3주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231515 이름: 김다은

이 문제를 효율적으로 해결하기 위해 **OOP(객체지향프로그래밍)**를 사용할 수 있다. 구현하고자 하는 목표를 세분화하여 객체라는 단위로 나눈 후, 이들을 상속 등의 방법으로 조합하여 프로그램을 설계하는 방법이 바로 OOP이다. OOP는 재사용 최대화, 캡슐화, 추상화, 상속 등의 장점을 가진다.

Array는 1차원 배열을 동적 할당하고 관리하는 역할을 하는 class이다. Array의 생성자에서는 `new int[size]`를 이용해 배열을 동적 할당한다. Array의 소멸자에서는 ‘delete[] data`를 이용해 배열에 할당된 메모리를 해제한다.

RangeArray는 Array를 **상속**하여 정의한다. 문제를 효율적으로 해결하기 위해서는 이미 선언된Array를 최대한 이용하는 것이 중요하다. 여기에서 상속이란 child class가 parents class로부터 변수나 함수를 물려받아 사용할 수 있도록 하는 것을 말한다. 즉, 기초가 되는 parents class를 정의하고 parents class의 일부를 변형하고 새로운 내용을 추가하여 child class를 정의하여 사용할 수 있다.

**Array는 인덱스를 0~len-1로 가진다. 하지만 RangeArray는 인덱스를 입력된 base~end**로 가져야 한다. 따라서 RangeArray에서 Array의 배열에 접근하는 변수를 다시 정의하여, Array의 인덱스에 정상적으로 접근할 수 있도록 해야 한다. 따라서 RangeArray에서 Array에 접근할 때 ‘(인덱스) – base’를 해주어 정상적인 접근이 이루어지도록 하는 것이 핵심이다.

문제 해결을 위한 단계별 수행 내용은 아래와 같다.

1. RangeArray 생성자가 호출됨 ex) RangeArray RA(-1,3)

RangeArray::RangeArray(int \_base, int \_end): base(\_base), end(\_end), Array(\_base-\_end) {};

입력된 인덱스 범위를 각각 base와 end에 저장한다. 또한 Array(base-end+1)를 통해 base-end+1의 크기를 가지는 동적 할당된 배열을 생성한다.

2. RangeArray 변수에 index 접근 ex) RA[2]

int& RangeArray::operator [](int i)

{

    return Array::operator [](i-base);

}

int RangeArray::operator [](int i) const

{

    return Array::operator [](i-base);

}

(입력된 index 값) – base의 인덱스로 변환한다. 이후 Array에 접근한다.