

교학습내용

- **♀** 12.1 동적 할당 개요



할 학습목표

- 12.1 동적 메모리 할당이 무엇인지 이해한다.
- 12.2 동적 메모리 할당 함수 사용법을 익힌다.





C프로그래밍및실습





- 1. 동적 할당 개요
- 2. 동적 메모리 사용 절차 (1)
- 3. 동적 메모리 사용 예제
- 4. 기타 동적 메모리 할당 함수



1. 동적 할당 개요

☑ 정적 메모리 할당

- 지금까지 기억 장소 확보를 위해 사용한 방식은 변수 선언
- 변수는 모두 코드 작성 단계에서 정해지고, 할당되는 메모리 크기는 프로그램을 실행할 때 마다 일정

프로그램 실행 전에 변수의 메모리 할당 크기가 정해지고 메모리 할당 크기가 실행 도중 변하지 않는 방식 → 정적 할당



1. 동적 할당 개요



🔳 정적 할당 방식의 문제점

- 필요한 메모리 크기를 알 수 없는 경우, 충분히 큰 크기를 가정해서 선언해야 함
- 프로그램 실행 전, 메모리를 얼마나 필요로 할 것인지 정확하게 알 수 없는 경우, 메모리를 효율적으로 사용하기 어려움
- 어떤 회사의 웹 사이트에서 가입하는 회원의 ID를 저장해야 하는 배열을 선언해야 한다면?
 - ✔ 몇 명의 사람들이 회원가입을 할 지 정확하게 알 수 없음
 - → 충분히 많은 사람들이 회원가입을 할 것으로 가정
 - ✓예) int id [50000];
 - ✓ 회원 수가 1,000 명 밖에 안되는 경우 → 메모리 공간 낭비
 - ✓ 회원 수가 5만명을 넘는 경우 → 프로그램에서 배열 크기 조정 후 재 컴파일 필요



1. 동적 할당 개요

등적 (dynamic) 메모리 할당

- 프로그램을 실행하는 동안 결정되는 크기에 따라 메모리를 할당하는 방식
- 메모리를 효율적으로 사용할 수 있음
- 앞 웹사이트 상황에서, 크기가 5만인 메모리를 할당 받아 사용하다,
 - ✓ 회원수가 5만명을 넘어가면 → 10만명의 회원 정보를 저장할 수 있는 메모리를 새로이 할당 받아 사용
 - ✓ 회원이 1,000명으로 줄어들면 → 1,000명 정도의 회원 정보를 저장할 수 있는 메모리 새로이 할당 받아 사용
 - ✓ 기존 크기가 5만인 메모리는 시스템에 반납



메모리 사용 절차

- 메모리 사용을 위해서는 메모리를 **할당**하고 **해제**하는 과정 필요
 - ✓ 할당: 필요한 메모리를 시스템으로부터 받기
 - ✔ 사용:할당된 메모리 사용
 - ✓ 해제 : 사용이 끝난 메모리를 시스템에 반납
- 정적 할당 vs. 동적 할당
 - ✓ 정적 할당: 이 과정이 프로그램과 함수의 실행 및 종료에 따라 자동으로 수행됨
 - ✓ 동적 할당: 프로그래머가 필요한 과정을 코드에 명시적으로 작성해 주어야함



(교) 동적 할당 기본 예제

● 동적으로 메모리를 할당하고 해제하는 라이브러리 함수는 (stdlib.h) 헤더 파일에 선언

```
#include <stdlib.h> // 동적 메모리 관련 함수 사용을
                 // 위한 헤더
int main() {
  int *p = NULL; // 동적 메모리 접근을 위한 포인터 변수
  p = (int *) malloc( 5*sizeof(int) ); // 동적 메모리 할당
  p[0] = 1;
                // 동적 메모리 사용: 배열 형태
  *(p+2) = 3;
               // 동적 메모리 사용: 포인터 형태
  free(p);
                // 동적 메모리 해제
  return 0;
```



■ 동적 메모리 할당 : malloc() 함수

● 요청된 크기의 연속된 메모리 할당 (memory allocation)

함수 원형	<pre>void * malloc(unsigned int size);</pre>
함수 인자	✓ 할당받을메모리크기(바이트단위) ✓ 보통 sizeof() 연산자활용
기능 및 반환 값	 ✓ 할당 받은 메모리의 시작 주소를 반환 ✓ 할당에 실패하면 → NULL 반환 ✓ 반환 값의 유형: void형 포인터 (void *)

- ✓ void * : 특정 자료형을 나타내지 않는 주소를 의미(13.6절 참조)
- ✓ 주의!! void (반환 값이 없음을 의미)와 혼동하지 말자



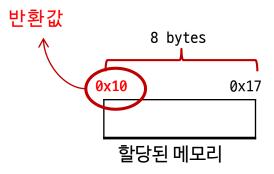
malloc() 함수 사용 예

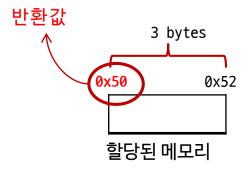
● 정수 2개를 저장할 공간 할당

● 문자 3개를 저장할 공간 할당

```
malloc( 3*sizeof(char) );
malloc( 3 ) ; 

⇒ 3 바이트 할당
```







동적 메모리 연결

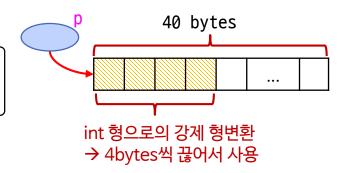
● 변수 이름을 붙일 수 없기 때문에, **포인터 변수를 이용하여 간접적으로 참조**

```
자료형 * 포인터_변수명 = NULL;
포인터_변수명 = (자료형 *) malloc(메모리_크기);
```

사용하고자 하는 자료형으로 명시적 형변환 할당된 메모리를 어떤 자료형으로 사용할 지 모르므로 (void *)형 반환

• 예시)

```
int *p = NULL;
p = (int *) malloc( 10*sizeof(int) );
```





등적 메모리 사용

- 할당된 메모리를 포인터 변수에 연결한 후에는 9장(포인터)에서 배운대로 사용
- 예시) 크기가 10인 배열 처럼 사용

학습정리

- 값을 저장하기 위해 필요한 기억 장소를 확보하는 것을 메모리 할당이라고 하고, 정적 메모리 할당과 동적 메모리 할당, 두 가지로 구분됨
- 동적 메모리 사용 절차는 동적 메모리 할당, 동적 메모리 사용, 동적 메모리 해제의 세 단계로 구성됨
- malloc() 함수는 메모리를 동적으로 할당하고, 인자로 할당할 메모리의 크기를 전달함
- 동적 할당된 메모리 영역은 이름이 없기 때문에, 포인터 변수에 연결하여 사용함

