

ECE20023, Spring 2021



오픈소스 소프트웨어 실습



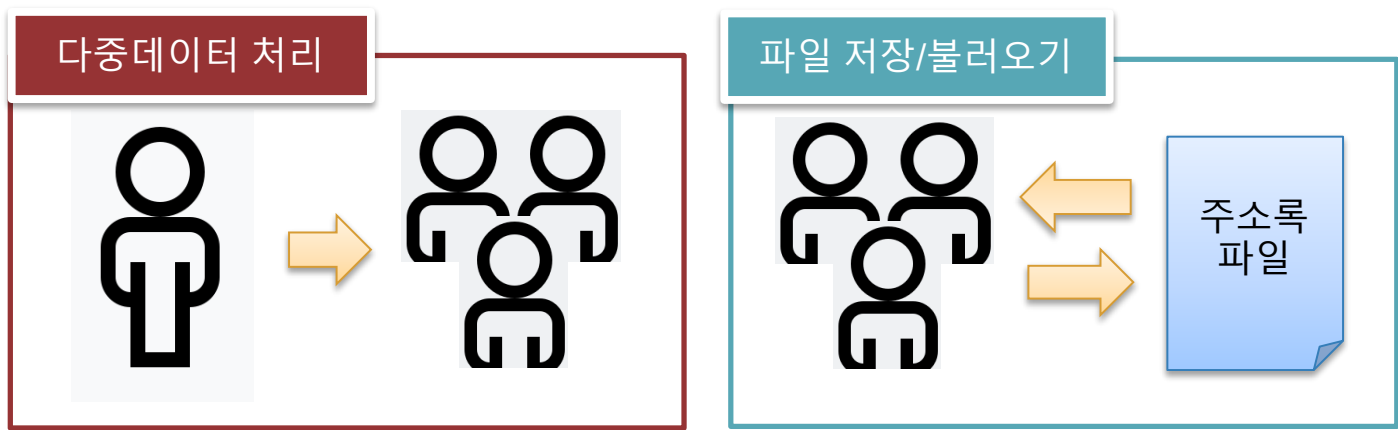


4

CRUD, 다중데이터 (C Programming)

CRUD

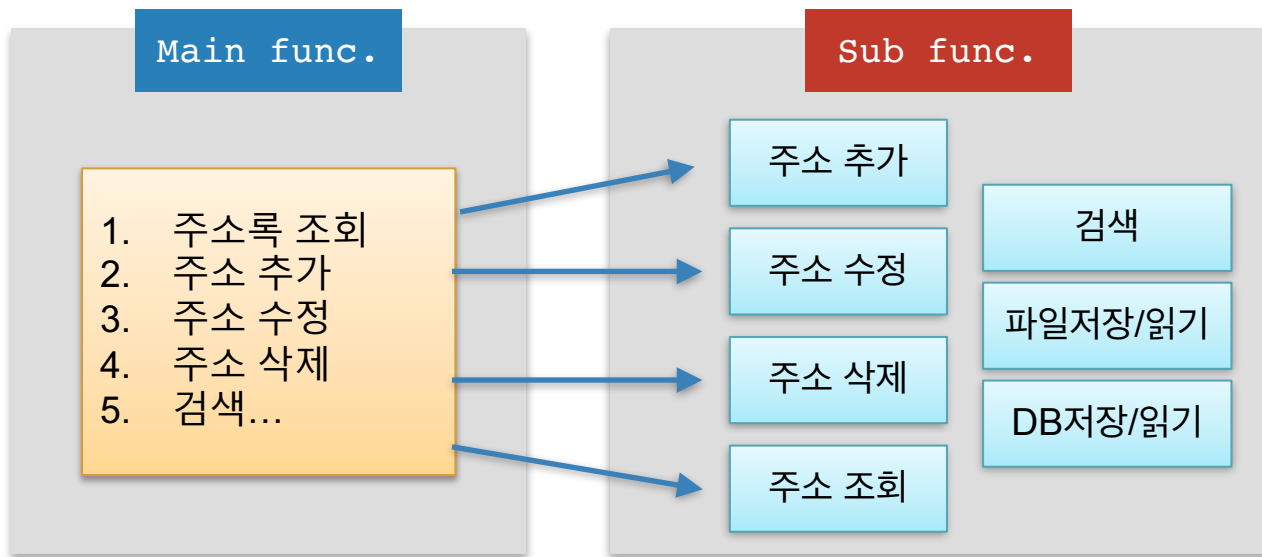
- Create(추가), Read(조회), Update(수정), Delete(삭제)
- 다중 데이터 처리
- 데이터 저장 / 불러오기 기능(File or Database)
- 부가기능 : 검색, 보고서 기능



CRUD Structure



주소록 프로그램



CRUD 예제 : 성적관리

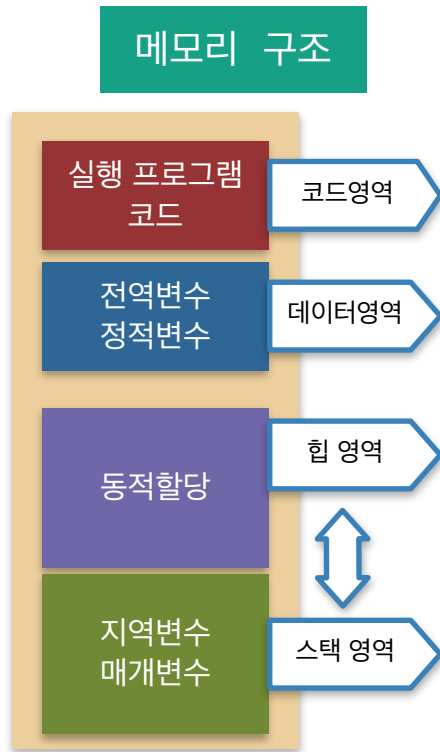
이름은?	홍길동
국어는?	90
영어는?	90
수학은?	95

1. 사용자 데이터 타입 정의
2. 하나의 데이터를 추가하는 기능(함수) 구현 (C)
3. 하나의 데이터 조회 기능(함수) 구현(R)
4. 하나의 데이터 수정 기능(함수) 구현(U)
5. 하나의 데이터 삭제 기능(함수) 구현(D)
6. 각 함수를 사용하여 전체 프로그램을 구성하는 main 함수 구현
7. 다중 데이터 처리
8. 데이터 파일로 저장
9. 파일 데이터 가져오기 기능

CRUD 예제 : 성적관리

7. 다중 데이터 처리

- 변수 배열 , 포인터 배열 변수
- 포인터 배열 변수 사용시 데이터가 추가될 때 동적 메모리 할당 (malloc)
 - **malloc 함수** : 힙 영역에 메모리를 동적으로 할당
 - **free 함수** : 힙 영역에 할당된 메모리를 해제



CRUD 예제 : 성적관리 : 다중데이터(배열)

1. 다중 데이터 처리를 위한 배열 변수 선언(데이터 수 고정)
2. 데이터 추가 함수 호출 수정
3. 리스트 함수 추가(listScore)
4. 수정 함수 호출 수정
5. 삭제 함수 호출 수정

CRUD 예제 : 성적관리 : 다중데이터 (배열)

1. 다중데이터 처리를 위한 배열 변수 선언

- 배열 사용시 현재 위치(index)와 개수(count) 변수가 필요함

```
int main(void) {  
    int result = 0;  
    Score s1;  
  
    result = addScore(&s1);  
  
    return 0;  
}
```



```
int main(void) {  
    int count = 0; //데이터개수  
    int result = 0;  
  
    Score slist[100];  
    int index = 0; //데이터번호  
  
    result = addScore(&slist[index]);  
  
    return 0;  
}
```


CRUD 예제 : 성적관리 : 다중데이터 (배열)

2. 데이터 추가 함수 호출 수정

- 배열 사용시 현재 위치(index)와 개수(count) 변수가 필요함

```
int addScore(Score *s);  
int main(void) {  
    ...  
    else if(menu == 2) addScore(&s1);  
    ...  
    return 0;  
}
```



```
int main(void) {  
    ...  
    else if(menu == 2)  
        count += addScore(&slist[index++]);  
    ...  
    return 0;  
}
```

slist[0]

slist[1]

slist[2]

slist[3]

...

CRUD 예제 : 성적관리 : 다중데이터 (배열)

3. 리스트 함수 추가(listScore)

- 기존 readScore() 함수 이용
- listScore() / 데이터배열, 배열개수/결과값 없음

No	Name	Kor	Eng	Math	Sum	Avg
=====						
1	철수	100	100	100	300	100.0
2	영희	90	95	100	285	95.0
3	서연	100	95	90	285	95.0

```
void listScore(Score *s, int count){
    printf("\nNo Name Kor Eng Math Sum Avg\n");
    printf("=====\n");
    for(int i = 0; i < count ; i++){
        if(slist[i].kor == -1) continue;
        printf("%2d ", i+1);
        readScore(slist[i]);
    }
    printf("\n");
}
```



```
int main(void) {
    ...
    if(menu == 1)
        if(count > 0)
            listScore(slist, index);
    ...
    return 0;
}
```

CRUD 예제 : 성적관리 : 다중데이터 (배열)

4. 수정 함수 호출 수정

- 수정하기 위한 리스트 보여준 후 선택한 번호 수정
- selectDataNo() / 데이터배열, 개수 / 선택한 번호 리턴

=> 원하는 메뉴는? 3

No	Name	Kor	Eng	Math	Sum	Avg
1	철수	100	100	100	300	100.0
2	영희	90	95	100	285	95.0
3	서연	100	95	90	285	95.0

번호는 (취소 :0)? 1

이름은?김철수

국어는?95

영어는?95

수학은?95

=> 수정됨!

```
int selectDataNo(Score *s, int count){
    int no;
    listScore(s, count);
    printf("번호는 (취소 :0)? ");
    scanf("%d", &no);
    return no;
}
```



```
int main(void) {
    ...
    else if(menu == 3){
        int no = selectDataNo(slist, index);
        if(no == 0){
            printf("=> 취소됨!\n");
            continue;
        }
        updateScore(&slist[no-1]);
    }
    ...
    return 0;
}
```

CRUD 예제 : 성적관리 : 다중데이터 (배열)

5. 삭제 함수 호출 수정

- 삭제하기 위한 리스트 보여준 후 선택한 번호 수정
- 수정에서 리스트 보여주는 함수 사용(selectDataNo)
- 삭제한 데이터는 보이지 않도록 처리(listScore)

=> 원하는 메뉴는? 4

No	Name	Kor	Eng	Math	Sum	Avg
1	김철수	95	95	95	285	95.0
2	영희	90	95	100	285	95.0
3	서연	100	95	90	285	95.0

번호는 (취소 :0)? 1

정말로 삭제하시겠습니까?(삭제 :1)1

=> 삭제됨!

```
int main(void) {  
  
    int no = selectDataNo(slist, index);  
    if(no == 0){  
        printf("> 취소됨!\n");  
        continue;  
    }  
    int deleteok;  
    printf("정말로 삭제하시겠습니까?(삭제 :1)");  
    scanf("%d", &deleteok);  
    if(deleteok == 1){  
        if(deleteScore(&slist[no-1])) count--;  
    }  
}
```

CRUD 예제 : 성적관리 : 다중데이터(포인터배열)

1. 다중 데이터 처리를 위한 포인터배열 변수 선언
2. 데이터 추가 함수 호출 수정
3. 리스트 함수 추가(listScore)
4. 수정 함수 호출 수정
5. 삭제 함수 호출 수정

CRUD 예제 : 성적관리 : 다중데이터(포인터배열)

1. 다중데이터 처리를 위한 포인터 배열 변수 선언

```
int main(void) {  
  
    int count = 0;  
    int result = 0;  
  
    Score slist[100];  
    int index = 0;  
  
    return 0;  
}
```



```
int main(void) {  
    int count = 0;  
    int result = 0;  
  
    Score *sp[100];  
    int index = 0;  
  
    result = addScore(&slist[index]);  
  
    return 0;  
}
```

CRUD 예제 : 성적관리 : 다중데이터(포인터배열)

2. 데이터 추가 함수 호출 수정

```
int main(void) {  
    ...  
    else if(menu == 2) addScore(&s1);  
    ...  
    return 0;  
}
```



```
#include <stdio.h>  
#include <stdlib.h>
```

```
int main(void) {  
    ...  
    else if(menu == 2){  
        sp[index] = (Score *)malloc(sizeof(Score));  
        count += addScore(sp[index++]);  
    }  
    return 0;  
}
```

CRUD 예제 : 성적관리 : 다중데이터(포인터배열)

3. 리스트 함수 추가(listScore)

- 조회함수(readScore)함수를 이용하여 리스트보기 함수 구현

No	Name	Kor	Eng	Math	Sum	Avg
=====						
1	철수	100	100	100	300	100.0
2	영희	90	95	100	285	95.0
3	서연	100	95	90	285	95.0

```
void listScore2(Score *s[], int count){
    printf("\nNo Name Kor Eng Math Sum Avg\n");
    printf("===== \n");
    for(int i=0; i < count ; i++){
        if(s[i]->kor == -1) continue;
        printf("%2d ", i+1);
        readScore(*s[i]);
    }
    printf("\n");
}
```



```
int main(void) {
    Score *sp[100];
    ...

    if(menu == 1){
        if(count > 0)
            listScore2(sp, index);
        else
            printf("데이터가 없습니다.\n");
    }
    ...

    return 0;
}
```


CRUD 예제 : 성적관리 : 다중데이터(포인터배열)

4. 수정 함수 호출 수정

- 수정하기 위한 리스트 보여준 후 선택한 번호 수정

=> 원하는 메뉴는? 3

No	Name	Kor	Eng	Math	Sum	Avg
1	철수	100	100	100	300	100.0
2	영희	90	95	100	285	95.0
3	서연	100	95	90	285	95.0

번호는 (취소 :0)? 1

이름은?김철수

국어는?95

영어는?95

수학은?95

=> 수정됨!

```
int selectDataNo2(Score *s[], int count){
    int no;
    listScore2(s, count);
    printf("번호는 (취소 :0)? ");
    scanf("%d", &no);
    return no;
}
```



```
int main(void) {
    ...
    else if(menu == 3){
        int no = selectDataNo2(sp, index);
        if(no == 0){
            printf("=> 취소됨!\n");
            continue;
        }
        updateScore(sp[no-1]);
    }
    ...
    return 0;
}
```

CRUD 예제 : 성적관리 : 다중데이터(포인터배열)

5. 삭제 함수 호출 수정

- 수정에서 리스트 보여주는 함수 사용(selectDataNo2)

```
int main(void) {
    ...

    int no = selectDataNo2(sp, index);
    if(no == 0){
        printf("=> 취소됨!\n");
        continue;
    }
    int deleteok;
    printf("정말로 삭제하시겠습니까?(삭제 :1)");
    scanf("%d", &deleteok);
    if(deleteok == 1){
        if(sp[no-1]) free(sp[no-1]);
        sp[no-1] = NULL;
        count--;
    }
    ...
}
```

```
void listScore2(Score *s[], int count){
    printf("\nNo Name Kor Eng Math Sum Avg\n");
    printf("=====\n");
    for(int i =0; i <count ; i++){
        if(s[i] == NULL) continue;
        printf("%2d ", i+1);
        readScore(*s[i]);
    }
    printf("\n");
}
```



요약

1. 배열 변수 사용
2. 배열 변수와 반복문
3. 포인터 변수의 이해
4. 포인터 배열 변수의 이해
5. 동적 메모리 할당 / 해제하는 법
6. 다수 데이터 관리방법

과제설명

다중 데이터 다루기(금요일 수업)

3. [3번. 결과 참고 자료]와 같은 결과가 나오도록, 1번 문제에서 CRUD를 구현하여 만든 본인의 소스를 수정하여 다중 데이터를 처리할 수 있도록 배열버전과 포인터배열 버전으로 개발하고, 개발 코드 및 결과화면을 캡처하여 제출하세요.(vscode로 개발).(배열버전: score2.c(20명까지 관리), 포인터배열버전 : score3.c)
4. (응용문제) 2번에서 본인이 개발한 레스토랑 메뉴관리 프로그램에 다중 데이터를 처리할 수 있도록 배열버전과 포인터배열 버전으로 완성하고, 해당 코드와 결과화면을 캡처하여 제출하세요.(vscode사용, 결과는 3번 프로그램 참조, menu2.c, menu3.c)
5. LAB1에서 제출한 dynalist 본인 링크에 아래 항목을 추가하여 문법 정리한 후 링크 제출합니다.(sharing 필수), 검색한 링크를 추가해도 됩니다. 이전에 만든것도 수정해도 됩니다. (본인이 추후 계속사용할 수 있도록 알찬 문서로 만들어 보세요)
참조링크 : <https://dynalist.io/d/ArHusIKz79zSckv8DTwae31T>
 - 함수 생성 방법
 - call by value / call by address
 - 구조체, typedef 차이
 - 변수와 포인터변수의 차이
 - 포인터 변수로 구조체 자료형 멤버변수 접근
 - 구조체 변수 멤버변수 접근방법
 - 문자열 입력
 - 공백이 있는 문자열 입력방법
 - 입력버퍼
 - 포인터 변수
 - 포인터 배열 변수
 - 포인터 배열 변수에 메모리 할당하는 법
 - 메모리 해제
 - C언어의 동적배열과 고정배열