方

- スペース4つ分
- 「Tab」を使わないように
- .cファイルの編集なら，多くのエディタで自動でインデント付けしてくれる

今後のレポート提出

必ずインデントを付けてください

インデントのついていない
プログラムを載せたレポートは
再提出対象