**실험 CPP-1: 결과 보고서**

전공: 수학과 학년: 4 학번: 20161255 이름: 장원태

1. 실험 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오.

이번 실험 시간에 작성한 프로그램의 큰 틀은, Array 클래스를 구축한 뒤 그를 상속하는 RangeArray 클래스를 제작하는 것이다. 우선, Array 클래스를 구성하는 요소는 다음과 같다.

Protected: int형 배열인 data, 그 배열의 길이를 담고 있는 int형 변수인 len

Public: 아래의 6개이다.

Array(int size) – 생성자로서, 크기가 size인 int형 배열을 선언해준다. 또한, 그 배열의 크기를 len 변수에 저장한다.

~Array() – 소멸자로서, Array 클래스의 멤버 변수인 int형 배열인 data를 삭제한다.

Length() – 배열의 길이, 즉 len 변수의 값을 리턴하는 메소드이다.

Int& operator[](int i) – 배열의 원소에 새로운 값을 대입할 수 있는 left value용도의 메소드로, 주소값을 반환하여 해당 주소에 원소를 삽입한다. 단, 인덱스를 초과하면 error를 출력하고 메소드 내의 변수인 tmp를 리턴한다.

Int operator[](int i) – 배열의 원소값을 참조하는 right value용도의 메소드로, 연산자 []를 다중정의하며, int형 배열인 data의 i번째 인덱스 값, 즉 data[i]를 리턴한다. 단, 인덱스를 초과하면 error를 출력하고 0을 리턴한다.

Print() – 배열의 모든 원소를 출력하는 메소드이다. 여는 대괄호로 시작하여 data 배열의 모든 원소를 for문을 이용하여 출력한 뒤, 닫는 대괄호로 마무리한다.

이 클래스를 기반 클래스로 하여, 그에 대한 파생 클래스로 정의되는 것이 바로 RangeArray 클래스, 즉 우리가 궁극적으로 만들고자 하는 클래스이다. 대부분의 알고리즘은 Array 클래스를 따라가되, 인덱스의 경우 Array 클래스에서 0~(n-1) 이었던 것이 RangeArray 클래스에서는 low~high가 되므로, Array 클래스에 인자로 넣을 때 이를 조절하여 넣어야 한다. Array 클래스의 인자는 만드는 배열의 크기를 받으므로, RangeArray 클래스에서 정의하는 배열의 크기, 즉 low 이상 high 이하의 정수의 개수인 (high-low+1)을 Array 클래스에 인자로서 대입해야 한다.

이러한 알고리즘을 바탕으로 구성하는 RangeArray 클래스는 다음과 같이 구성한다.

Protected: 배열의 시작 인덱스를 의미하는 int형 변수 low와, 끝 인덱스를 의미하는 int형 변수 high

Public: 아래의 6개이다.

RangeArray(int, int) – 생성자로서, 시작 인덱스와 끝 인덱스를 인자로 가진다. Array 클래스로는 해당 배열의 크기인 j-i+1 을 대입하게 된다.

~RangeArray() – 소멸자로서, 생성된 배열을 삭제해야 하지만, Array 클래스를 상속하는 과정에서 이미 소멸자가 정의되어 있으므로 다시 정의하지 않고 빈 메소드로 남긴다.

baseValue() – 배열의 시작 인덱스 값을 리턴하는 메소드이다.

endValue() – 배열의 끝 인덱스 값을 리턴하는 메소드이다.

Int& operator[](int i) – 기반 클래스인 Array 클래스의 int &operator[](int i) 클래스를 상속받는 메소드이다. 이 때, RangeArray 클래스의 인덱스 범위는 low ~ high이므로, 0~n인 Array 클래스의 인덱스 범위에 맞추기 위해서는 Array 클래스에 (i-low)를 인자로 넘기면

된다.

Int operator[](int i) – 기반 클래스인 Array 클래스의 int operator[](int i) 클래스를 상속받는 메소드이다. Int& operator[](int i) 메소드와 마찬가지로, Array 클래스에 (i-low)를 인자로 넘긴다.

이들을 바탕으로 main 함수의 출력 결과를 보면, 우선 초반에는 Array 클래스를 통해 배열을 정의하고 있으며, a.print()를 실행하면 a 배열의 모든 원소가, b.print()를 실행하면 b 배열의 모든 원소가 출력된다. 그 후에, a[0]과 b[0]을 출력하면 정상적으로 해당 인덱스 값이 출력이 된다. 단, a[-1]를 출력하려 하면 인덱스가 범위를 벗어나므로 error가 출력된다.

후반에는 RangeArray 클래스를 통해 배열을 정의한다. 즉, 인덱스가 -1~5인 c 배열과, 인덱스가 2~8인 d 배열을 정의했다. 이 경우도, c.print()와 d.print()를 실행하면 각각 시작 인덱스 값부터 끝 인덱스 값까지 정상적으로 출력이 된다. 또한, c[-1]과 d[3]을 출력하면 각 인덱스가 각 배열의 범위 내에 있으므로 정상적으로 값이 출력된다. 단, c[-2]를 출력하려 하면 인덱스가 -1~5의 범위를 벗어나므로 error가 출력된다.

1. 숙제 문제를 해결하기 위한 알고리즘 및 자료구조를 요약하여 기술하시오.

이번 숙제 문제는 일반적인 스트링형 변수를 Str 클래스로 정의해보는 문제로, 스트링의 길이를 받거나 문자열을 받아 문자열 객체를 생성한다. Str 클래스를 구성하는 요소들은 다음과 같다.

Private: 문자열의 내용을 담는 str 변수와, 그 문자열의 길이를 받는 int형 변수인 len

Public: 다음 9가지이다.

Str(int leng) – 생성자로, 크기가 leng인 문자열을 생성한다.

Str(char \*neyong) – 생성자로, 내용이 neyong인 문자열을 생성한다. 이 때, 변수 len에 그 문자열의 길이를 저장한다.

~Str() – 소멸자로, 해당 문자열을 삭제한다.

int length(void) – 문자열의 길이를 리턴하는 메소드이다.

Char \*contents(void) – 문자열의 내용을 반환하는 메소드이다.

Int compare(class Str& a) – str 문자열과 a 문자열의 내용이 같은지를 비교하는 메소드로, 그 두 문자열의 strcmp 값을 리턴한다.

Int compare(char \*a) – str 문자열과 a 문자열의 내용이 같은지를 비교하는 메소드로, 그 두 문자열의 strcmp 값을 리턴한다. 이 경우, a의 포인터 자체를 가리키므로 a와 바로 비교할 수 있다.

Void operator=(char \*a) – “=” 연산자를 다중정의하는 메소드로, str 문자열에 a 문자열을 복사하여 덮어씌운다. 단, len 변수에도 그 문자열의 길이를 다시 저장해주어야 한다.

Void operator=(class Str& a) – “=” 연산자를 다중정의하는 메소드로, str 문자열에 a 문자열을 복사하여 덮어씌운다. 단, len 변수에도 그 문자열의 길이를 다시 저장해주어야 한다.

위와 같은 클래스를 이용하여, main 함수의 경우 우선 a 배열에 “I’m a girl”이라는 문자열을 저장했다. 그 후, a.contents를 출력하면 “I’m a girl”이 출력된다. 그 후, a=”I’m a boy\n”과 같이, 다중정의된 “=” 연산자를 이용하여 a 배열을 재정의한 뒤 다시 a.contents를 출력하면, “I’m a boy”가 출력되고 줄바꿈까지 일어난다. 그 후, a.compare(“I’m a a”)를 출력하게 되면, 현재 a 배열에 있는 “I’m a boy\n”과 “I’m a a”를 비교하여 strcmp 값이 출력되게 된다. 이 경우, “I’m a boy\n”에서 ‘b’의 위치부터 달라지며, 비교한 대상인 ‘a’보다 아스키코드 값이 더 크므로 1을 반환하게 된다.

1. OOP에 대해 서술하시오.

OOP란 객체 지향 프로그래밍(Object-Oriented Programming)의 약자로, 컴퓨터 프로그래밍에서 필요한 데이터를 추상화시킴을 통해 객체(objects)를 만들어 사용하는 프로그래밍 패러다임이다. 이 때 객체는 데이터 필드와 메소드 및 그 객체들간의 상호작용을 통해 구성된다. 근본적인 특성으로는 동적 결합, 캡슐화, 서브타입 다형성, 상속 등이 있다. 알고리즘을 우선적으로 고려하고 그 뒤에 자료구조를 고려하는 전통적인 절차적 프로그래밍과는 다르게, OOP는 자료구조를 우선적으로 고려하고 그 뒤에 알고리즘을 고려한다.

1. OOP를 쓰는 이유에 대해 서술하시오.

OOP를 사용하면 우선, 재사용성이 최대화된다. 즉, 캡슐화와 상속 등을 통해 기존 코드들을 재사용할 수 있다는 장점이 있다. 또한, 비슷한 종류의 데이터나 속성들을 모아서 묶어둘 수 있기 때문에, 개발 시간 및 디버깅 시간이 단축된다. 이러한 성질 덕분에 코드를 재사용하는 편리함도 생긴다.

1. 객체에 대해 서술하시오.

객체는 함수 같은 느낌으로, OOP에서 클래스의 특정한 인스턴스이다. 즉, OOP에서, 필요한 데이터를 추상화시킴을 통해 만들어진다. 객체는 상태(state)와 행동(behavior)으로 구성된다. 상태란 객체가 가지는 속성 및 특성, 즉 데이터와 속성을 의미한다. 행동이란 객체가 가지는 행동이나 할 수 있는 반응 양식, 즉 메소드와 오퍼레이션을 의미한다.

1. 클래스에 대해 서술하시오.

클래스란 OOP에서 자신의 인스턴스를 만들기 위하여 형틀(청사진)로 사용되는 구조물로, 같은 종류의 집단에 속하는 속성과 행동을 정의한 것이다. 이는 곧 객체를 생성하기 위한 일종의 형틀이라고 생각하면 된다.

1. 인스턴스에 대해 서술하시오.

인스턴스는 동일한 클래스에서 다수의 객체들이 생성되며, 그 과정에서 생성된다. 하나의 클래스에서 생성된 인스턴스는 해당 클래스의 모든 메소드를 공유하게 된다.

1. 상속의 개념과 사용 이유에 대해 서술하시오.

상속이란, 클래스가 인스턴스를 생성할 때, 다른 클래스의 속성을 해당 클래스의 속성과 합친 후, 하나의 인스턴스를 생성하는 것을 의미한다. 상속이 이루어지면, 기반 클래스와 파생 클래스의 분류가 생긴다. 기반 클래스에서 상속되어 파생 클래스가 만들어지는 것이다. 이런 식으로 만들 때, 하나의 기반 클래스에 대한 파생 클래스를 여러 개 만들게 되면, 파생 클래스의 공통된 부분에 대해 분석, 설계, 구현, 테스트, 디버그 및 유지 관리할 필요가 없어진다. 또한, 클래스를 아무리 파생시켜도 기반 클래스에는 영향이 가지 않는다는 편리함도 있다. 이러한 장점을 통해 코드의 재사용성이라는 편리한 장점을 활용하기 위해 상속을 사용한다.