<13. 구조체>

<13.5 공용체 – union>

-union example { // ‘키워드’ 쓰고 ‘태그’

int stu\_number; // 멤버변수

char reg\_number[15];

};  
- 정의 : 같은 메모리 영역을 여러 개의 변수들이 공유할 수 있게 함 -> 메모리 절약

- 공용체 변수의 크기 : 멤버 중에서 크기가 가장 큰 변수와 크기가 같음

- 초기화 : 첫번째 멤버만 초기화된다.

- 접근 방법 : 구조체와 동일. ‘.’ 이용

- tip) 공용체에 저장된 값이 어떤 타입인지 어떻게 아나?

=> 모름. 프로그래머가 기억해야함. 기억하기 위해 보통 현재 저장된 값의 타입을 나타내는 별도의 변수가 추가됨

<13.6 구조체와 함수>

- 구조체가 매개변수 또는 반환값으로 사용될 때 => 값에 의한 호출

=> 구조체 변수의 ‘복사본’이 인수로 전달

=> 함수 안에서 인수의 값 변경되더라도 원본 구조체에 영향 주지 않는다.

단점 : 구조체 크기가 크면 상당한 시간 소요 -> 구조체의 포인터를 사용하는 것이 바람직

<13.7 열겨형>

- enum days {SUN, MON, TUE, WED, THU, FRI, SAT}; // 첫번째 식별자 값은 0

- 열거형으로 선언된 변수는 열거형에 정의된 상수들만 가질 수 있다.

- 가독성 높이기 위해 씀

- 장점 :

1. 특정한 숫자 대신에 기호를 사용 -> 가독성 올림

2. 변수가 열거된 값 이외의 값을 취하는 것을 막아서 오류 줄여줌

<13.8 typedef>

- 새로운 자료형을 정의하는 것

- 복잡한 형식(예를 들어 어떤 구조체)도 새로운 자료형으로 만들 수 있다.