3주차 결과보고서

전공: 수학/컴퓨터공학 학년: 3학년 학번: 20181294 이름: 임승섭

**1. 실습 파일**

1-1. Array.h

Array라는 class를 생성한다.

protected(상속 관계에서 접근이 가능)

: int형 데이터를 원소로 갖는 \*data 배열과

그 배열의 길이를 나타내는 정수형 변수 len이 있다.

public(어디서든 접근이 가능)

: size의 크기만큼 배열의 길이를 할당하여 배열을 생성하는 Array(int size). 그 길이를 len에 저장한다.

이렇게 할당한 배열은 소멸자 ~Array()를 이용하여 삭제한다.

배열의 길이를 리턴하는 함수 length().

연산자 다중정의를 이용하여 배열의 i번째 인덱스에 값을 삽입하는 int& operator[](int i)

연산자 다중정의를 이용하여 배열의 i번째 인덱스의 값을 리턴하는 int operator[](int i)

반복문을 이용해서 배열의 원소를 모두 출력하는 print()

1-2. array.cpp

위에 설명했던 “Array.h” 파일을 include 한다.

Array::Array(int size) : size가 음수이면 에러 메시지를 출력하고, size가 양수이면 그 크기만큼의 배열을 생성하고, len에는 size의 값을 넣어준다.

Array::~Array() : 소멸자이다. data 배열에 할당되었던 메모리를 해제한다.

int Array::length() const : 배열의 크기로 저장해두었던 len을 반환한다.

int& Array::operator[](int i) : int형 변수 tmp를 선언한다. i가 인덱스의 범위 내에 있으면 배열의 i번째 인덱스 값을 반환하고, 인덱스의 범위 밖에 있을 때에는 에러메세지를 출력하고 tmp를 반환한다.

int Array::operator[](int i) const : i가 인덱스의 범위 내에 있으면 배열의 i번째 값을 반환하고, 인덱스의 범위 밖에 있을 때에는 에러메세지를 출력하고 0을 반환한다.

void Array::print() : int형 변수 i를 선언한다. “[”를 출력한다. i를 사용한 반복문을 활용해 “ ”와 배열의 인덱스 값을 모두 출력한다. 마지막에는 “]”를 출력하여 마무리한다.

1-3. RangeArray.h

“Array.h”를 include 한다.

RangeArray라는 class를 생성한다. 이 때 Array class 의 속성 접근 지정자의 의미를 그대로 상속한다.

protected (상속 관계에서 접근 가능)

: 범위의 첫번째 인덱스를 저장하는 변수 low와 마지막 인덱스를 저장하는 변수 high를 선언한다.

public (어디서든 접근 가능)

: low 값과 high값을 받아서 배열을 할당하는 RangeArray(int, int)

할당했던 배열을 삭제하는 소멸자 ~RangeArray()

배열의 첫번째 인덱스의 값을 반환하는 int baseValue()

배열의 마지막 인덱스의 값을 반환하는 int endValue()

연산자 다중정의를 이용하여 i번째 인덱스에 값을 삽입하는 int& operator[](int)

연산자 다중정의를 이용하여 i번째 인덱스의 값을 반환하는 int operator[](int) const

1-4. rangearray.cpp

위에 설명했던 “RangeArray.h” 파일을 include한다.

RangeArray::RangeArray(int i, int j):Array(j-i+1) : RangeArray는 인자로 첫번째와 마지막 인덱스 번호를 받지만, Array는 배열의 크기를 받는다. 그래서 low부터 high까지 정수 개수를 j-i+1으로 표현하여 Array에 대입한다. 이를 통해 배열을 생성한다.

RangeArray::~RangeArray() : 소멸자. 이미 Array클래스에 소멸자가 정의되어 있어서 빈 공간으로 둔다.

int RangeArray::baseValue() : low, 즉 첫 번째 인덱스 값을 반환한다.

int RangeArray::endValue() : high, 즉 마지막 인덱스 값을 반환한다.

int RangeArray::operator [](int i) : Array class의 함수를 상속받는다. 마찬가지로 인자를 맞추기 위해 i-low를 대입한다.

int RangeArray::operaotr [](int i) const : Array class의 함수를 상속받는다. 마찬가지로 인자를 맞추기 위해 i-low를 대입한다.

1-5. main.cpp

“Array.h”, “RangeArray.h”를 include한다.

정수형 변수 i, x, y를 선언한다. x, y는 배열의 값을 받고, i는 반복문에서 사용할 예정이다.

Araay를 이용하여 a는 20만큼, b는 10만큼의 크기를 할당하여 배열을 생성한다.

i = 0부터 a의 길이(20) 만큼 for loop 을 돌면서 i번째 인덱스에 i+1값을 대입한다.

i = 0부터 b의 길이(10) 만큼 for loop 을 돌면서 i번째 인덱스에 i\*2값을 대입한다.

“a(10)”을 출력하고 그 뒤에 print함수를 이용해서 a의 모든 인덱스 값을 출력한다.

“b(5)”를 출력하고 그 뒤에 print함수를 이용해서 b의 모든 인덱스 값을 출력한다.

“a[-1]”을 출력하고 a[-1]에 7을 대입한다. 하지만 인덱스 범위에 맞지 않기 때문에 에러 메시지를 출력한다.

x에 a[0], y에 b[0] 값을 대입한다.

“a[0] = ”을 출력하고 뒤에 x 출력, “b[0] = ”을 출력하고 뒤에 y를 출력한다.

RangeArray를 이용하여 -1부터 5까지 인덱스를 갖는 배열 c, 2부터 8까지의 인덱스를 갖는 배열 d를 생성한다.

c의 첫 번째 인덱스부터 마지막 인덱스까지 i로 for loop을 돌면서 i번째 인덱스에 i\*3을 대입한다.

d의 첫 번째 인덱스부터 마지막 인덱스까지 i로 for loop을 돌면서 i번째 인덱스에 i\*4를 대입한다.

“c(-1, 3) ”을 출력하고, 뒤에 print를 이용해서 c의 모든 인덱스 값을 출력한다.

“d(3, 7) ”을 출력하고, 뒤에 print를 이용해서 d의 모든 인덱스의 값을 출력한다.

“c[-2] ”을 출력하고, c[-2]에 3을 대입한다. 인덱스 범위에 맞지 않기 때문에 에러 메시지를 출력한다.

x에 c[-1], y에 d[4] 값을 대입한다.

“c[-1] = ” 출력, x 출력, “ d[3] = ” 출력, y 출력.

**2. 과제 파일**

2-1. Str.h

Str class를 생성한다.

private(해당 클래스에서만 접근 가능)

string의 내용을 저장하는 str과 string의 길이를 저장하는 len이 있다.

public(어디서나 접근 가능)

string의 길이 leng을 받아 문자열을 생성하는 생성자 Str(int leng)

string의 내용 neyong을 받아 문자열을 생성하는 생성자 Str(char \*neyong)

만든 문자열을 삭제하는 소멸자 ~Str()

string의 길이(len)을 리턴하는 int length(void)

string의 내용(str)을 리턴하는 char \*contents(void)

a의 내용과 string의 내용을 비교하는 int compare(class Str&a), int compare(char \*a)

string에 원하는 값을 대입하는 void operator=(char \*a), void operator=(class Str& a);

2-2. Str.cpp

위에 설명한 “Str.h”를 include한다.

Str(int leng) : leng이 양수일 때, len에 문자열의 길이를 저장하고 str에는 새로운 배열을 할당한다. 이 때 마지막 Null문자의 공간을 위해 len+1만큼의 크기를 할당한다. 현재 내용이 없기 때문에 0번째 인덱스를 Null문자로 해준다. leng이 음수일 때는 에러 메시지를 출력한다.

Str(char \*neyong) : len에는 neyong의 길이를 strlen을 이용해서 받아온 후 대입한다. str에는 위와 마찬가지로 neyong의 길이 + 1 만큼의 크기를 할당한 문자열 배열을 선언하고, strcpy를 이용하여 neyong의 값을 str에 복사한다.

~Str : 소멸자. delete를 이용하여 배열을 삭제한다.

length(void) : 문자열의 len 값을 반환한다. (문자열의 길이)

contents(void) : 문자열의 str 값을 반환한다. (문자열의 내용)

compare(char \*a) : 문자열의 str 값과 a를 strcmp한 값을 반환한다. 두 문자열이 같으면 0을 반환한다.

compare(class Str& a) : 문자열의 str 값과 a.contents()를 strcmp한 값을 반환한다. 이 때 a는 class 형태로 들어오기 때문에 contents() 함수를 이용해 내용을 받은 후 strcmp에 대입한다.

operator=(char \*a) : 우선 delete를 이용하여 문자열을 삭제하고 len에는 a의 길이, str에는 len+1만큼의 배열을 할당한다. strcpy를 이용하여 문자열의 내용에 a를 복사한다.

operator=(class Str& a) : 우선 delete를 이용하여 문자열을 삭제하고 len에는 a의 길이, str에는 len+1만큼의 배열을 할당한다. 이 때 a는 class 형태로 들어오기 때문에 length() 함수를 이용해서 길이를 받는다. strcpy를 이용하여 문자열의 내용에 a의 내용을 복사한다.