

실험 CPP-1: RangeArray

1. 목적

C++는 객체지향 프로그래밍 개념을 도입하여 클래스(Class)를 확장하고 수정하는 수단으로 클래스 상속(Class Inheritance)을 제공한다. 이번 실험에서는 효율적인 프로그래밍을 위하여 클래스의 상속 방법을 익혀보도록 한다.

2. 문제 설명

아래 부분에 추상화된 기초 클래스 Array는 new 연산자를 사용하여 정수 타입의 1차원 배열을 동적으로 생성한다. 또한 배열의 인덱스를 나타내는 '['에 대하여 연산자 다중정의의를 하여 Array클래스로 생성된 객체에서도 사용할 수 있도록 한다. 우리는 Array클래스를 상속받아 RangeArray를 추가로 구현하려 한다. RangeArray클래스의 특징은 배열이 인덱스 0에서 시작되는 것이 아니라, 생성자에 전달된 임의의 범위를 갖도록 한다. 한 예로 RangeArray A가 A(-10,10)로 생성되었다면, 인덱스 (-10 ~ 10)를 가지고, 그 배열은 21개의 원소를 저장할 수 있다. 즉 A[-10], A[-9], ..., A[0], A[1], ..., A[10]은 모두 올바른 배열 표현이다. 객체 지향 프로그래밍에 적합하도록 Array클래스로부터 가능한 많은 기능을 상속받아 RangeArray클래스를 구현한다.

```
class Array {
protected:
    int *data;    // 배열 포인터
    int len;      // 배열의 크기
public:
    Array(int size) { // 생성자
        // size가 0 이하 값이면 에러메시지 출력하고 종료
        // size가 양수이면 new를 사용하여 배열 data를 할당, len값 초기화
    }
    ~Array() { ... } //소멸자; 할당된 메모리 해제
    int length() const { ... } // 배열의 크기 리턴
    // 배열에 원소를 대입하거나 값을 반환하는 부분으로 []연산자의 다중정의이다.
    int & operator [](int i) { // left value : 배열에 원소를 삽입
        static int tmp;
        // 배열의 인덱스가 범위 내에 있으면 해당 원소를 리턴
        // 그렇지 않으면 에러 메시지 출력하고 tmp를 리턴
    }
    int operator [](int i) const { // right value : 배열의 원소의 값을 반환
        // 배열의 인덱스가 범위 내에 있으면 값을 리턴
        // 그렇지 않으면 에러 메시지 출력하고 0을 리턴
    }
    void print() { ... } // 배열의 모든 내용을 출력해주는 함수
};
```

[그림 1] Array클래스의 추상 데이터 타입

3. 입력

RangeArray클래스를 구현한 후에, 이것을 테스트해보기 위하여 아래의 main()함수를 수행하여 본다.

```
int main() {
    int i, x, y;
    Array a(10), b(5);
    for(i=0; i<a.length(); i++) a[i]= i+1;    //left value
    for(i=0; i<b.length(); i++) b[i]= i*2;

    cout<<"a(10)": a.print();
    cout<<"b(5)": b.print();
    cout<<"a[-1] ";
    a[-1]=7; // Array bound error!
    x=a[0]; y=b[0]; // right value
    cout<<"a[0]="<<x<<" b[0]="<<y<<endl;
    RangeArray c(-1, 3), d(3, 7);
    for(i=c.baseValue(); i<=c.endValue(); i++) c[i]=i*3; //left value
    for(i=d.baseValue(); i<=d.endValue(); i++) d[i]=i*4;
    cout<<"c(-1,3)": c.print();
    cout<<"d(3,7)": d.print();
    cout<<"c[-2] ";
    c[-2]=3; // Array bound error!
    x=c[-1]; y=d[3]; // right value
    cout<<"c[-1]="<<x<<" d[3]="<<y<<endl;
    return 0;
}
```

[그림 2] main() 함수

4. 출력 (위 main()함수를 수행하면 아래와 같은 결과가 출력된다.)

```
a(10)[ 1 2 3 4 5 6 7 8 9 10]
b(5)[ 0 2 4 6 8]
a[-1] Array bound error!
a[0]=1 b[0]=0
c(-1,3)[ -3 0 3 6 9]
d(3,7)[ 12 16 20 24 28]
c[-2] Array bound error!
c[-1]=-3 d[3]=12
```

5. 실험 방법

5-1. 문제 해결

기본 클래스인 Array 클래스를 먼저 구현하여야 한다. Array클래스의 생성자는 사용자가 입력한 크기의 정수 타입의 배열을 new 연산자를 통하여 할당받는다. 마찬가지로 소멸자에서 delete 연산자를 통하여 할당된 메모리를 해제한다. 일반적인 배열에서는 '['에 인

텍스를 넣어 해당 원소를 참조하거나 값을 대입한다. Array클래스에서도 같은 용도로 '[]'을 사용하기 위하여 연산자 다중정의를 이용한다. 배열의 원소값을 참조하는 right value용도의 `int operator []`와 배열의 원소에 새로운 값을 대입할 수 있는 left value용도의 `int &operator []`를 각각 구현하도록 한다. 각 함수에서 잘못된 인덱스를 가질 때에는 에러 메시지를 출력한다. 또한 배열의 모든 원소에 대하여 출력하는 `print()`멤버함수를 구현한다.

RangeArray 클래스는 인덱스의 범위가 0에서 시작하는 것이 아니라 생성시 동적으로 정해지므로 클래스 내부에 배열의 인덱스 범위를 나타내는 base와 end변수를 선언하여야한다. RangeArray 클래스의 생성자는 입력받은 범위로부터 배열크기를 계산하여 부모 클래스인 Array클래스의 생성자를 통하여 메모리를 할당받는다. `int operator []`, `int &operator []` 멤버함수는 배열의 범위에 속하는 원소에 대하여 수행되어야 한다. 사용자가 RangeArray클래스의 배열 범위를 알 수 있도록 base값과 end값을 리턴해주는 `baseValue()`, `endValue()` 멤버함수를 추가한다.

5-2. 프로그램 작성

본 실험은 Unix 또는 Linux환경에서 g++을 사용한다. 생각한 해결방법을 다음과 같은 단계를 거쳐 프로그래밍을 수행한다.

1. 문제 해결에 필요한 클래스의 데이터 멤버와 멤버함수를 설계한다.
2. 간단한 예를 만들어 이에 자신이 설계한 클래스에 적용하여 실행 및 검토한다.
3. 잘못된 입력에 대해 적절한 에러 메시지를 출력해준다.
4. 문제의 주어진 예 외에도 여러 입력을 적용하여 자신의 프로그램을 수행시켜 보고 필요한 경우 디버깅 과정을 통하여 수행과정을 살펴보거나 발견된 오류를 정정한다.

5-3. 프로그램 작성 확인

담당 강사 또는 조교에게 작성한 프로그램을 확인 받는다.

6. 숙제 및 보고서 작성

6-1. 예비 보고서

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.

작성한 예비 보고서는 실험 시작 전에 제출하여야한다.

6-2. 숙제

실험 당일 강사가 지정한 문제를 해결하는 프로그램을 해결하는 프로그램을 작성한다.

6-3. 결과 보고서

아래에 보인 사항을 작성하여 다음 실험 시간에 제출하시오.

1. 실험 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오.
2. 숙제 문제를 해결하기 위한 알고리즘 및 자료구조를 요약하여 기술하시오.
3. 작성한 프로그램을 공지한 제출요령에 의거하여 기한 내에 제출하시오, 요청이 있을 경우 실험 시간에 작성한 프로그램도 아울러 제출한다.

실험 CPP-1: 예비 보고서

전공:

학년:

학번:

이름

실험 CPP-1: 결과 보고서

전공:

학년:

학번:

이름

