プログラミング演習
～構造体～

1　目的

C 言語における構造体の使い方を理解し，バンディット問題 [1] 等を例題にしながら実践し理解を深める．

2　製作対象 構造体の基礎

配列の演習 [2] にて，多次元配列を用いたバンディットスコアの取得を行ったのを覚えているだろうか?具体的には，

- double reward[2][10];
  - reward[1][]:選択した腕番号
  - reward[0][]:その時の報酬

として記憶していた．問題は，報酬の型 (double) と腕番号 (int) の型が異なり，腕番号の型を報酬の型に変換して記憶していた．

このように配列では，一種類の型を複数個記憶することはできるが，複数種類の型を記憶することはできない．これを実現するのが構造体である．int 型や double 型はシステムが用意している型であるが，構造体はこれら用意してある型を組み合わせて自分独自の型を作るものである．そのため，構造体を使う上では，

- 型の宣言
- 変数の宣言
- 変数の使用

が必要となる．システムが用意している従来の変数では型の宣言はない (システムが行っている)．変数の宣言は従来の変数と類似しているが，変数の使用方法は少々異なっている．次に 2.2 プログラム中の具体例を見ながら，それぞれについて説明する．

2.2 プログラム中の 9 行目から 12 行目が型の宣言になっている．このように，

- structure 自作の型の名前 {
- 自作の型の内容 (従来の変数の型や他の構造体の組み合わせで作る)

- };

となっている．このようにすることで，”自作の型の名前”(今回は score) という型を作ることが出来る．

自作の型の使用方法は他の型と類似している．2.2 プログラム中では，16 行目から 19 行目まで，変数からポインタ，配列などの宣言を行っている．通常の型と異なることは，型の名前の前に”structure”が必要なところである．

自作の型の変数に対して，使用方法は通常の変数と少々異なる．自作の型は，最終的にはシステムが用意した型の変数の集まりであり，その変数群を扱うことになる．2.2 プログラム中の構造体では，最終的にはシステムが用意した型と変数である int select_arm と double reward に対して値を出し入れすることになる．

値の出し入れの方法 (構造体の各要素へのアクセスの仕方) は通常の構造体変数とポインタの構造体変数で異なる．

まず通常の構造体変数の場合，値の出し入れの方法は 23 行目，24 行目，26 行目，27 行目のように

- 構造体変数名. 要素となる変数名

と”.”を使って要素にアクセスする．

一方，構造体変数がポインタの場合，例えば 31 行目のように

- 構造体変数名-> 要素となる変数名

と”->”を使って要素にアクセスする．ただし，38 行目，39 行目のように*を使うとポインタ変数の持つアドレスの値，となるので通常変数と同じになるので”.”で要素にアクセスすることが出来る．

2.1 準備

ディレクトリ ”programming11”を作成する．今回の演習では，プログラムの作成や必要ファイルのダウンロードは，programming11 ディレクトリで行う．

2.2 プログラム

```
1  /*********************************
2        構造体の基本的な使い方
3  *********************************/
4
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  /* 構造体の型を宣言 */
9  struct score{
10   int select_arm; /* 選択した腕 */
11   double reward;  /* 得られた報酬 */
12 };
```

```
13
14  int main(){
15
16    /* 構造体の変数の宣言 */
17    struct score normal;
18    struct score* pointer;
19    struct score array[2];
20    struct score* parray;
21
22    /* 通常変数の扱い型 */
23    normal.select_arm=3;
24    normal.reward=5.9;
25
26    printf("normal の select_arm:%d\n",normal.select_arm);
27    printf("normal の reward:%lf\n",normal.reward);
28
29    /* ポインタの扱い型 */
30    pointer = &normal;
31    pointer->select_arm=6;
32    pointer->reward=11.8;
33
34    printf("pointer の select_arm:%d\n",pointer->select_arm);
35    printf("pointer の reward:%lf\n",pointer->reward);
36
37    /* ポインタの扱い型 こちらでも大丈夫*/
38    (*pointer).select_arm=6;
39    (*pointer).reward=11.8;
40
41    printf("pointer の select_arm:%d\n",(*pointer).select_arm);
42    printf("pointer の reward:%lf\n",(*pointer).reward);
43
44    /* 配列の扱い型 */
45    array[0].select_arm=12;
46    array[0].reward=23.6;
47
48    printf("array の select_arm:%d\n",array[0].select_arm);
49    printf("array の reward:%lf\n",array[0].reward);
50
51    /* ポインタで動的配列 */
52    parray = (struct score*)malloc(sizeof(struct score)*2);
53    /* 使い方は配列と同じ */
54    /* parray[1] の代わりに, *(parray+1) でも大丈夫 */
55
```

```
56      free(parray);
57      return 0;
58    }
59
```

2.3 コンパイル

上記のプログラムを作成し，ファイル名 structure-sample.c として保存してコンパイルを行う．コンパイル後の名前を structure-sample とする．

> gcc -o structure-sample structure-sample.c

2.4 動作実験

作成したプログラムを実行させ，動作を理解しよう．

2.5 調べてみよう

構造体の中に構造体が入るとどうなるか，調べてみよう．

2.6 やってみよう

2.6.1 選択した腕と報酬の出力

バンディットで高得点をとるために，どの腕を選択したときに，どのような報酬が得られたか，という情報が非常に有用である．そこで下記のプログラムを参考に，各回の選択した腕とその時得られた報酬を構造体で記憶し，最後に一覧を表示するプログラムをつくってみよう．

```
1   /*********************************************************
2                         腕の選択と分析
3   *********************************************************/
4
5   #include <stdio.h>  /* システムの用意した入出力        */
6   #include <stdlib.h> /* システムの用意した便利関数      */
7   #include "bandit.h" /* ユーザが用意したバンデット関連 */
8
9   /* 構造体の型を宣言 */
10  struct score{
11    int select_arm; /* 選択した腕 */
12    double reward;  /* 得られた報酬 */
13  };
14
15
16  int main(){
```

```
17      /* ここに腕・報酬の情報を記憶する構造体を定義する */
18      /* 複数の情報を記憶するために，構造体を配列化する */
19
20      int loop, num_arm, arm_select;
21      double reward;
22
23      init_bandit();           /* バンディットの初期化 */
24
25      num_arm = get_arm_num();/* バンディットの腕の数を取得 */
26      printf("このバンディットの腕の数は%d です\n",num_arm);
27
28      /* バンディットの腕の選択（10 試行分） */
29      for(loop=0 ; loop<10 ; loop++){
30        printf("バンデットの腕の番号を選択してください [1-%d]:",num_arm);
31        scanf("%d", &arm_select);
32        reward = bandit(arm_select);
33
34        /* 腕・報酬の情報を構造体に記憶 */
35
36      }
37
38      /* 記憶した情報の一覧を表示 */
39
40
41      return 0;
42    }
```

3　参考文献

参考文献

[1] プログラミング演習～ガイダンス～

[2] プログラミング演習～配列～