

C++ / JAVA 프로그래밍

09주차 실습
구조체 사용하기

- 구조체의 기본적인 사용법 익히기
 - 정의, 선언, 멤버변수 접근
- 구조체를 함수에서 사용하는 방법
 - 구조체를 인자로 전달하기
 - 함수의 결과를 구조체로 돌려받기
 - 구조체 배열을 인자로 전달하고 참조호출로 결과 돌려받기
- 구조체를 이용해 간단한 도서 관리 프로그램 작성

구조체가 필요한 이유 (1/2)

- 예를 들어 학생에 대한 정보를 저장하는 프로그램을 만든다고 가정해봅시다.
- 이름, 학과, 학번, 주민등록번호, 주소, 전화번호 등 여러가지 정보를 저장할 수 있습니다.
- 변수를 이용해 저장하는 경우
 - name1, department1, std_no1, citizen_no1, ... 등의 변수로 한 사람의 정보를 저장할 수 있습니다.
 - 두 번째 사람의 정보를 저장한다면 숫자를 바꿔 변수를 선언하고 저장하면 되겠지요?
 - 정보들 사이의 연관관계는 변수명 뒤의 숫자로 판단해야 합니다.
 - 조금 더 세련된 방법은 없을까요?
- 2차원 배열을 이용해 저장하는 경우
 - 한 행의 정보는 모두 한 사람의 정보입니다.
 - 다만, 각각의 열들이 어떤 정보인지 미리 정해둬야 합니다.
 - 예를 들어 person[0][0]은 이름, person[0][1]은 학과, person[0][2]는 학번처럼 관리해줘야 합니다.
 - 그냥 알아보기 쉽게 이름은 name, 학과는 department로 쓸 수는 없을까요?

구조체가 필요한 이유 (2/2)

- 구조체를 쓰면 이런 문제를 해결할 수 있습니다.
 - 학생의 정보를 저장하는 구조체 Person을 만들고 그 안에 저장하려는 정보를 멤버변수로 선언합니다.
 - ```
struct Person {
 string name;
 string department;
 ...
};
```
  - Person 타입의 변수를 배열로 선언합니다.
    - ```
struct Person students[MAX];
```
 - 이제 `students[0].name = "은재형"`과 같이 조금 더 읽기 편한 코드가 만들어 졌습니다.
- 이처럼 구조체는 데이터를 조금 더 체계적으로 다루기 위해 사용합니다.
- C++에서는 객체를 이용해 더 자유롭게 정의하는 방법이 존재합니다.
 - C++에서 구조체는 C언어와의 호환성을 위해 그 명맥을 유지하는 정도입니다.
 - 다만 객체를 학습하기 전, 개념을 익히기에는 조금 더 용이합니다.

예제 1. 학번과 이름 (revisited)

- 1주차에 진행한 학번과 이름 예제를 떠올려봅시다.
 - 지난 예제에서는 학번과 이름을 바로 출력했습니다.
 - 이번 예제에서는 구조체를 만들어 학번과 이름을 저장하고 이를 출력하는 프로그램을 작성해봅시다.
 - 우선 main()함수 위에 구조체를 정의합니다.
 - 이 구조체는 학번과 이름을 멤버변수로 갖습니다.
 - 학번과 이름은 각각 string 타입으로 선언합니다.
 - main()함수 안에서는 앞서 정의한 구조체를 이용해 구조체 변수를 하나 생성합니다.
 - cin으로 구조체 변수의 학번과 이름에 각각 값을 입력합니다.
 - 이제 구조체의 데이터를 출력합니다.
 - 구조체 변수에 저장한 화면에 학번과 이름을 출력합니다.
 - 학번과 이름 사이에는 공백 문자를 하나 출력해야 합니다.

```
➤ ./main
1234567890
은재형
1234567890 은재형
```

구조체 중첩

- 구조체의 멤버변수로 구조체 변수를 사용할 수도 있습니다.
 - (단방향 컴파일러인 점을 고려해서) 맴버변수로 사용할 구조체를 먼저 정의해야 합니다.
- 예를 들어 학교에 대한 정보를 관리하는 구조체를 생각해봅시다.
 - 학교 정보에는 학교이름, 종류, 주소, 전화번호 등이 포함될 수 있습니다.
 - ```
struct School {
 string name;
 int type;
 string address;
 string tel;
};
```
- 이력서 프로그램이 있다면, 위의 학교 정보 구조체를 이용해 최종학력 등을 관리하기 좋을 것입니다.
  - ```
struct CV {  
    School academic_background;  
    string name; ...  
};
```

구조체를 함수에서 사용하기

- 구조체를 인자로 사용하는 방법은 일반적인 변수를 사용하는 것과 동일합니다.
 - 구조체를 정의하고, 함수의 인자 자리에 "구조체명 변수명" 순으로 적습니다.
 - 예를 들어 `struct Person`이라는 구조체를 정의했다면...
 - `void displayData(Person p);` 처럼 함수의 인자로 사용할 수 있습니다.
- 함수의 결과로 구조체 돌려받기
 - 구조체를 돌려 받으려면 반환형에 구조체 이름을 적고, `return` 문으로 구조체 변수를 반환하면 됩니다.
 - 위와 동일하게 구조체가 정의되어 있다면, `Person getData();` 처럼 사용할 수 있습니다.

예제 2. 명함관리 프로그램

- 사용자로부터 명함의 정보를 입력받아 관리하는 프로그램을 작성해봅시다.
 - 명함에는 "이름", "전화번호", 이메일" 등이 포함될 수 있습니다.
 - main() 함수에서는 명함 구조체 변수를 MAX개 생성합니다.
 - 메뉴를 출력하고 그에 따라 동작합니다.
 - 메뉴에는 입력, 제거, 출력 등이 포함될 수 있습니다.
- 제한 사항
 - 각 명함의 정보를 체계적으로 관리하기 위해 구조체로 만들어 봅시다.
 - 명함정보를 입력하는 함수로부터 결과를 반환 받아 변수에 저장해야 합니다.
 - 명함정보를 인자로 받아 화면에 출력하는 함수를 만들고 사용해야 합니다.

출력 예시 (1/2)



- 명함 입력

```
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 1
이름을 입력하세요: 강동한
전화번호를 입력하세요: 01011112222
이메일을 입력하세요: dhkang@naver.com
추가한 명함의 번호는 0입니다.
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 1
이름을 입력하세요: 구도연
전화번호를 입력하세요: 01033334444
이메일을 입력하세요: dhkoo@gmail.com
추가한 명함의 번호는 1입니다.
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 3
번호: 0
이름: 강동한
전화번호: 01011112222
이메일: dhkang@naver.com
=====
번호: 1
이름: 구도연
전화번호: 01033334444
이메일: dhkoo@gmail.com
=====
```

- 명함 삭제

```
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 2
삭제할 명함의 번호를 입력하세요: 0
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 3
번호: 0
이름: 구도연
전화번호: 01033334444
이메일: dhkoo@gmail.com
=====
```

출력 예시 (2/2)



- 삭제 후 입력

```
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 1
이름을 입력하세요: 김규리
전화번호를 입력하세요: 01055556666
이메일을 입력하세요: grkim@hanyang.ac.kr
추가한 명함의 번호는 1입니다.
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 1
이름을 입력하세요: 김익제
전화번호를 입력하세요: 01077778888
이메일을 입력하세요: ijkim@kakao.com
추가한 명함의 번호는 2입니다.
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 3
번호: 0
이름: 구도연
전화번호: 01033334444
이메일: dhkoo@gmail.com
=====
번호: 1
이름: 김규리
전화번호: 01055556666
이메일: grkim@hanyang.ac.kr
=====
```

- 추가 입력 결과

```
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 3
번호: 0
이름: 구도연
전화번호: 01033334444
이메일: dhkoo@gmail.com
=====
번호: 1
이름: 김규리
전화번호: 01055556666
이메일: grkim@hanyang.ac.kr
=====
번호: 2
이름: 김익제
전화번호: 01077778888
이메일: ijkim@kakao.com
=====
1. 입력, 2. 제거, 3. 출력, 4. 종료
메뉴를 입력하세요: 4
```

연습문제. 도서관 프로그램

- 책을 대여, 반납, 추가, 제거, 목록출력 할 수 있는 프로그램을 만들어봅시다.
- 이를 위해 우선 책의 정보를 관리하는 구조체를 만들어야 합니다.
 - 책 구조체는 "책이름, 저자이름, 대여여부"가 포함되어야 합니다.
- main() 함수에서는 책 구조체를 MAX개 만큼 선언하고 사용자에게 메뉴를 출력합니다.
- 사용자가 메뉴를 입력하면, 그에 따라 특정 동작을 수행합니다.
 - 대여를 선택하면 목록에 있는 책을 대여합니다.
 - 대여하려면 책의 번호를 입력해야 하며, 이미 대여된 책은 추가로 대여할 수 없습니다.
 - 반납을 선택하면 대여한 책을 반납합니다.
 - 반납하려면 책의 번호를 입력해야 하며, 대여된 책만 반납할 수 있습니다.
 - 추가를 선택하면 새로운 책을 추가합니다.
 - 제거를 선택하면 기존의 책을 제거합니다.
 - 목록출력은 현재 저장된 책의 목록을 대여 여부와 함께 출력합니다.

