

8장. 구조체

Visualstudio 2019



구조체의 개념

◆ 구조체는 왜 필요할까?

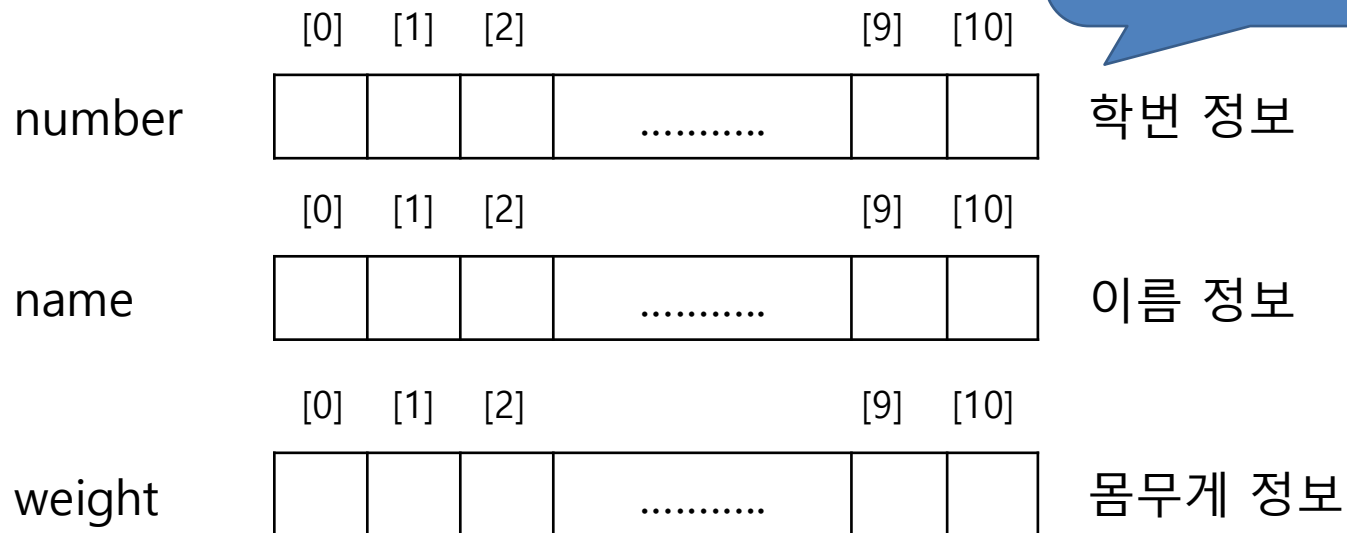
학생 10명의 학번과 이름, 몸무게 정보 저장 – 배열 자료형 이용

```
int number[10];
```

```
char name[20];
```

```
double weight[10];
```

정보가 흩어져서 저장되는 한계 발생



구조체란 무엇인가?

◆ 구조체(structure)란?

다양한 자료형을 그룹화하여 하나의 변수로 처리할 수 있게 만든 자료형이다. 개발자가 다양한 정보를 저장하기 위해 필요에 따라 생성하는 자료형을 사용자 정의 자료형 또는 구조체라 한다.

- 구조체 정의

```
struct 구조체이름{  
    자료형 멤버이름;  
};
```

- 객체 생성

```
struct 구조체이름 변수이름;
```



구조체의 정의 및 사용

◆ 구조체 정의 – 멤버 변수는 일반 변수처럼 초기화 할 수 없음

```
struct student {  
    int number;  
    char name[20];  
    double weight;  
};
```

int a →

정수

4byte

struct student;

정수

문자

실수

4byte 20byte 8byte

◆ 구조체 객체(변수) 생성

```
struct student s;
```



구조체의 정의 및 사용

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

struct Student {
    int number;      //번호(멤버변수)
    char name[20];   //이름
    double weight;   //몸무게
};
```

```
int main() {
    struct Student s; //객체 변수 s 생성

    s.number = 101;
    //s.name = "강하늘";
    strcpy(s.name, "강하늘");
    //strcpy_s(s.name, sizeof(s.name), "강하늘");
    s.weight = 62.3;

    printf("학번 : %d\n", s.number);
    printf("이름 : %s\n", s.name);
    printf("몸무게 : %3.11f\n", s.weight);

    return 0;
}
```



구조체 배열

◆ 구조체 배열 – 객체를 여러 개 생성

```
//구조체 배열
/*struct Student s1 = {101, "안산", 50.2};
struct Student s2 = { 102, "박대양", 70.5 };*/

struct Student s[2] = {
    {101, "안산", 50.2},
    {102, "박대양", 70.5}
};
int i;

for (i = 0; i < 2; i++) {
    printf("학번 : %d\n", s[i].number);
    printf("이름 : %s\n", s[i].name);
    printf("몸무게 : %3.11f\n", s[i].weight);
}

/*printf("학번 : %d\n", s1.number);
printf("이름 : %s\n", s1.name);
printf("몸무게 : %3.11f\n", s1.weight);
```



구조체의 복사

◆ 구조체 복사 – 대입(할당)연산자로 쉽게 복사 가능

```
int main() {  
    //구조체의 복사  
    struct Three_D d3;  
    struct Three_D d4;  
  
    d3.x = 100;  
    d3.y = 200;  
    d3.z = 300;  
  
    d4 = d3; //d3의 값을 d4에 저장  
  
    printf("x좌표 : %d, y좌표 : %d, z좌표 : %d\n", d3.x, d3.y, d3.z);  
    printf("x좌표 : %d, y좌표 : %d, z좌표 : %d\n", d4.x, d4.y, d4.z);  
  
    return 0;  
}
```

```
struct Three_D {  
    int x;  
    int y;  
    int z;  
};
```



구조체 변수를 함수의 매개변수로 사용

- 함수의 매개변수를 구조체 변수로 사용
 - 눈의 왼쪽, 오른쪽 시력을 교환하는 함수 정의

```
struct Eyesight swap(struct Eyesight);
struct Eyesight {
    double left;
    double right;
};

int main() {

    struct Eyesight eye = { 1.5, 1.2 };

    eye = swap(eye); //교환 함수 호출

    printf("왼쪽눈 시력 : %.11f\n", eye.left);    //1.2
    printf("오른쪽눈 시력 : %.11f\n", eye.right); //1.5

    return 0;
}
```



구조체 변수를 함수의 매개변수로 사용

- 눈의 왼쪽, 오른쪽 시력을 교환하는 함수 정의

```
struct Eyesight swap(struct Eyesight sight) {  
    double temp;  
    temp = sight.left;  
    sight.left = sight.right;  
    sight.right = temp;  
  
    return sight;  
}
```



구조체 typedef 키워드 사용

- 구조체 변수를 선언할때 struct 키워드 생략

typedef로 구조체 정의 – 별칭 사용

```
typedef struct 구조체이름{  
    자료형 멤버이름;  
} 구조체별칭;
```



```
typedef struct _Eyesight{  
    double left;  
    double right;  
} Eyesight;
```

구조체 이름은 **_구조체** 이름으로 한다.

```
Eyesight eye;
```



구조체 typedef 키워드 사용

- 구조체 변수를 선언할때 struct 키워드 생략

```
Eyesight swap(Eyesight);
typedef struct _Eyesight {
    double left;
    double right;
} Eyesight;

int main() {

    Eyesight eye = { 1.5, 1.2 };

    eye = swap(eye); //교환 함수 호출

    printf("왼쪽눈 시력 : %.11f\n", eye.left);    //1.2
    printf("오른쪽눈 시력 : %.11f\n", eye.right); //1.5

    return 0;
}
```

```
Eyesight swap(Eyesight sight) {
    double temp;

    temp = sight.left;
    sight.left = sight.right;
    sight.right = temp;

    return sight;
}
```



구조체 참조 관계

◆ 구조체 참조 관계

```
struct Point {  
    int x;  
    int y;  
};
```

```
struct Rectangle {  
    struct Point p1;  
    struct Point p2;  
};
```

사각형

점

점1(x, y)
점2(x, y)

x, y

```
int main() {  
    struct Rectangle rect;  
  
    rect.p1.x = 1;  
    rect.p1.y = 2;  
    rect.p2.x = 3;  
    rect.p2.y = 4;  
  
    printf("p1(%d, %d), p2(%d, %d)\n",  
        rect.p1.x, rect.p1.y, rect.p2.x, rect.p2.y);  
  
    return 0;  
}
```



구조체 참조 관계

➤ 두 점 사이의 거리 구하는 프로그램

```
int main()
{
    struct point p1, p2;
    int xDiff, yDiff;
    double distance;

    printf("1번 점의 좌표를 입력하세요 : ");
    scanf_s("%d %d", &p1.x, &p1.y, sizeof(p1.x), sizeof(p1.y));

    printf("2번 점의 좌표를 입력하세요 : ");
    scanf_s("%d %d", &p2.x, &p2.y, sizeof(p2.x), sizeof(p2.y));

    xDiff = p1.x - p2.x;
    yDiff = p1.y - p2.y;
    distance = sqrt(xDiff*xDiff + yDiff * yDiff);
    printf("두 점 사이의 거리는 %1f입니다.", distance);
    return 0;
}
```

```
#include <stdio.h>
#include <math.h>
struct point {
    int x;
    int y;
};
```

Microsoft Visual Studio 디버그 콘솔

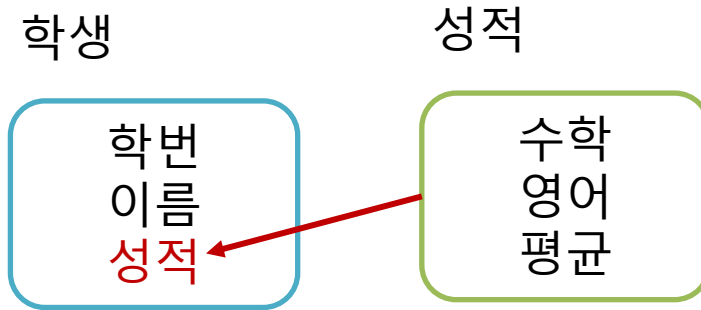
```
1번 점의 좌표를 입력하세요 : 1 1
2번 점의 좌표를 입력하세요 : 5 5
두 점 사이의 거리는 5.656854입니다.
```



구조체 참조 관계

◆ 구조체 참조 관계 – 헤더 파일로 만들기

학생이 성적을 참조하는 관계.



students.h

```
//Students가 Score 구조체를 참조
struct Score {
    int math;
    int eng;
    double avg;
};

struct Students {
    int number;
    char name[20];
    struct Score score;
};
```



구조체 참조 관계

◆ 구조체 참조 관계 – 헤더 파일로 만들기

학생이 성적을 참조하는 관계.

```
#include <stdio.h>
#include "students.h"

int main() {

    struct Students s1 = { 101, "서우주", {85, 90, 0.0} };
    s1.score.avg = (double)(s1.score.math + s1.score.eng) / 2;

    printf("번호 : %d\n", s1.number);
    printf("이름 : %s\n", s1.name);
    printf("평균 : %3.11f\n", s1.score.avg);
    return 0;
}
```



구조체 포인터 변수

- 구조체 포인터 변수

구조체 변수가 닷(.) 연산자를 사용하는 반면, 구조체의 포인터 변수는 참조 연산자(->)를 사용한다.

ptr -> num = 1003

strcpy(ptr->name, "장그래")

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Person {
    char name[20];
    int age;
    char address[100];
};
```

```
int main() {
    struct Person* p;
    p = (struct Person*)malloc(sizeof(struct Person));

    strcpy(p->name, "더조은");
    p->age = 30;
    strcpy(p->address, "서울시 노원구 상계동");

    printf("이름 : %s\n", p->name);
    printf("나이 : %d\n", p->age);
    printf("주소 : %s\n", p->address);

    return 0;
}
```



구조체 포인터 변수

- typedef로 구조체 정의

```
typedef struct Person {  
    char name[20];  
    int age;  
    char address[100];  
}person;  
  
int main() {  
    person* p = (person*)malloc(sizeof(person));  
  
    strcpy(p->name, "더조은");  
    p->age = 30;  
    strcpy(p->address, "서울시 노원구 상계동");  
  
    printf("이름 : %s\n", p->name);  
    printf("나이 : %d\n", p->age);  
    printf("주소 : %s\n", p->address);  
  
    return 0;  
}
```



함수의 파라미터로 구조체 사용

```
typedef struct _student {
    int number;
    char name[20];
    int korean; //국어 성적
}student;

void ShowTable(student);
int total = 0;
int main()
{
    student s[3];
    int i;

    printf("1학년의 성적 관리 프로그램입니다.\n");
    for (i = 0; i < 3; i++)
    {
        printf("학번을 입력하세요 : ");
        scanf_s("%d", &s[i].number);
        printf("%d번째 학생의 이름을 입력하세요 : ", i + 1);
        scanf_s("%s", s[i].name, sizeof(s[i].name));
        printf("국어 성적을 입력하세요 : ");
        scanf_s("%d", &s[i].korean);
        printf("\n");
    }
}
```

```
printf("=====\n");
printf("학번\t이름\t국어 성적\n");
for (i = 0; i < 3; i++) {
    ShowTable(s[i]);
}
printf("=====\n");
printf("평균\t%d\n", total/ 3);
return 0;
}

void ShowTable(student tbl)
{
    printf("%d\t%s\t%d\n", tbl.number, tbl.name, tbl.korean);
    total += tbl.korean;
}
```

함수의 파라미터
로 구조체 사용



구조체 포인터의 사용

```
void ShowTable(student *);
int total = 0;
int main()
{
    student s[3];
    char buf[20];
    int i;

    printf("1학년의 성적 관리 프로그램입니다.\n");
    for (i = 0; i < 3; i++)
    {
        printf("학번을 입력하세요 : ");
        scanf_s("%d", &s[i].number);
        printf("%d번째 학생의 이름을 입력하세요\n");
        scanf_s("%s", buf, sizeof(buf));
        strcpy_s(s[i].name, sizeof(buf), buf);
        printf("국어 성적을 입력하세요 : ");
        scanf_s("%d", &s[i].korean);
        printf("\n");
    }
}
```

```
printf("=====\n");
printf("학번\t\t이름\t\t국어 성적\n");
for (i = 0; i < 3; i++) {
    ShowTable(&s[i]);
}
printf("=====\n");
printf("평균\t%d\n", total / 3);
return 0;
```

```
void ShowTable(student *ptr)
{
    printf("%d\t\t%s\t\t%d\n", ptr->number, ptr->name, ptr->korean);
    total += ptr->korean;
}
```

함수의 파라미터로
구조체 포인터 사용



구조체 동적 메모리 할당

- Book 구조체 만들기

```
Book* book;
book = (Book*)malloc(sizeof(Book) * 2);
if (book == NULL) {
    printf("동적 메모리 할당에 실패했습니다.\n");
    exit(1);
}

book->bookNum = 10;
strcpy_s(book->title, sizeof(book->title), "모두의 C언어");

(book + 1) -> bookNum = 11;
strcpy_s((book + 1)->title, sizeof((book + 1)->title), "C++ 언어본색");

showBookInfo(book, 2);
free(book);
```

```
typedef struct _Book {
    int bookNum;
    char title[100];
} Book;
```



구조체 동적 메모리 할당

- Book 출력 함수

```
void showBookInfo(Book* b, int len) {  
    int i;  
    for (i = 0; i < len; i++) {  
        printf("책 번호 %d : %s\n", (b + i) -> bookNum, (b + i) -> title);  
    }  
}
```

