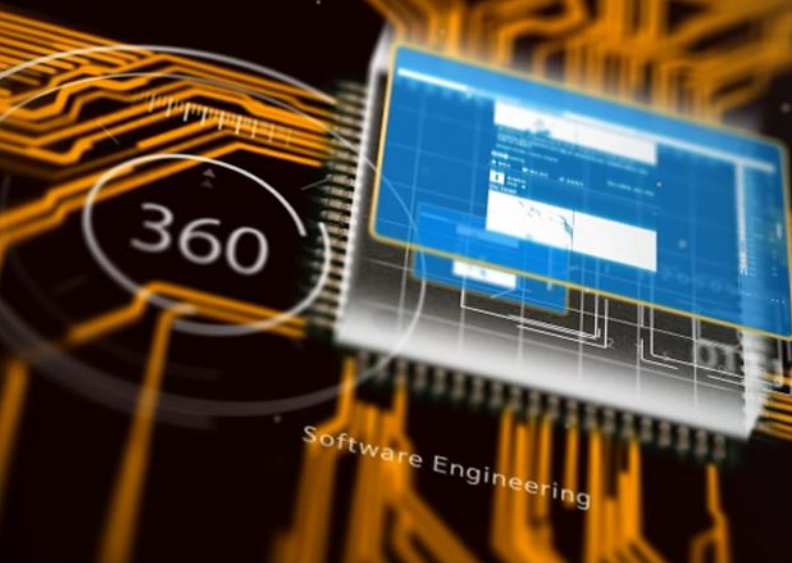


프로그래밍 언어 활용

1011101010001010101

part 1



공용체와 열거체



한국기술교육대학교
온라인평생교육원

학습내용

- 공용체
- 열거체

학습목표

- 공용체의 개념을 알고 구현할 수 있다.
- 열거체의 용도를 알고 코드에 적용할 수 있다.

공용체

1 공용체 활용



동일한 저장 장소에 여러 데이터 타입을 저장하는 자료구조

1 멤버들이 **메모리**를 공유해서 사용하는 기법

2 공용체의 크기는 공용체의 멤버 중 **가장 크기가 큰 멤버에 의해 결정**

```
union 공용체명 {
    멤버;
    멤버;
};
```

```
union sungjuk {
    char grade;
    int score;
}
```

첫 1byte만 참조

4개 byte를 참조
초기화한 값만 유의미

grade

score



공용체

1 공용체 활용

공용체 변수를 초기화할 때는 첫 번째 멤버의 초기값만 지정

```
union sungjuk {
    char grade;
    int score;
}

union sungjuk std = { 'B' };
```

공용체의 멤버에 접근할 때도 '.'와 '->' 연산자를 사용함

```
union sungjuk std;

std.grade = 'B';

std.score = 97 ;
```

```
union sungjuk std;
union sungjuk *pu = &std;

pu->grade = 'B';

pu->score = 97 ;
```

공용체

1 공용체 활용

```
#include <stdio.h>
union sungjuk {
    char grade;
    int score;
};

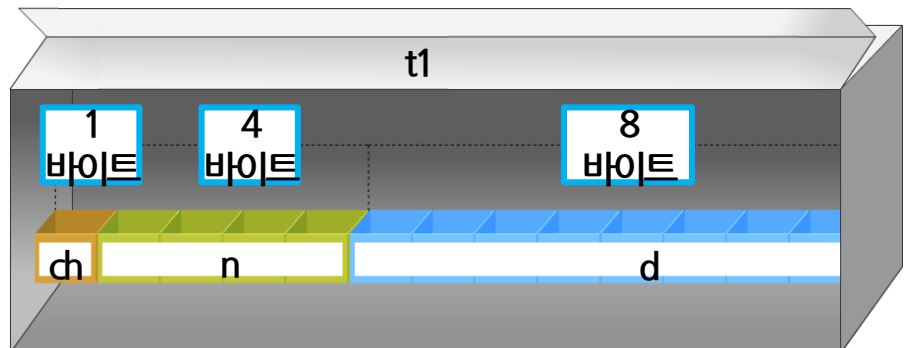
int main()
{
    union sungjuk std ;
    std.grade = 'B';
    printf("%c %d Wn", std.grade, std.score);
    std.score = 97;
    printf("%c %d Wn", std.grade, std.score);
    printf("size %d", sizeof(std));
    return 0;
}
```

```
C:\Users\Wit\Documents\프로젝트\part1\source\04-01.exe
B 66
a 97
size 4

Process exited after 0.03114 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

구조체와 공용체 비교

```
struct test 1 {
    char ch;
    int n;
    double d;
} t1;
```



구조체의 정의

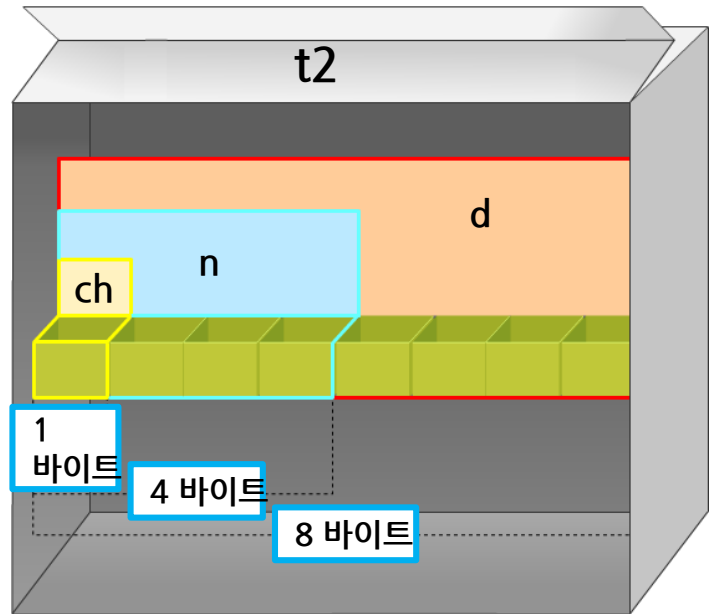
공용체

1 공용체 활용

구조체와 공용체 비교

```
union test 2 {
    char ch;
    int n;
    double d;
} t2;
```

공용체의 정의

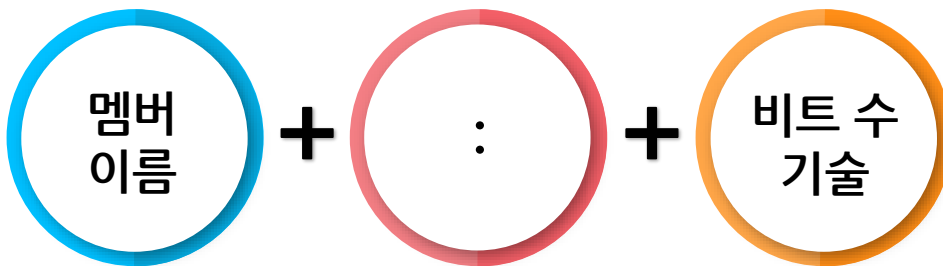


공용체

2 비트필드

1 구조체가 가진 멤버를 **비트 단위**로 사용

2 비트필드 정의



```
struct Date {
    int day;
    int month;
    int year
};
```

```
struct Date {
    int day:5;           // 1~31
    int month:4;         // 1~12
    int year:13;         // 1~4096
};
```



공용체

2 비트필드

3 메모리에 할당할 때, 첫 번째 멤버를 **최하위 비트에서부터 할당**

4 비트필드의 멤버에 표현 가능한 범위 밖의 값을 저장하면 **오버플로우가 발생**

```
#include <stdio.h>
struct Date {
    int day:5;           // 1~31
    int month:4;         // 1~12
    int year:13;         // 1~4096
};
```

```
int main()
{
    struct Date ent;
    ent.month = 17;

    printf("%d \n", ent.month);
    printf("size %d", sizeof(ent));
    return 0;
}
```

year

month

day



```
C:\Users\wit\Documents\프로젝트\part1\source\04-0...
1
size 4
-----
Process exited after 0.03157 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```


공용체

2 비트필드

5

비트필드를 정의할 때는 중간에 일부 비트를 비워두고 멤버를 특정 비트에 할당할 수 있음

```
struct product
```

```
    unsigned int code : 6;  ●————— 상품코드 6비트
```

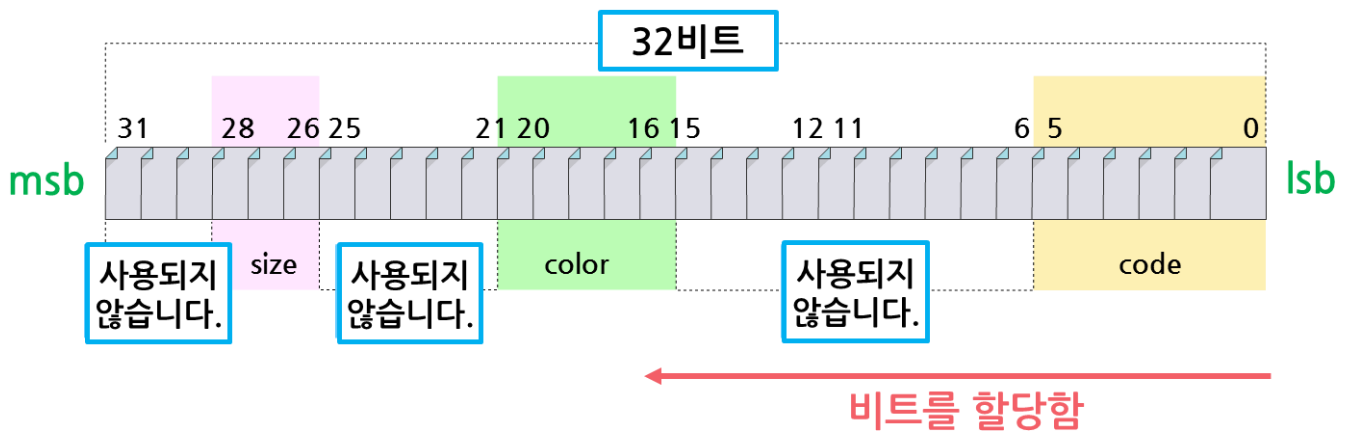
```
    unsigned int : 10;  ●————— 10비트는 사용하지 않음
```

```
    unsigned int color : 5;  ●————— 색상 5비트
```

```
    unsigned int : 5;  ●————— 5비트는 사용하지 않음
```

```
    unsigned int size : 3;  ●————— 크기 3비트
```

```
};
```



공용체

2 비트필드

리틀 엔디안과 빅 엔디안

리틀 엔디안

최하위 바이트부터
메모리에 저장하는 방식



인텔 계열의 CPU

빅 엔디안

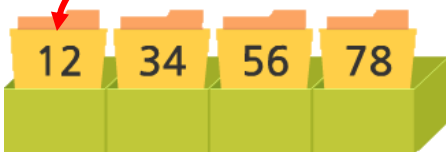
최상위 바이트부터
메모리에 저장하는 방식



모토로라 계열의 CPU

unsigned int n = 0x12345678;

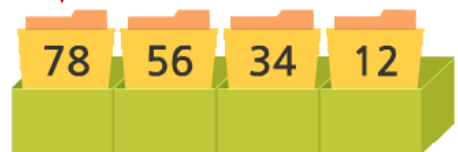
최상위 바이트부터
저장함



0x1000 번지 0x1001 번지 0x1002 번지 0x1003 번지

빅 엔디안 방식

최하위 바이트부터
저장함



0x1000 번지 0x1001 번지 0x1002 번지 0x1003 번지

리틀 엔디안 방식

열거체

1 열거체 활용

나열된 정수 값 중 하나를 갖는 정수형의 일종

형식

```
enum 태그명 { 열거상수1, 열거상수2, 열거상수3, ... };
```

예제

```
enum week { sun, mon, tue, wed, thu, fri, sat };
```

태그명

열거상수

```
enum week { sun, mon, tue, wed, thu, fri, sat };
```

●— sun은 0으로, mon은 1로, ... sat은 6으로 정의됨

```
enum color { red = 10, green = 20, blue = 30 };
```

●— red는 10, green은 20, blue는 30으로 정의됨

일종의 사용자 정의형

열거체 변수에는 열거체 정의에 나열된 열거상수 중 하나를 저장하고 사용

```
enum week weekday;
```

●— 열거체형 변수를 선언함

```
weekday = mon;
```

●— mon 열거상수는 한눈에 월요일이라는 것을 알 수 있음

열거체

1 열거체 활용

열거 상수만 정수형 상수로 정의할 수도 있음

```
enum {red = 10, green = 20, blue = 30};
```



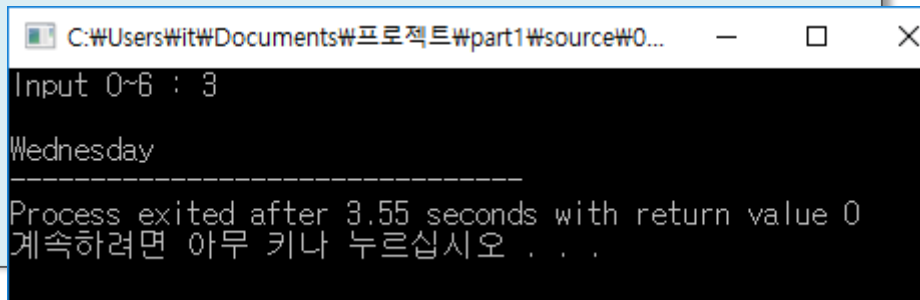
프로그램의 **가독성(Readability)**을 향상시키기 위해 사용

```
#include <stdio.h>
enum weekday {sun, mon, tue, wed, thu, fri, sat};
int main()
{
    enum weekday w_day;
    printf("Input 0~6 : ");
    scanf("%d", &w_day); printf("%Wn");
}
```

열거체

1 열거체 활용

```
switch(w_day){
    case sun : printf("Sunday"); break;
    case mon : printf("Monday"); break;
    case tue : printf("Tuesday"); break;
    case wed : printf("Wednesday"); break;
    case thu : printf("Thursday"); break;
    case fri : printf("Friday"); break;
    case sat : printf("Saturday"); break;
    default : printf(" Input Correct Number ");
}
return 0;
}
```



```
enum group {management, accounting, research,
planning};
```

```
enum group {management=101, accounting, research,
planning};
```

```
enum group {management=101, accounting=201,
research=212, planning=301};
```

열거체

2 typedef

1 재정의

01

데이터 타입의 이름을 **새로운 이름**으로 재정의

02

코딩의 편리성 증대

03

프로그램의 시스템 간 호환성 향상

형식

typedef 기존 데이터형 새 이름;

예제

typedef unsigned int
typedef unsigned char
typedef struct point

UINT;
BYTE;
POINT;

기존의 데이터형

새이름

열거체

2 typedef

2 편리성

구조체에서의 typedef

```
struct telephone {
    char *name;
    int num;
};

typedef struct telephone TEL;
```

or

```
typedef struct telephone {
    char *name;
    int num;
}TEL;
```

TEL my_friend;

3 호환성

```
struct salary {
    int pay;
    int time;
}
```

```
struct salary {
    double pay;
    double time;
}
```

```
typedef int SIZE;
struct salary {
    SIZE pay;
    SIZE time;
}
```

```
typedef double SIZE;
struct salary {
    SIZE pay;
    SIZE time;
}
```

학습정리

1. 공용체



- 공용체는 멤버들이 메모리를 공유해서 사용하는 자료구조임
- 비트필드는 구조체 멤버를 비트 단위로 할당하여 사용함

2. 열거체



- 열거체는 정수형의 일종으로 변수가 가질 수 있는 값을 열거 상수로 나열하는 것임
- typedef은 기존의 데이터형에 새로운 이름을 만드는 기능임
- typedef는 호환성과 가독성을 향상시키는 목적으로 사용함