

컴퓨터공학실험1 3주차 결과보고서

전공: 컴퓨터공학과

학년: 2

학번: 20191559

이름: 강상원

문제 1. 주어진 문제 및 5-1의 문제 해결에 관한 내용을 이해하고 이 문제를 효율적으로 해결하기 위한 방법을 생각하여 이를 1쪽 이내로 요약하여 제출하시오. 문제 해결을 위한 간단한 단계별 수행 내용, 자료구조 등을 기술하시오. (RangeArray)

5-1의 문제는 일반적인 0-indexed 배열과는 다른 형식의 배열을 만드는 문제이다. 일례로 RangeArray(-5, 5)를 선언하면 -5부터 5까지 11개의 원소 위치를 갖게 된다.

문제를 구현하기 위해서는 다중정의가 필요하다. 함수 다중정의는 C++에서, 함수의 이름이 같아도 자료형이나 인자 개수가 다른 함수들을 알아서 구분하는 성질을 의미한다.

예를 들어, `int sum(int a, int b);` `int sum(float a, float b, float c);` 함수에서 입력값이 `int` 변수 2개이면 앞의 함수, `float` 3개이면 후자의 함수를 택하는 것을 말한다.

RangeArray 클래스는 Array 클래스를 상속한다. RangeArray는 `[low, high]` 범위여서 `[0, n-1]`의 Array에서는 `n=high-low+1`, `RangeArray[i]=Array[i-low]`이다.

➔ 다중정의를 이용해 찾고자 하는 index가 음수인 경우, 양수인 경우를 나누어 함수를 정의한다.

문제 2. 실험 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오. (프로그램 구조도 포함)

RangeArray 클래스는 Array 클래스를 상속한다. RangeArray는 `[low, high]` 범위여서 `[0, n-1]`의 Array에서는 `n=high-low+1`, `RangeArray[i]=Array[i-low]`이다.

- Class Array

- protected

- `int *data` : 메모리를 동적으로 할당하여 자료 저장
 - `int len` : 배열의 길이 저장

- public

- `Array(int size)` : 크기 size인 배열 생성
 - `~Array()` : 소멸자 : 배열을 메모리에서 제거
 - `int length()` : len의 값 반환

- int& operator[](int i) : 배열에 값 할당 (i번째 위치)
- int operator[](int i) : 배열 값 읽어들이기 (i번째 위치)
- void print() : 배열 원소 출력

● Class RangeArray

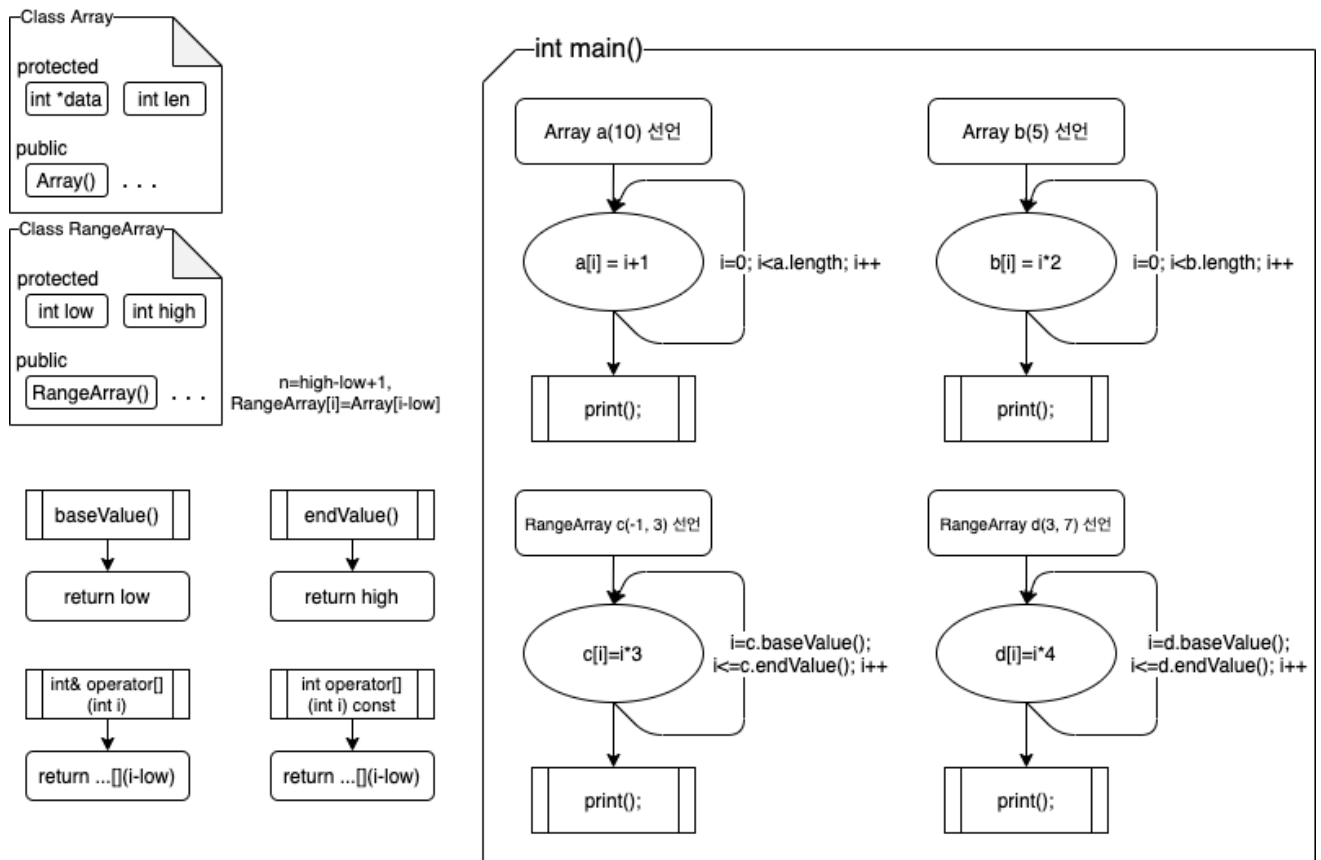
➤ protected

- int low : 배열 시작 인덱스
- int high : 배열 끝 인덱스

➤ public

- RangeArray(int low, int high) : high-low+1 크기의 배열 생성
- ~RangeArray() : 소멸자 : 배열을 메모리에서 제거
- int baseValue() : 배열의 시작 인덱스
- int endValue() : 배열의 끝 인덱스
- int& operator[](int i) : 배열에 값 할당 (i번째 위치)
- int operator[](int i) const : 배열 값 읽어들이기 (i번째 위치)

프로그램 구조도



문제 3. 숙제 문제를 해결하기 위한 알고리즘 및 자료구조를 요약하여 기술하십시오. (3주차 과제)

문자열의 길이를 `char*` 형으로 저장한다. 문자열 값을 입력하기 위한 등호 (=) 연산자를 새롭게 정의한다. `strcmp` 함수를 이용한 문자열 비교 함수도 구현한다.

- **Class Str**

- **private**

- `char *str`: 문자열 내용
 - `int len`: 문자열 길이

- **public**

- `Str(int leng)`: `leng+1`만큼 `char`을 동적 할당하여 `str` 만들기, `len` 값 부여
 - `Str(char *neyong)`: `len`만큼 `char`을 동적 할당하여 `str` 만듦-> `strcpy` 이용해 `str`에 `neyong`의 내용을 넣는다.
 - `~Str()`: 소멸자 : 문자열을 메모리에서 제거
 - `int length()`: `private`로 보호되어 있는 문자열의 `length` 반환
 - `char *contents()`: 문자열 내용 반환
 - `int compare(class Str &a)`: 두 문자열 비교 (`strcmp(str, a)` 이용)
 - `int compare(char *a)`: 두 문자열 비교 (`strcmp(str, a.contents())` 이용)
 - `void operator=(char *a)`: `str` 메모리 해제, `len+1`만큼의 문자열 동적 할당, `a`의 내용 복사
 - `void operator=(class Str &a)`: `str` 메모리 해제, `len` 값 넣기, `len+1`만큼의 문자열 동적 할당, `a.contents()`의 내용 복사

3주차 강의 PPT의 실습 결과 레포트에 적혀있는 OOP, 객체, 클래스 등은 예비보고서에 작성했으므로 결과보고서에는 작성하지 않습니다.