入門講習会 構造体

1. 構造体

構造体とは, いくつかの変数をまとめた新しい型である。

意味的なまとまりとして新しい型をつくれば, プログラムの見通しがよくなることがある。

構造体は以下の書式で宣言する。

|  |
| --- |
| struct 構造体名 {  メンバ型 メンバ変数名;  メンバ型 メンバ変数名;  メンバ型 メンバ変数名;  ...  }; |

メンバ型 メンバ変数名;

の部分は, 通常の変数の宣言と同じように行う。構造体宣言の}の後にはセミコロンをつけることに注意しよう。

(例)

人間の名前と現在いる座標(二次元座標)をメンバとする構造体Humanは以下のように書ける。

struct Human {

char name[1000];

int x;

int y;

};

さて構造体の宣言によって新しい型が作られた。しかしあくまで作られたのは型であって, 実態はまだできていない。構造体を型とした変数を宣言するには, 次のように書く。

|  |
| --- |
| struct 構造体名 変数名; |

構造体によって作られた変数は, その構造体で定義されたメンバを要素として持っている。そのメンバを利用するには, 次のように書く。

|  |
| --- |
| 変数名.メンバ変数名 |

この記法によって, メンバに値を代入したり, メンバを使って何か計算をしたりできる。

(例)

さきほど作った構造体Humanを型とした変数aを宣言するには次のように書く。

struct Human a;

名前と座標を入力させるためには, 次のように書けばよいだろう。

scanf("%s %d %d", a.name, &a.x, &a.y);

aの原点からの距離(ユークリッド距離)を求めるためには, 次のように書けばよいだろう。

double d = sqrt(a.x\*a.x+a.y\*a.y);

※sqrt(x)とはxの平方根を求めることができる関数。math.hをincludeしなければならない。

1. typedef

型に同義語を定義する命令がある。それがtypedefである。長い型を短い名前に変えたいときに便利。次のように用いる。

|  |
| --- |
| typedef 型名 新しい型名; |

ここで,

型名 新しい型名

は, 変数宣言と全く同じように書くことを意識しよう(そうしないと, ポインタ型をtypedefするときに意味が分からなくなる)。

(例)

typedef long long ll;

と書けば, long long型をllとして書けるようになる。次のようにlong long型を宣言することができ, 名前が短くなってタイプ数が減らせる。

ll x;

(例)

typedef struct Human Hm;

と書けば, struct Human型をHmとして書けるようになる。次のようにstruct Human型を宣言することがきるようになる。

Hm a;

構造体型に関しては, 次のように宣言と同時にtypedefを書くことができる。

typedef struct Human {

char name[1000];

int x;

int y;

} Hm;