

객체지향프로그래밍[02]

202304035 김유진

202104354 지현준

#문제 정의

mem 배열을 0으로 초기화하고 size를 초기화하는 생성자, 주어진 주소의 메모리 바이트를 리턴하는 read 멤버 함수, 한 주소에 값을 저장하는 write 멤버 함수, "메모리 제거됨"을 출력하는 소멸자를 구현한다. 이때, 선언부를 포함하는 헤더 파일과 구현부를 포함하는 cpp 파일(Ram.cpp, main.cpp) 2개로 분리하여 프로그램을 작성하는 문제다.

```
G+ main.cpp
G+ Ram.cpp
C Ram.h
```

#문제 해결 방법

1. mem 배열을 0으로 초기화하기 위해 for 반복문을 이용했다.
2. 프로그램이 종료될 때 호출되는 소멸자를 "메모리 제거됨"을 출력하도록 정의한다.
3. 주어진 주소의 메모리 바이트를 리턴해주도록 address를 인자로 가지고 char 타입의 메모리 바이트를 리턴해주는 read 멤버 함수를 구현한다.
4. 주어진 주소에 값을 저장하도록 address, value를 인자로 가지는 write 멤버 함수를 구현한다.

#아이디어 평가

1. for문을 사용하여 mem 배열을 0으로 초기화할 수 있었다.
2. mem 배열의 인덱스에 address로 접근하여 해당 주소에 대한 데이터를 얻어올 수 있었다.
3. mem 배열의 인덱스에 대해 주소(address)에 value 값을 대입하여 주어진 주소에 대해 데이터를 기록할 수 있었다.

아래 사진과 같이 문제의 출력값과 동일하게 출력되는 것을 확인할 수 있었다.

```
102 번지의 값 = 50
메모리 제거됨
```

#문제를 해결한 키 아이디어 또는 알고리즘 설명

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

#include "Ram.h"

int main() {
    Ram ram;
    ram.write(100, 20);
    ram.write(101, 30);
    char res = ram.read(100) + ram.read(101);
    ram.write(102, res);
    cout << "102 번지의 값 = " << (int)ram.read(102) << endl;
}
```

main.cpp 소스 코드

```

class Ram {
    char mem[100 * 1024];
    int size;
public:
    Ram(); //mem의 배열을 0으로 초기화하고 size를 100*1024로 초기화
    ~Ram(); // "메모리 제거됨" 문자열 출력
    char read(int address); //address 주소의 메모리 바이트 리턴
    void write(int address, char value); //address 주소에 한 바이트로 value 저장
};

```

Ram.h 소스 코드

```

#include <iostream>
using namespace std;

#include "Ram.h"

Ram::Ram() {
    for (int i = 0; i < 100 * 1024; i++)
        mem[i] = 0;
    size = 100 * 1024;
}

Ram::~Ram() {
    cout << "메모리 제거됨";
}

//address 주소의 메모리 바이트 리턴
char Ram::read(int address) {
    return mem[address];
}

//address 주소에 한 바이트로 value 저장
void Ram::write(int address, char value) {
    mem[address] = value;
}

```

Ram.cpp 소스 코드

우선, 소멸자는 프로그램이 종료될 때 호출되므로 소멸자에 어떤 문자열을 출력하는 명령이 있다면 가장 마지막에 해당 문자열이 출력된다.

문제의 실행 결과에서 “메모리 제거됨”이라는 문구를 출력했기 때문에 소멸자 함수를 직접 구현하여 객체가 소멸될 때 “메모리 제거됨” 문구가 출력되도록 만들었다.

따라서, 문제에서 ~Ram() 소멸자 함수를 직접 생성한 것이 키 포인트라고 생각한다.