Structure

C 중급 세미나 - 김혜윤 -

Contents

1. 구조체
   - 구조체 문법
   - 연산자 .
   - 구조체 대입

2. 구조체 포인터
   - 구조체 포인터 / 구조체 포인터 배열 / 활용
   - 연산자 -> / 연산자 -> 활용

3. 실습

구조체

구조체 문법 이전 코드 (배열 사용)

```
void main() {

    int choice; // 유저가 선택한 메뉴
    char food[max_food][10]; // 식재료 개수 5개, 글자 수 10자 제한
    int expiration_date[max_food][3]; // 유통기한 개수 5개, year, month, date
    int food_num = 0; // 현재 식재료 개수
    int date[3] = { 2020, 3, 15 }; // 오늘 날짜

    printf("<< 자취생 냉장고 관리 프로그램 >> \n\n");
```

구조체 문법 구조체 적용

- 구조체의 원소: 멤버(member)
- 멤버 초기화 불가
- 구조체도 멤버로 올 수 있음

```
void main() {

    int choice; // 유저가 선택한 메뉴
    int food_num = 0; // 현재 식재료 개수
    int date[3] = { 2020, 3, 15 }; // 오늘 날짜
    struct Food food[max_food];
    struct Food* food_ptr = food;
    printf("<< 자취생 냉장고 관리 프로그램 >> \n\n");
```

```
struct Food {
    char name[10];
    int expiration_date[3];
    int num;
};
```

```
struct Food {
    char name[10];
    int expiration_date[3];
    int num = 0; // error
};
```

연산자 .

```c
#include <stdio.h>

struct Food {
    char name[10]; // 이름
    int expiration_date[3]; // 유통기한
    int num; // 개수
 };

int main() {
    struct Food ff;
    strcpy(ff.name, "apple");
    ff.num = 3;
    printf("ff.name: %s\n", ff.name);
    printf("ff.num : %d", ff.num);

    return 0;
}
```

연산자 . 으로 구조체 멤버에 접근

```
1  ff.name: apple
2  ff.num : 3
```

구조체 대입

```c
#include <stdio.h>
struct Food {
    char name[10]; // 이름
    int expiration_date[3]; // 유통기한
    int num; // 개수
};
int main() {
    struct Food ff, pp;

    strcpy(ff.name, "apple");
    ff.num = 5;
    pp = ff;
    printf("ff.name: %s\n", ff.name);
    printf("ff.num : %d\n\n", ff.num);
    printf("pp.name: %s\n", pp.name);
    printf("pp.num : %d", pp.num);

    return 0;
}
```

변수처럼 대입 가능
멤버 값들이 대입됨

```
1  ff.name: apple
2  ff.num : 5
3
4  pp.name: apple
5  pp.num : 5
```

구조체 포인터

구조체 포인터 / 포인터 배열

```
void main() {

    int choice; // 유저가 선택한 메뉴
    int food_num = 0; // 현재 식재료 개수
    int date[3] = { 2020, 3, 15 }; // 오늘 날짜
    struct Food food[max_food];
    struct Food* food_ptr = food;
    printf("<< 자취생 냉장고 관리 프로그램 >> \n\n");
```

```
struct Food {
    char name[10];
    int expiration_date[3];
    int num;
};
```

구조체 포인터 활용

```c
#include <stdio.h>
struct Food {
    char name[10]; // 이름
    int expiration_date[3]; // 유통기한
    int num; // 개수
};
int main() {
    struct Food ff;
    struct Food *ff_ptr = &ff;

    strcpy((*ff_ptr).name, "apple");
    (*ff_ptr).num = 5;
    printf("ff.name: %s\n", (*ff_ptr).name);
    printf("ff.num : %d", (*ff_ptr).num);

    return 0;
}
```

우선순위	연산자(연산기호)
1	() [] -> .
2	! ~ ++ -- * & sizeof() (자료형)
3	* / %

*보다 .이 우선순위가 높음
→ (*ptr).member 괄호 필수

```
1  ff.name: apple
2  ff.num : 3
```

연산자 ->

```c
#include <stdio.h>
struct Food {
    char name[10]; // 이름
    int expiration_date[3]; // 유통기한
    int num; // 개수
};
int main() {
    struct Food ff;
    struct Food *ff_ptr = &ff;

    strcpy(ff_ptr->name, "apple");
    ff_ptr->num = 5;
    printf("ff.name: %s\n", ff_ptr->name);
    printf("ff.num : %d", ff_ptr->num);

    return 0;
}
```

연산자 ->으로 구조체 포인터에서
역참조하지 않고
바로 멤버 접근 가능

```
1   ff.name: apple
2   ff.num : 3
```

연산자 -> 활용

```c
#include <stdio.h>
struct Food {
    char name[10]; // 이름
    int expiration_date[3]; // 유통기한
    int *num; // int형 포인터
};
int main() {
    struct Food ff;
    struct Food *ff_ptr = &ff;
    int i  = 0;

    strcpy(ff_ptr->name, "apple");
    ff_ptr->num = &i;
    *ff_ptr->num = 5;
    printf("ff.name: %s\n", ff_ptr->name);
    printf("ff.num : %d", *ff_ptr->num);

    return 0;
}
```

우선순위	연산자(연산기호)
1	() [] -> .
2	! ~ ++ -- * & sizeof() (자료형)
3	* / %

*보다 ->이 우선순위가 높음
→ ->먼저 해석

```
1  ff.name: apple
2  ff.num : 3
```

실습

실습 코드 설명 기본 변수 및 구조체

```
void main() {

    int choice; // 유저가 선택한 메뉴
    int food_num = 0; // 현재 식재료 개수
    int date[3] = { 2020, 3, 15 }; // 오늘 날짜
    struct Food food[max_food];
    struct Food* food_ptr = food;
    printf("<< 자취생 냉장고 관리 프로그램 >> \n\n");
```

```
struct Food {
    char name[10];
    int expiration_date[3];
    int num;
};
```

실습 코드 설명 추가된 부분

```
<< 자취생 냉장고 관리 프로그램 >>

행동을 선택하세요
1. 식재료 추가하기
2. 현재 있는 식재료 보여주기
3. 유통기한 지난 식재료 보여주기
4. 식재료 개수 증감
5. 프로그램 종료

번호 입력 :
```

```c
while (1) {
        printf("행동을 선택하세요 \n");
        printf("1. 식재료 추가하기 \n");
        printf("2. 현재 있는 식재료 보여주기 \n");
        printf("3. 유통기한 지난 식재료 보여주기 \n");
        printf("4. 식재료 개수 증감 \n");
        printf("5. 프로그램 종료 \n\n");

        printf("번호 입력 : ");
        scanf("%d", &choice);
        printf("\n");
        switch (choice) {
            case 1: add_food(food_ptr, &food_num); break; /* 식재료 추가 */
            case 2: show_food(food_ptr, food_num); break; /* 현재 식재료 보여주기 */
            case 3: show_expired_food(food_ptr, food_num, date); break; /* 유통기한 지난 식재료 보여주기 */
            case 4: change_food_num(food_ptr, &food_num, date); break; /* 식재료 개수 증감 */

        }
        printf("\n");
        if (choice == 5) {
            /* 프로그램을 종료한다. */
            break;
        }
    }
```

```c
struct Food {
    char name[10];
    int expiration_date[3];
    int num;
};
```

실습 구현 내용 #1, 2, 3

```c
void add_food(struct Food *ptr, int* food_num) {
    /*** Your Code Here ***/
    printf("add food\n"); // delete
}

void show_food(struct Food* ptr, int food_num) {
    /*** Your Code Here ***/
    printf("show food\n"); // delete
}

void show_expired_food(struct Food* ptr, int food_num, int date[]) {
    /*** Your Code Here ***/
    printf("show expired food\n"); // delete
}
```

- 식재료 개수 멤버를 추가해서 배열을 구조체로 바꾸세요
- 각 함수의 내용은 포인터 실습과 동일합니다

실습 구현 내용 #4

```c
void change_food_num(struct Food* ptr, int* food_num, int date[]) {
    /*** Your Code Here ***/
    printf("change_food_num\n"); // delete
}
```

```c
struct Food {
    char name[10];
    int expiration_date[3];
    int num;
};
```

- 지금 냉장고에 무슨 음식이 있는지 먼저 보여주기
- 식재료가 없으면 함수 종료
- 식재료가 있으면 개수 입력받아서 조정하기(형식 자유)

```
번호 입력 :4

<< 현재 냉장고 상황 >>
식재료가 없습니다. 냉장고에게 식재료를 주세요ㅠㅠ.
```

```
번호 입력 :4

<< 현재 냉장고 상황 >>
1: apple      | 2020년  1월  1일 |  5개

개수를 바꾸고 싶은 식재료 번호:1
 바꾸고 싶은 개수 (ex. 2: 2개 증가, -2: 2개 감소):-3
 바뀐 식재료 정보
apple      | 2020년  1월  1일 |  2개
```