구조체

- 구조체 선언, 구조체 배열, 구조체 포인터 -

성공회대학교 IT융합자율학부
소프트웨어공학전공
홍 성 준

구조체란 무엇인가?

◎ 구조체(structure)의 정의

- 하나 이상의 변수(포인터 변수, 배열을 포함)를 묶어서 새로운 자료형을 정의하는 도구
  - 마우스 커서의 좌표를 저장하기 위해서는 x 좌표와 y 좌표가 필요

```
int xpos;   // 마우스의 x 좌표
int ypos;   // 마우스의 y 좌표
```

  - 구조체를 이용하여 int 형 변수 xpos와 ypos를 하나로 묶어 point라는 새로운 자료형을 정의

```
struct point   // point라는 이름의 구조체 정의
{
    int xpos;   // point 구조체를 구성하는 멤버 xpos
    int ypos;   // point 구조체를 구성하는 멤버 ypos
};
```

구조체란 무엇인가?

◎ 구조체 변수의 선언과 접근

구조체 정의

```
struct point   // point라는 이름의 구조체 정의
{
    int xpos;   // point 구조체를 구성하는 멤버 xpos
    int ypos;   // point 구조체를 구성하는 멤버 ypos
};
```

```
struct person
{
    char name[20];     // 이름 저장
    char phoneNum[20];   // 전화번호 저장
    int age;   // 나이 저장
};
```

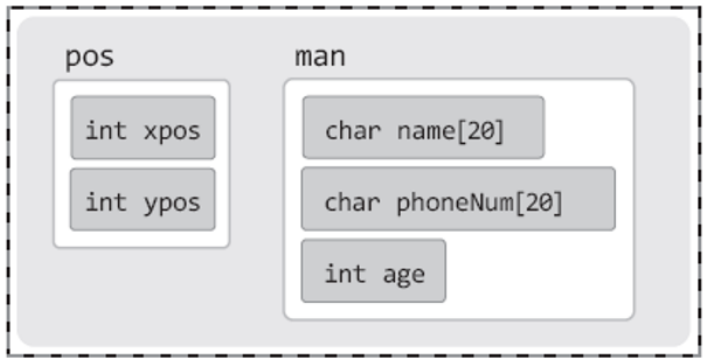
구조체 변수선언의 기본 형태

struct *type_name val_name ;*

struct point pos;

struct person man;

구조체 변수선언의 예



멤버의 접근방식

구조체 변수의 이름 . 구조체 멤버의 이름

pos.xpos=20;

구조체 변수 pos의 멤버 xpos에 20을 저장

printf("%s \n", man.name);

man의 멤버 name에 저장된 문자열 출력

구조체란 무엇인가?

◎ TwoPointDistance.c

- 두 점을 입력 받아 두 점 사이의 거리를 출력하는 프로그램

```c
struct point    // 구조체 point의 정의
{
    int xpos;
    int ypos;
};

int main(void)
{
    struct point pos1, pos2;
    double distance;
    fputs("point1 pos: ", stdout);
    scanf("%d %d", &pos1.xpos, &pos1.ypos);
    fputs("point2 pos: ", stdout);
    scanf("%d %d", &pos2.xpos, &pos2.ypos);
    /* 두 점간의 거리 계산 공식 */
    distance=sqrt((double)((pos1.xpos-pos2.xpos) * (pos1.xpos-pos2.xpos)+
            (pos1.ypos-pos2.ypos) * (pos1.ypos-pos2.ypos)));
    printf("두 점의 거리는 %g 입니다. \n", distance);
    return 0;
}
```

```c
#include <math.h>

double sqrt(double x);    // 제곱근 x의 값을 반환
```

```
point1 pos: 1 3
point2 pos: 4 5
두 점의 거리는 3.60555 입니다.
```

구조체란 무엇인가?

◎ TelephoneInfo.c

```
struct person
{
    char name[20];
    char phoneNum[20];
    int age;
};
int main(void)
{
    struct person man1, man2;
    strcpy(man1.name, "안성준");
    strcpy(man1.phoneNum, "010-1122-3344");
    man1.age=23;
    printf("이름 입력: "); scanf("%s", man2.name);
    printf("번호 입력: "); scanf("%s", man2.phoneNum);
    printf("나이 입력: "); scanf("%d", &(man2.age));
    printf("이름: %s \n", man1.name);
    printf("번호: %s \n", man1.phoneNum);
    printf("나이: %d \n", man1.age);
    printf("이름: %s \n", man2.name);
    printf("번호: %s \n", man2.phoneNum);
    printf("나이: %d \n", man2.age);
    return 0;
}
```

```
이름 입력: 김수정
번호 입력: 010-0001-0002
나이 입력: 27
이름: 안성준
번호: 010-1122-3344
나이: 23
이름: 김수정
번호: 010-0001-0002
나이: 27
```

- 구조체 변수의 멤버에 대한 접근은 일반적인 접근 방식을 그대로 따름

구조체란 무엇인가?

◎ 구조체 변수의 초기화

- 배열과 유사하게 초기화 할 데이터를 중괄호 안에 순서대로 나열
- InitStructVal.c

```
struct point
{
    int xpos;
    int ypos;
};
struct person
{
    char name[20];
    char phoneNum[20];
    int age;
};
int main(void)
{
    struct point pos={10, 20};
    struct person man={"이승기", "010-1212-0001", 21};
    printf("%d %d \n", pos.xpos, pos.ypos);
    printf("%s %s %d \n", man.name, man.phoneNum, man.age);
    return 0;
}
```

```
10 20
이승기 010-1212-0001 21
```

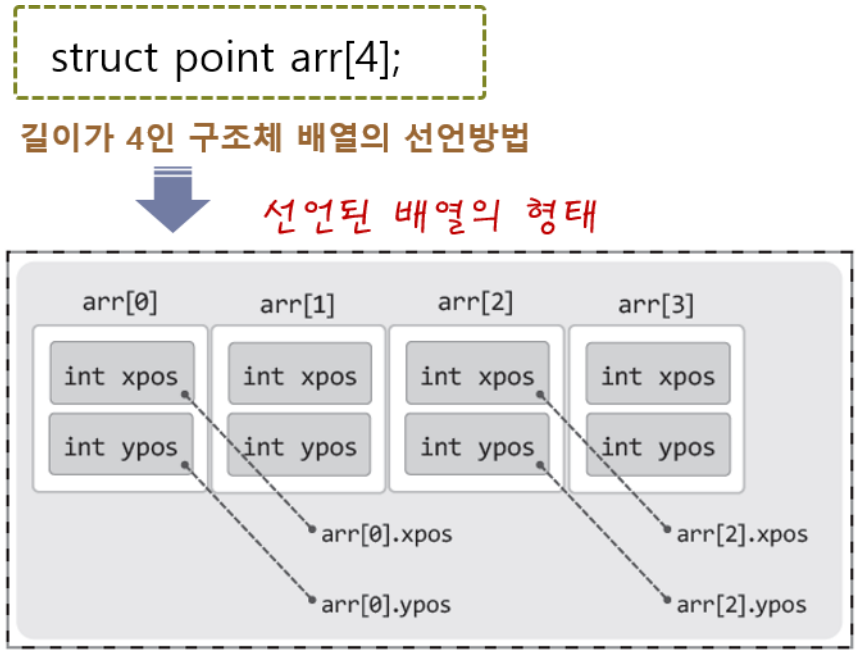
구조체와 배열 그리고 포인터

◎ 구조체 배열의 선언과 접근

- StructArray.c

```
struct point
{
    int xpos;
    int ypos;
};

int main(void)
{
    struct point arr[3];
    int i;

    for(i=0; i<3; i++)
    {
        printf("점의 좌표 입력: ");
        scanf("%d %d", &arr[i].xpos, &arr[i].ypos);
    }

    for(i=0; i<3; i++)
        printf("[%d, %d] ", arr[i].xpos, arr[i].ypos);

    return 0;
}
```

```
점의 좌표 입력: 2 4
점의 좌표 입력: 3 6
점의 좌표 입력: 8 9
[2, 4] [3, 6] [8, 9]
```



구조체와 배열 그리고 포인터

◎ 구조체 배열의 초기화

- InitStructArray.c

```
struct person
{
    char name[20];
    char phoneNum[20];
    int age;
};

int main(void)
{
    struct person arr[3]={
        {"이승기", "010-1212-0001", 21},    // 첫 번째 요소의 초기화
        {"정지영", "010-1313-0002", 22},    // 두 번째 요소의 초기화
        {"한지수", "010-1717-0003", 19}     // 세 번째 요소의 초기화
    };

    int i;
    for(i=0; i<3; i++)
        printf("%s %s %d \n", arr[i].name, arr[i].phoneNum, arr[i].age);

    return 0;
}
```

```
이승기 010-1212-0001 21
정지영 010-1313-0002 22
한지수 010-1717-0003 19
```

구조체와 배열 그리고 포인터

◎ 구조체 변수와 포인터

- 구조체 포인터 변수의 선언 및 연산 방법도 일반적인 포인터 변수의 선언과 연산 방법과 비슷함

```
struct point pos={11, 12};
struct point * pptr=&pos;
        // 구조체 point의 포인터 변수 선언
(*pptr).xpos=10;
        // pptr이 가리키는 구조체 변수의 멤버 xpos에 접근
(*pptr).ypos=20;
        // pptr이 가리키는 구조체 변수의 멤버 ypos에 접근
```

- -> 연산자를 이용한 구조체 변수의 멤버에 접근하기

```
(*pptr).xpos=10;  ⟷  pptr->xpos=10;

(*pptr).ypos=20;  ⟷  pptr->ypos=20;
```

구조체와 배열 그리고 포인터

◎ StructPointer.c

```
struct point
{
    int xpos;
    int ypos;
};

int main(void)
{
    struct point pos1={1, 2};
    struct point pos2={100, 200};
    struct point * pptr=&pos1;
    (*pptr).xpos += 4;
    (*pptr).ypos += 5;
    printf("[%d, %d] \n", pptr->xpos, pptr->ypos);
    pptr=&pos2;
    pptr->xpos += 1;
    pptr->ypos += 2;
    printf("[%d, %d] \n", (*pptr).xpos, (*pptr).ypos);
    return 0;
}
```

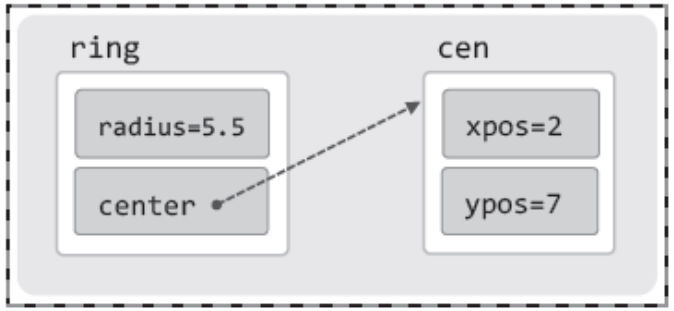
```
[5, 7]
[101, 202]
```

구조체와 배열 그리고 포인터

◎ 구조체의 멤버로 포인터 변수 선언하기

- PointRelation.c

```
struct point
{
    int xpos;
    int ypos;
};
struct circle
{
    double radius;
    struct point * center;
};

int main(void)
{
    struct point cen={2, 7};
    double rad=5.5;
    struct circle ring={rad, &cen};
    printf("원의 반지름: %g \n", ring.radius);
    printf("원의 중심 [%d, %d] \n", (ring.center)->xpos, (ring.center)->ypos);
    return 0;
}
```




```
원의 반지름: 5.5
원의 중심 [2, 7]
```

구조체와 배열 그리고 포인터

◎ 구조체의 멤버로 포인터 변수 선언하기

- StructValAddress.c

```
struct point
{
    int xpos;
    int ypos;
    struct point * ptr;
};

int main(void)
{
    struct point pos1={1, 1};
    struct point pos2={2, 2};
    struct point pos3={3, 3};
    pos1.ptr = &pos2;     // pos1과 pos2를 연결
    pos2.ptr = &pos3;     // pos2와 pos3를 연결
    pos3.ptr = &pos1;     // pos3를 pos1과 연결
    printf("점의 연결관계... \n");
    printf("[%d, %d]와(과) [%d, %d] 연결 \n",
        pos1.xpos, pos1.ypos, pos1.ptr->xpos, pos1.ptr->ypos);
    printf("[%d, %d]와(과) [%d, %d] 연결 \n",
        pos2.xpos, pos2.ypos, pos2.ptr->xpos, pos2.ptr->ypos);
    printf("[%d, %d]와(과) [%d, %d] 연결 \n",
        pos3.xpos, pos3.ypos, pos3.ptr->xpos, pos3.ptr->ypos);
    return 0;
}
```

```
점의 연결관계...
[1, 1]와(과) [2, 2] 연결
[2, 2]와(과) [3, 3] 연결
[3, 3]와(과) [1, 1] 연결
```