

객체지향프로그래밍 소스 구현 설명

ICT공학부

202104361

천승환

1. 문제 정의 : 컴퓨터의 주기억장치를 모델링하는 클래스를 구현하는 것.

Ram클래스는 메모리 공간을 나타내는 데이터 구조를 가지고, 특정 address에 데이터를 read & write 할 수 있는 기능을 알려준다.

2. 문제 해결 방법 : Ram.cpp의 Ram클래스의 메서드 구현. mem배열 0으로 초기화하기 위해 for반복문 사용. 주어진 주소의 데이터를 반환하기 위해 return값 할당. 그 주소를 한 바이트로 value에 저장.

3. 아이디어 평가 : Ram.cpp의 Ram클래스의 메서드를 구현하였다. Ram()에서 mem배열을 0으로 초기화하기 위해 for반복문을 사용하였고, size를 100*1024로 할당하였다. 객체가 사라질 때 마무리 작업을 위해 ~Ram()에서 "메모리 제거됨" 문자열을 출력하도록 하였다. 데이터를 읽어오는 read(int address) 메서드를 구현하여 주어진 주소의 데이터를 반환하였다. 마지막으로, 데이터의 저장을 위한write(int address, int data) 메서드를 구현하여 주어진 주소에 데이터를 기록하였다.

4. 문제를 해결한 키 아이디어 또는 알고리즘 설명 :

```
for(int i = 0; i < mem[100 * 1024]; i++) {
```

```
    mem[i] = 0;
```

} 이러한 반복문은 방이 많은 배열을 한번에 초기화 시킬 수 있도록 하며 이는 복잡한 과정을 단순화하고 직관적으로 만든다.

```
char Ram::read(int address) {
```

```
    return mem[address];
```

```
}
```

```
void Ram::write(int address, char value) {
```

```
mem[address] = value;
```

} 메모리는 배열 형태로 구현되어 있기 때문에, address를 통해 특정 위치에 직접 접근할 수 있다. 이는 원하는 메모리 위치에 접근할 수 있기에, 다양한 위치에서 데이터를 read & write할 수 있다.