3주차 결과보고서

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

**1. 실험 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오.**

먼저 만들어진 1차원 배열을 관리하는 class Array를 상속하여 class RangeArray 자료구조를 만드는 것이 목적이었다.

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

Array는 인덱스 범위를 0~len-1로 가진다. 그러나 RangeArray는 인덱스를 입력된 high~low로 가지는 class이다. 그러므로 RangeArray에서 Array의 배열에 접근하는 인덱스 변수를 다시 정의하여, Array의 인덱스에 정상적으로 접근할 수 있도록 해야 한다. 따라서 RangeArray에서 Array에 접근할 때 ‘(인덱스 값) – low’를 해주어 정상적인 접근이 이루어지도록 하는 게 중요했다.

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

{

    return Array::operator [](i-low);

*// (인덱스 값) – low를 해주어 정상적으로 접근하도록 함*

}

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

{

    return Array::operator [](i-low);

*// (인덱스 값) – low를 해주어 정상적으로 접근하도록 함*

}

**2. 숙제 문제를 해결하기 위한 알고리즘 및 자료구조를 요약하여 기술하시오.**

숙제는 class Str의 메서드를 정의하여 문자열을 저장하고 처리하는 자료구조를 만드는 것이었다. Str은 문자열의 내용을 저장하는 char \*str과 문자열의 길이를 저장하는 int len을 private 변수로 가진다. 이때 str에 new를 이용해 메모리를 동적 할당하여 사용한다.

이미 동적 할당된 메모리의 크기를 수정하는 것은 불가능하다. 따라서, 새로운 문자열을 저장하거나 길이가 다른 문자열을 저장해야 하는 경우 기존 메모리를 delete를 이용해 할당 해제하고, new를 이용해 새로운 크기로 재할당해야 한다는 점이 핵심이었다.

void Str::operator=(char \*a) { *// string의 값을 대입.*

*// a 문자열의 길이 저장*

    len = strlen(a);

*// str에 동적 할당된 메모리 해제*

    delete[] str;

*// a의 길이(len)에 맞추어 메모리 동적 할당*

    str = new char[len+1];

*// str에 a의 내용 복사*

    strcpy(str, a);

}

또한 다른 Str a가 인수로 주어지는 함수가 있다. 이때 다른 Str인 a의 private 변수에 직접 접근할 수 없으므로, a.content()나 a.length()처럼 public 함수를 통해 a의 len과 str을 받아서 처리해야 하는 점이 두 번째 핵심이었다.

void Str::operator=(class Str& a) { *// Str의 내용을 대입.*

*// Str a의 len을 저장*

    len = a.length();

*// str에 동적 할당된 메모리 해제*

    delete[] str;

*// len(a.len)에 맞추어 메모리 동적 할당*

    str = new char[len+1];

*// str에 a.str의 내용 복사*

    strcpy(str, a.contents());

}