전공 : 컴퓨터공학과 학년 : 4 학번 : 20212020 이름 : 박민준

1. 실험 시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오.

이번 실습은 c++의 객체지향 프로그래밍 개념 중에서 클래스 상속을 활용하여 배열을 추상화하고 확장하는 방식으로 프로그램을 작성하였다.

[알고리즘]

먼저 가장 기본이 되는 Array 클래스를 구현한다. Array 클래스의 생성자에서는 new 연산자를 사용하여 사용자가 지정한 크기의 정수 배열을 동적으로 할당한다. 이렇게 동적 할당된 메모리는 클래스의 data 포인터 변수에 저장된다. 동적 메모리 할당을 통해 정수 배열을 생성한 뒤, 연산자 다중정의(Overloading, ‘[ ]’)를 통해 배열 요소에 접근하거나 수정할 수 있다. Overloading된 연산자는 배열의 인덱스가 유효한 범위 내에 있는지 확인하고, 그렇지 않을 경우 에러 메시지를 출력한다. 또한, Array 클래스에는 배열의 크기를 반환하는 length() 함수와 배열의 내용을 출력하는 print() 함수를 포함한다. 마지막으로 ~Array() 소멸자에서는 delete 연산자를 사용하여 할당된 메모리를 해제함으로써 메모리 누수를 방지한다.

Array 클래스를 구현한 뒤, 해당 클래스의 기능을 모두 사용할 수 있도록 Array 클래스를 상속받는 RangeArray 클래스를 구현한다. RangeArray 클래스는 인덱스 범위가 0에서 시작하지 않고, 배열의 시작과 끝이 동적으로 지정될 수 있도록 클래스를 확장한다. 즉, 생성자에서 시작 인덱스와 끝 인덱스를 받아 이를 바탕으로 Array의 생성자를 호출할 때 필요한 배열의 크기를 계산하게 된다. 이를 위해 생성자 내부에서는 시작 인덱스와 끝 인덱스의 차이를 계산하고, 필요한 배열의 크기를 도출하여 부모 클래스의 생성자에게 전달한다. 또한, 클래스 내에 low와 high 변수를 추가하여 사용자가 지정한 인덱스 범위를 관리한다. 이러한 두 변수는 배열의 시작 인덱스와 끝 인덱스의 값을 반환하는 baseValue(), endValue() 함수를 통해 외부에서 접근할 수 있다. 마지막으로 Array에서 정의된 ‘[ ]’ 연산자를 RangeArray에서 재정의하여, 사용자가 지정한 인덱스 범위에 맞게 조정한다.

[자료구조]

Array 클래스에서 int \*data 변수는 동적으로 할당된 정수 배열을 가리키는 포인터이고, int len 변수는 배열의 크기를 저장한다.

RangeArray 클래스에서는 상속된 data와 len 변수를 사용하며, 추가적으로 배열의 시작과 끝 인덱스를 관리하는 low, high 변수를 추가한다. 이를 통해 배열이 기존 0에서 시작하는 인덱스가 아니라 사용자가 지정한 범위로 시작할 수 있게 된다. 사용자 정의 인덱스 범위를 가진 RangeArray 클래스는 Array 클래스의 기능을 확장하며, 상속과 오버로딩을 통해 더 유연하고 강력한 데이터 구조를 제공한다.

이러한 알고리즘 및 자료구조를 통해 수행한 실습 결과는 다음과 같다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

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

[알고리즘]

3주차 과제는 사용자 정의 문자열 처리 클래스인 Str을 구현하는 방식이다. Str 클래스의 알고리즘은 다음과 같다. 먼저 생성자와 소멸자를 살펴보면, Str(int leng) 생성자는 주어진 길이를 가진 빈 문자열을 생성한다. 해당 생성자를 통해 len을 초기화하고, leng에 해당하는 길이만큼의 공간을 str에 동적 할당하게 된다. 또한, Str(char \*neyong) 생성자는 주어진 문자열 neyong으로 Str 객체를 초기화한다. strlen(neyong)을 사용하여 문자열의 길이를 계산하여 len에 저장한 후, str에 메모리를 할당하고 strcpy 함수로 내용을 복사한다. ~Str() 소멸자는 Str 객체가 소멸될 때 동적 할당된 메모리를 해제한다. str에 할당된 메모리를 해제할 때는 delete[]를 사용한다.

Str 클래스의 멤버 함수는 총 4가지가 있는데, int length(void)는 len의 값을 반환하여 문자열의 길이를 제공하고 char \*contents(void)는 str 포인터를 반환하여 문자열의 내용에 접근할 수 있게 해주는 함수이다. 세 번째는 compare 함수이다. int compare(class Str& a)와 int compare(char \*a) 함수는 각각 다른 Str 객체와 C언어 문자열을 현재 객체와 비교한다. 이러한 compare 멤버 함수는 strcmp 함수를 사용하여 문자열의 내용을 비교하고, 비교 결과를 정수로 반환한다.

마지막으로는 operator 함수가 있다. void operator=(char \*a)와 void operator=(class Str& a)는 문자열 대입 연산을 구현하는 함수이다. 해당 연산자는 먼저 현재 str에 할당된 메모리를 해제하고, 새로운 문자열의 길이를 len에 저장한 후 새로운 내용을 str 변수에 동적 할당하여 복사한다.

[자료구조]

Str 클래스는 2개의 변수를 갖는다. 먼저, char \*str은 문자열의 내용을 저장하는 동적 할당된 문자 배열 포인터이다. 해당 포인터는 문자열의 실제 데이터를 가리킨다. 또한, int len은 문자열의 길이를 저장하는 정수형 변수이다. 이 변수는 항상 str 포인터가 가리키는 문자열 데이터의 길이와 일치해야 한다.

이러한 내용을 토대로 main함수를 임의로 작성하고, 프로그램을 실행하면 다음과 같은 결과를 얻을 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

[main 함수 코드]

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

[프로그램 실행 결과]