

4장. 배열과 구조체



Visualstudio 2019



배열(Array)

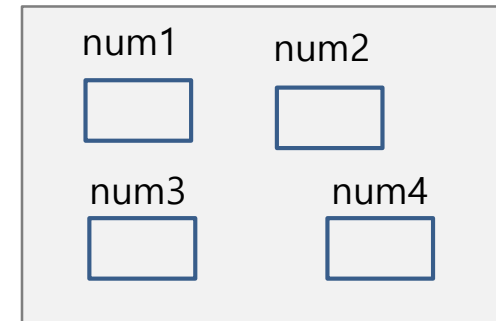
❖ 배열은 왜 써야 할까?, 사용의 필요성

- 정수 10개를 이용한 프로그램을 할 때 10개의 변수를 선언

`int num1, int num2, int num3... num10;`

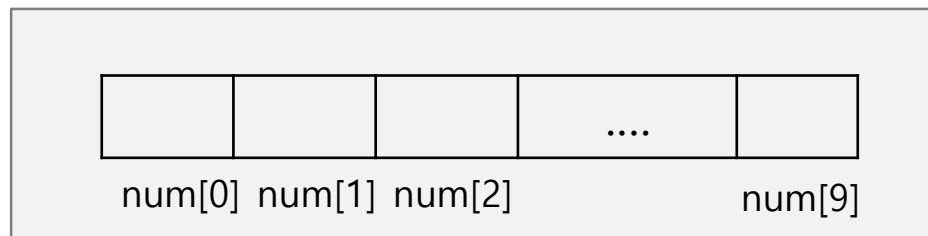
연산이나 자료의 교환 등이 불편하다.

메모리



- 배열은 동일한 자료형의 변수를 한꺼번에 순차적으로 관리할 수 있다.

메모리



배열(Array)

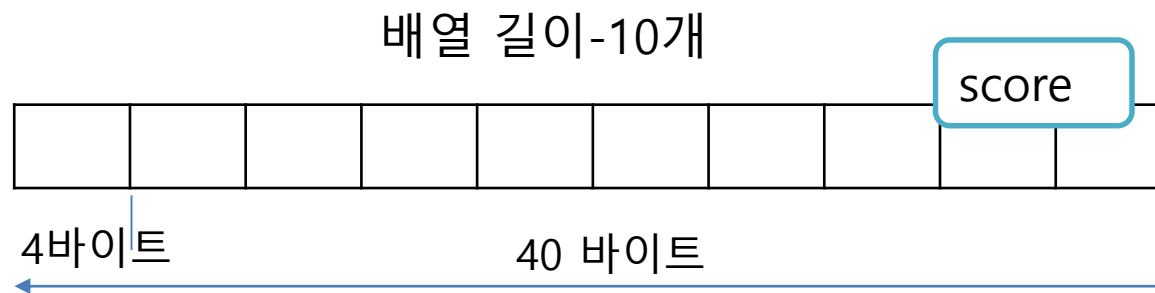
- 배열이란?

여러 개의 동일한 자료형의 데이터를 한꺼번에 만들때 사용한다.

배열 변수는 []안에 설정한 값만큼 메모리를 할당하여 저장한다.

- 배열 변수의 선언과 사용

```
int score[10];
```



배열(Array)

- 문자형 배열

```
char alphabets[26];
char ch = 'a';

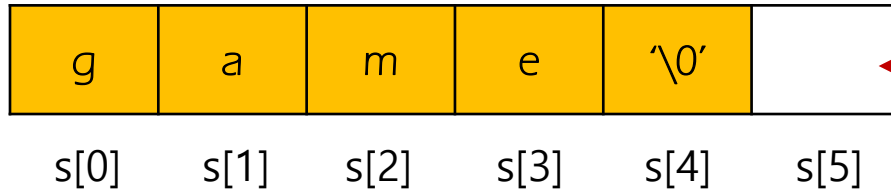
//저장
for (int i = 0; i < 26; i++) {
    alphabets[i] = ch;
    ch++;
}

//출력
for (int i = 0; i < 26; i++) {
    cout << alphabets[i] << ", " << (int)alphabets[i] << endl;
}
```



배열(Array)

■ 문자열 배열



문자열의 끝을 나타내는 공백
문자 '\0' 이 자동으로 추가됨

문자열 함수	설명	예제
strlen()	문자열의 개수	s1 = "game" \0을 제외하고 4개
strcpy()	문자열 복사	char s2[10]; strcpy(s2, "over")
strcmp()	문자열 비교	strcmp("aa", "aa") -> 0 반환 strcmp("aa", "bb") -> -1 strcmp("bb", "aa") -> 1



배열(Array)

- 문자열 배열

```
char s1[] = "game", s2[20];

//문자열 개수 - '\0'(공백문자)는 포함되지 않음
cout << "s1의 개수 : " << strlen(s1) << endl;

//문자열 복사
//s2[20] = "over";
strcpy_s(s2, "over");

cout << "s1 = " << s1 << ", s2 = " << s2 << endl;

//문자열 비교
cout << strcmp(s1, s2) << endl;
//다른 경우 -1(문자가 앞에 있으면), 1(뒤에 있으면)
cout << strcmp(s1, "game") << endl; //같으면 0을 반환
```



배열(Array)

- 최대값 구하기

```
int arr[10] = { 2, 71, 59, 33, 94, 25, 85, 9, 24, 11 };
char str[] = "game over !";

//최대 정수
int maxVal = arr[0];
for (int i = 1; i < 10; i++)
    if (maxVal < arr[i]) maxVal = arr[i];

cout << "max val = " << maxVal << endl;

//최대 문자
cout << strlen(str) << endl;

char maxChar = str[0];
for (int i = 1; i < strlen(str); i++)
    if (maxChar < str[i]) maxChar = str[i];

cout << "max char = " << maxChar << endl;
```



구조체(Struct)의 개념

◆ 구조체는 왜 필요할까?

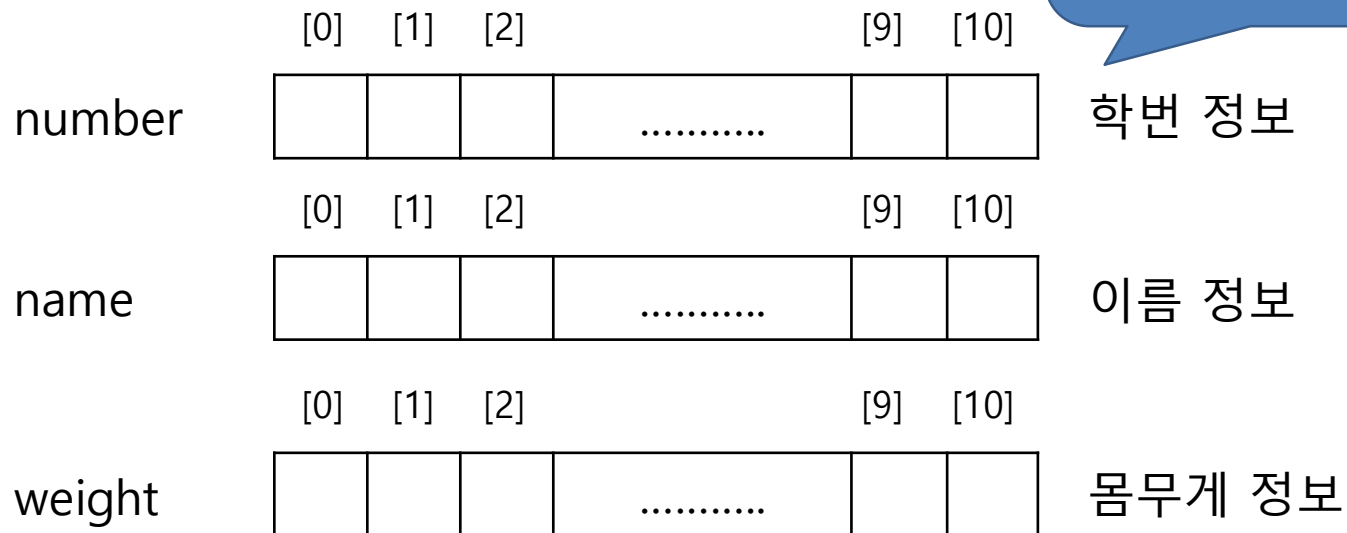
학생 10명의 학번과 이름, 몸무게 정보 저장 – 배열 자료형 이용

```
int number[10];
```

```
char name[20];
```

```
double weight[10];
```

정보가 흩어져서 저장되는 한계 발생



◆ 구조체(structure)란?

다양한 자료형을 그룹화하여 하나의 변수로 처리할 수 있게 만든 자료형이다. 개발자가 다양한 정보를 저장하기 위해 필요에 따라 생성하는 자료형을 사용자 정의 자료형 또는 구조체라 한다.

- 구조체 정의

```
struct 구조체이름{  
    자료형 멤버이름;  
};
```

- 객체 생성

```
구조체이름 변수이름;
```



C++ 구조체

◆ 구조체(structure)란?

```
#include <iostream>
#include <string>
using namespace std;

struct Student {
    string name;    //이름
    int grade;      //학년
    string address; //주소
};
```

```
int main() {
    Student st1 = Student(); //기본 생성자로 객체 생성
    //struct는 public 이므로 멤버 접근 허용됨.
    st1.name = "이우주";
    st1.grade = 3;
    st1.address = "서울시 노원구 상계동";

    cout << "학생 이름 : " << st1.name << endl;
    cout << "학년 : " << st1.grade << endl;
    cout << "주소 : " << st1.address << endl;

    return 0;
}
```



구조체 배열

◆ 구조체 배열 – 객체를 여러 개 생성

```
//구조체 배열
Student studentArr[3] = {
    {"김지구", 1, "서울시 종로구"},
    {"박화성", 2, "서울시 은평구"},
    {"최목성", 3, "서울시 강남구"}
};

for (int i = 0; i < 3; i++) {
    cout << studentArr[i].name << " 학생은 "
         << studentArr[i].grade << "학년 입니다." << endl;
}
```



열거형 자료형 enum

enum 자료형

- enumeration(열거하다)의 영문 약자 키워드로, 사용자가 직접 정의하여 사용할 수 있는 자료형이다.
- 열거형은 정수형 상수에 이름을 붙여서 코드를 이해하기 쉽게 해줌
- 열거형은 상수를 편리하게 정의할 수 있게 해줌

```
const int VALUE_A = 1;  
const int VALUE_B = 2;  
const int VALUE_C = 3;
```



```
enum VALUE{  
    VALUE_A = 1,  
    VALUE_B  
    VALUE_C  
}
```

상수의 개수가 많아지면 선언
하기 힘들어짐



열거형 자료형 enum

enum 자료형

```
enum 열거형 이름{  
    값1 = 초기값,  
    값2,  
    값3  
}
```



사용
enum 열거형 이름 변수 이름

```
enum WEEK {  
    SUN = 1, //초기값  
    MON,  
    TUE,  
    WED,  
    THR,  
    FRI,  
    SAT  
};
```

```
int main() {  
    enum WEEK week; //선언  
    week = WED;  
    //int week = WED; 사용 가능  
  
    cout << week << endl;  
  
    return 0;  
}
```



열거형 자료형 enum

- switch ~ case 문에서 사용하기

```
enum MEDAL {  
    GOLD = 1,  
    SILVER,  
    BRONZE  
};
```

```
//enum MEDAL medal;  
//medal = SILVER;  
int medal;    //주의! cin으로 입력할 경우 변수 선언  
  
cout << "1 ~ 3중에 입력하세요:";  
cin >> medal;  
  
switch (medal) {  
case GOLD:  
    cout << "금메달입니다." << endl;  
    break;  
case SILVER:  
    cout << "은메달입니다." << endl;  
    break;  
case BRONZE:  
    cout << "동메달입니다." << endl;  
    break;  
default:  
    cout << "메달이 없습니다." << endl;  
    break;  
}
```



성적 관리 프로그램

● 구조체를 사용하여 성적 관리 프로그램 만들기

```
1. 학생등록
2. 학생검색
3. 학생삭제
4. 학생목록
5. 종료
```

```
메뉴를 선택하세요 : 1
===== 학생 정보 추가 =====
이름 : 1
주소 : 김하나
전화번호 : 010-1234-5678
국어 : 70
영어 : 80
수학 : 90
```

```
메뉴를 선택하세요 : 2
===== 학생 정보 검색 =====
검색할 이름을 입력하세요 : 김하나
학번 : 1
이름 : 김하나
전화번호 : 010-1234-5678
주소 : 서울시 노원구
국어 : 70
영어 : 80
수학 : 90
총점 : 240
평균 : 80
```

```
메뉴를 선택하세요 : 3
===== 학생 정보 삭제 =====
삭제할 이름을 입력하세요 : 이둘
학생을 삭제하였습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```
메뉴를 선택하세요 : 4
===== 학생 목록 =====
학번 : 1
이름 : 김하나
전화번호 : 010-1234-5678
주소 : 서울시 노원구
국어 : 70
영어 : 80
수학 : 90
총점 : 240
평균 : 80

학번 : 2
이름 : 이둘
전화번호 : 010-2222-3333
주소 : 서울시 종로구
국어 : 75
영어 : 85
수학 : 95
총점 : 255
평균 : 85

계속하려면 아무 키나 누르십시오 . . .
```



성적 관리 프로그램

- student.h – Student 구조체, enum 자료형 생성

```
#pragma once
#include <iostream>
using namespace std;

#define STUDENT_MAX 10      //학생수 최대값
#define NAME_SIZE 32       //이름 최대값
#define ADDRESS_SIZE 128   //주소 최대값
#define PHONE_NUMBER_SIZE 14 //전화번호 최대값
```

```
struct Student {
    char name[NAME_SIZE];
    char address[ADDRESS_SIZE];
    char phone[PHONE_NUMBER_SIZE];
    int number;
    int kor;
    int eng;
    int math;
    int total;
    float avg;
};

enum MENU {
    MENU_INSERT = 1, //삽입
    MENU_LIST,      //목록
    MENU_SELECT,    //검색
    MENU_DELETE,    //삭제
    MENU_EXIT       //종료
};
```



성적 관리 프로그램

- student_score.cpp – 전역변수 선언 , insert() 함수

```
#include "student.h"

Student studentArr[STUDENT_MAX] = {}; //배열 객체 생성
int stdCount = 0; //인덱스 번호
int stdNumber = 1; //학번
char searchName[NAME_SIZE] = {};

void insert() {
    cout << "===== 학생 정보 추가 =====" << endl;
    //이름, 주소, 전화번호, 국어, 영어, 수학은 입력받고
    //학번, 총점, 평균은 연산을 통해 계산함

    cout << "이름 : ";
    cin >> studentArr[stdCount].name;

    cin.ignore(1024, '\n'); //주소를 한 줄로 입력 받음

    cout << "주소 : ";
    cin.getline(studentArr[stdCount].address, ADDRESS_SIZE);

    cout << "전화번호 : ";
    cin.getline(studentArr[stdCount].phone, PHONE_NUMBER_SIZE);

    cout << "국어 : ";
    cin >> studentArr[stdCount].kor;
```



성적 관리 프로그램

- student_score.cpp – insert() 함수

```
cout << "영어 : ";
cin >> studentArr[stdCount].eng;

cout << "수학 : ";
cin >> studentArr[stdCount].math;

//총점 계산
studentArr[stdCount].total =
    studentArr[stdCount].kor +
    studentArr[stdCount].eng +
    studentArr[stdCount].math;

//평균 계산
studentArr[stdCount].avg =
    (float)studentArr[stdCount].total / 3;

//학번
studentArr[stdCount].number = stdNumber;
stdNumber++;    //학번 1증가
stdCount++;     //인덱스 1증가

cout << "**** 학생 추가 완료 ****" << endl;
```



성적 관리 프로그램

- student_score.cpp – listAll() 함수

```
void listAll() {  
    cout << "===== 학생 목록 =====" << endl;  
    //등록된 학생 수만큼 반복하며 학생 정보 출력  
    for (int i = 0; i < stdCount; i++) {  
        cout << "학번 : " << studentArr[i].number << endl;  
        cout << "이름 : " << studentArr[i].name << endl;  
        cout << "전화번호 : " << studentArr[i].phone << endl;  
        cout << "주소 : " << studentArr[i].address << endl;  
        cout << "국어 : " << studentArr[i].kor << endl;  
        cout << "영어 : " << studentArr[i].eng << endl;  
        cout << "수학 : " << studentArr[i].math << endl;  
        cout << "총점 : " << studentArr[i].total << endl;  
        cout << "평균 : " << studentArr[i].avg << endl << endl;  
    }  
}
```



성적 관리 프로그램

- student_score.cpp – select() 함수

```
void select() {
    cout << "===== 학생 정보 검색 =====" << endl;
    cin.ignore(1024, '\n');
    cout << "검색할 이름을 입력하세요 : ";
    cin.getline(searchName, NAME_SIZE);

    //등록되어 있는 학생 수만큼 반복하며 학생을 찾음
    for (int i = 0; i < stdCount; i++) {
        if (strcmp(studentArr[i].name, searchName) == 0) {
            //저장된 이름과 검색 이름이 일치하면 출력
            cout << "학번 : " << studentArr[i].number << endl;
            cout << "이름 : " << studentArr[i].name << endl;
            cout << "전화번호 : " << studentArr[i].phone << endl;
            cout << "주소 : " << studentArr[i].address << endl;
            cout << "국어 : " << studentArr[i].kor << endl;
            cout << "영어 : " << studentArr[i].eng << endl;
            cout << "수학 : " << studentArr[i].math << endl;
            cout << "총점 : " << studentArr[i].total << endl;
            cout << "평균 : " << studentArr[i].avg << endl << endl;
            break;
        }
    }
}
```



성적 관리 프로그램

- student_score.cpp – remove() 함수

```
void remove() {
    cout << "===== 학생 정보 삭제 =====" << endl;
    cin.ignore(1024, '\n');
    cout << "삭제할 이름을 입력하세요 : ";
    cin.getline(searchName, NAME_SIZE);

    for (int i = 0; i < stdCount; i++) {
        if (strcmp(studentArr[i].name, searchName) == 0) {
            for (int j = i; j < stdCount - 1; j++) {
                studentArr[j] = studentArr[j + 1];
            }
            stdCount--; //학생 수 감소
            cout << "학생을 삭제하였습니다." << endl;
            break;
        }
    }
}
```



성적 관리 프로그램

- student_main.cpp – main() 함수

```
#include "student.h"

extern int stdCount; //외부변수 호출
void insert();
void listAll();
void select();
void remove();

int main() {
    while (true) {
        system("cls"); //화면 지우기

        //메뉴 출력
        cout << "1. 학생 등록" << endl;
        cout << "2. 학생 목록" << endl;
        cout << "3. 학생 검색" << endl;
        cout << "4. 학생 삭제" << endl;
        cout << "5. 종료" << endl;
        cout << "메뉴를 선택하세요 : ";
        int menu;
        cin >> menu;
```



성적 관리 프로그램

- student_main.cpp – main() 함수

```
//문자를 입력할 경우 에러 발생 -> 에러 버퍼를 비워주고,  
//입력 버퍼에 있는 '\n' 앞부분의 문자를 지워준다  
if (cin.fail()) {    //에러 발생한 경우  
    cin.clear();  
    cin.ignore(1024, '\n');  
}  
  
if (menu == MENU_EXIT)  
    break;  
  
switch (menu) {  
case MENU_INSERT:  
    // 크기가 최대가 되면 등록되지 않도록 막는다  
    if (stdCount == STUDENT_MAX)  
        break;  
    insert();    //삽입  
    break;  
  
case MENU_LIST:  
    listAll();    //목록  
    break;
```



성적 관리 프로그램

- student_main.cpp – main() 함수

```
        case MENU_SELECT:
            select(); //검색

            break;
        case MENU_DELETE:
            remove(); //삭제
            break;

        default:
            cout << "메뉴를 잘못 입력했습니다." << endl;
            break;
    }

    system("pause"); //화면 멈춤
}

return 0;
}
```

