プログラミングII

第6回: 構造体とポインタ

2018年12月26日(水)

筑波大学 情報メディア創成学類
三河 正彦

引数として構造体を使う

- 教科書11.4 (p.368~)
- 関数の引数に構造体を使った場合
  - 値渡し: 関数には実引数のコピーが渡される.
- 関数の引数に構造体へのポインタを使った場合
  - 参照渡し:関数には参照先へのポインタが渡される.
- 以上これまでに習ったポインタの話と同じ.

構造体とポインタ

- 復習
  - メンバへのアクセス
    - 構造体型変数名.メンバ

```
struct SEITO {
  char name[20];
  int  kokugo;
  ...
};

int main(void)
{
  struct SEITO student;

  student.kokugo = 95;
  ...
}
```

- ポインタによるメンバへのアクセス
  - 構造体へのポインタ->メンバ
  - アロー演算子(間接メンバ演算子とも言う)
  - Sample9.c (p.371)

```
  struct SEITO *student;
  student->kokugo = 95;
```

構造体とポインタ (使用例)

```
 1 /* ptandst.c */
 2 #include <stdio.h>
 3 #include <string.h>
 4
 5 /* 構造体の宣言 */
 6 struct person {
 7   char name[20];
 8   int  age;
 9 };
10
11 int main(void)
12 {
13   struct person uni;
14   struct person *ptuni;
15
16   ptuni = &uni;
17
18   /* 構造体のメンバへの代入 */
19   strcpy(uni.name, "UNI TORO");
20   uni.age = 20;
21
22   /* 構造体のメンバの表示 */
23   printf("%s\n", uni.name);
24   printf("%d 才\n", uni.age);
25
26   /* 構造体のメンバの表示(ポインタ) */
27   printf("%s\n", (*ptuni).name);
28   printf("%d 才\n", (*ptuni).age);
29
30   /* 構造体のメンバの表示(あろ一演算子) */
31   printf("%s\n", ptuni->name);
32   printf("%d 才\n", ptuni->age);
33 }
```

構造体: 関数の引数

```
/* func_arg_st.c */
#include <stdio.h>

struct person {
    char name[20];
    double height;
    double weight;
};

void printperson(struct person p);

int main(void)
{
    struct person tanaka={"Tanaka", 170.0, 60.0},
                  yamada={"Yamada", 160.0, 70.0};

    printperson(tanaka); /* printpersonを呼び出しtanakaの内容を表示 */
    printperson(yamada); /* printpersonを呼び出しyamadaの内容を表示 */

    return 0;
}

void printperson(struct person p)
{
    printf("名前: %s\n", p.name);
    printf("身長: %f\n", p.height);
    printf("体重: %f\n", p.weight);
}
```

構造体: 関数の引数 (ポインタ)

```
/* func_arg_st.c */
#include <stdio.h>

struct person {
    char name[20];
    double height;
    double weight;
};

void printperson(struct person *p);

int main(void)
{
    struct person tanaka={"Tanaka", 170.0, 60.0},
                  yamada={"Yamada", 160.0, 70.0};

    printperson(&tanaka); /* printpersonを呼び出しtanakaの内容を表示 */
    printperson(&yamada); /* printpersonを呼び出しyamadaの内容を表示 */

    return 0;
}

void printperson(struct person *p)
{
    printf("名前: %s\n", p->name);
    printf("身長: %s\n", p->height);
    printf("体重: %s\n", p->weight);
}
```

構造体: 関数の引数 (ポインタ)

```c
/* func_return_st.c */
#include <stdio.h>

struct person {
    char name[20]; /* 名前を格納する変数 */
    int  age;      /* 年齢を格納する変数 */
};

/* プロトタイプ宣言 */
void inputperson(struct person *p);
void printperson(struct person p);

int main(void)
{
    struct person p;

    /* inputperson()の呼び出し */
    inputperson(&p);

    printf("入力されたデータ:\n");
    printperson(p);

    return 0;
}

void inputperson(struct person *p)
{
    printf("名前は?");
    fgets(p->name, 20, stdin);
    printf("年齢は?");
    scanf("%d", &p->age);
}

void printperson(struct person p)
{
    printf("名前: %s\n", p.name);
    printf("年齢: %d\n", p.age);
}
```

構造体: 関数の引数 (ポインタ)

```c
/* func_return_st2.c */
#include <stdio.h>

struct person {
    char name[20];   /* 名前を格納する変数 */
    int  age;     /* 年齢を格納する変数 */
};

/* プロトタイプ宣言 */
/* inputperson()の戻り値が構造体 */
struct person inputperson(void);
void printperson(struct person p);

int main(void)
{
    struct person p;

    /* inputperson()の呼び出し */
    p = inputperson();

    printf("入力されたデータ:\n");
    printperson(p);

    return 0;
}

struct person inputperson(void)
{
    struct person p;

    printf("名前は?");
    fgets(p.name, 20, stdin);
    printf("年齢は?");
    scanf("%d", &p.age);

    /* 構造体変数pの値をreturnする */
    return p;
}

void printperson(struct person p)
{
    printf("名前: %s\n", p.name);
    printf("年齢: %d\n", p.age);
}
```

共用体 union

- 教科書11.5 (p.378~)
- 各メンバは同じメモリを共有する.

列挙型 enum

- 教科書11.6 (p.381~)
- 列挙型変数はあらかじめ列挙した値のどれかしかとらない.

```
/* Talk mode */
#define   NUM_MODE  12
#define   BOOK      0
#define   WEATHER   1
#define   STAFF     2
#define   PLACE     3
#define   SELFINTRO 4
#define   JOB       5
#define   GREETING  6
#define   WAITING   7
#define   THANKYOU  8
#define   GOODBYE   9
#define   ANGRY     10
#define   UNKNOWN   11
```

```
typedef enum state_talk {
    TALK_WAITING=0,
    TALK_GREETING,
    TALK_WEATHER,
    TALK_LIBRARY,
    TALK_STAFF,
    TALK_PLACE,
    TALK_SELF,
    TALK_JOB,
    TALK_THANKYOU,
    TALK_GOODBYE,
    TALK_ANGRY,
    TALK_UNKNOWN,
    TALK_INIT,
    NUM_STATE_TALK
} state_talk;
```

列挙型 enum の使用例

```
#include <stdio.h>

typedef enum state_base {
    BASE_CONVERSATION=0,
    BASE_APPROACH,
    BASE_DEPARTURE,
    BASE_STAND,
    BASE_EMPTY,
    BASE_INIT,
    NUM_STATE_BASE
} state_base;
```

```
int main(void)
{
    state_base state;

    switch (state) {
    case BASE_CONVERSATION:
        // 会話中の処理

        break;
    case BASE_APPROACH:
        // 人間が近づいて来た時の処理

        break;
    }
}
```

おわり

筑波大学情報メディア創成学類 三河 正彦