

문제 정의

이 과제는 메모리 작동을 시뮬레이션 하는 프로그램 개발을 목표로 합니다. 주요 요구 사항은 100KB 크기의 메모리를 시뮬레이션하고, 이 메모리에 대해 데이터를 읽고 쓸 수 있는 기능을 제공하는 것입니다. 사용자는 주어진 메모리 주소에 데이터를 저장하고, 저장된 데이터를 읽을 수 있어야 합니다. 그러기 위해 우리에게 주어진 파일이 무엇이고 어떤 파일을 작성해야 하는지 살펴봅시다.

주어진 조건에 따르면 **Ram.h**, **Ram.cpp**, **main.cpp** 파일로 구분하여 작동하도록 해야 합니다. 그럼 각 파일들은 어떤 역할을 해야 하나 보자면

-Ram.h는 클래스의 정의를 제공합니다.

-Ram.cpp는 Ram.h에 정의된 클래스의 기능을 구현해야 합니다.

-main.cpp는 클래스의 기능을 사용합니다

먼저 주어진 두 파일을 살펴보면

클래스를 선언하니 **Ram.h** 역할 즉 메모리 작업을 위한 인터페이스를 정의하고, 다른 소스 파일에서 이 클래스를 참조할 수 있도록 하는 파일 하나와

Ram을 생성 및 사용하는 예제 코드를 주는 즉 **main.cpp**의 역할을 하는 코드가 주어졌습니다

그러니 우리는 Ram 클래스의 메소드 구현을 포함하는 즉 실제로 메모리를 조작을 구현하는 **Ram.cpp** 파일을 만드는 것이 목표입니다

문제 해결 방법

문제 해결방법은 위에서 말했듯 헤더파일인 **Ram.h**와 메인 프로그램 파일인 **main.cpp**가 제공되어 있으니 **Ram.h**에 선언된 Ram 클래스의 객체 생성 시 메모리를 초기화하고, 객체 소멸 시에 적절한 메시지를 출력하여 메모리가 해제되었음을 확인해주는 소멸자, 생성자 주소 값을 입력 받아 해당 위치에 데이터 read, write의 작동을 위한 구현파일인 **Ram.cpp**를 만들면 됩니다

아이디어 평가

구현하는 도중 소멸자, 생성자, 읽기, 쓰기에는 문제가 없었지만 컴파일 오류가 남 Ram 클래스에서 **Ram.h**가 다른 여러 소스 파일에 포함될 때 각 소스 파일마다 Ram 클래스의 정의를 중복해서 포함하게 되므로 문제가 생긴 듯합니다. Ram.h 파일에 헤더가드를 사용하니 정상적으로 작동했습니다.

문제를 해결한 키 아이디어 또는 알고리즘 설명

캡슐화와 제공된 코드의 분석이 이 문제를 해결하기 위한 키 아이디어였습니다.

객체 지향 프로그래밍의 관점으로 주어진 파일들을 보니 주어진 2개의 파일들이 어떤 역할을 하는지 알 수 있었고 Ram 클래스 메모리 관리를 캡슐화 하여, 외부에서 메모리에 직접 접근하는 것을 제한하는 것을 생각하니 빠르게 문제를 해결할 수 있었습니다.