3주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

주어진 코드에 선언된 두 클래스(Array, RangeArray)에 필요한 멤버 함수와 연산자 오버로딩을

구현한다.

**1. Array 클래스 멤버 함수 및 연산자 오버로딩**

1-1. 생성자

생성자는 new 연산자를 통해 인자로 받은 int size크기의 int 타입 배열을 선언하여 동적으로

메모리를 할당한다. 마지막으로 클래스 변수 len의 값을 size로 업데이트해준다.

1-2. 소멸자

소멸자는 delete 연산자를 통해 할당된 메모리를 해제하는데, 우리는 배열을 해제해야

하기 때문에 delete[] 를 사용한다.

1-3. 배열의 원소에 새로운 값을 대입하는left value용도의 operator[] 오버로딩

인자로 받은 index용도의 int i를 통해 data[i]을 반환하여 배열의 요소를 읽고 쓸 수 있게 한다.

1-4. 배열의 원소 값을 참조하는 right value용도의 operator[] 오버로딩

먼저 배열의 원소 값은 수정이 되면 안되기 때문에 함수 뒤에 const를 붙여 방지한다.

인자로 받은 index용도의 int i를 통해 data[i]을 반환하여 배열의 원소 값을 참조한다.

1-5. 배열의 모든 원소 출력

멤버 변수 len을 이용해 반복문으로 배열을 순회하면서 배열의 각 원소를 출력한다.

**2. Array의 자식 클래스 RangeArray의 멤버 함수 및 연산자 오버로딩**

2-1. 생성자

Array의 생성자를 통해 메모리를 동적 할당한다.

하지만 RangeArray의 범위가 0부터 시작하는 게 아니기 때문에 (i, j) 범위의 배열이

있다고 한다면 Array의 size인자에는 j-i+1을 입력해준다. [ex: (-1,5) 🡪 size = 7]

마지막으로 i는 low에, j는 high에 저장한다.

2-2. 소멸자

Array 클래스를 통해 배열을 생성했기 때문에 Array의 소멸자를 통해 할당된 메모리가 해제된다.

따라서 함수 내부에 추가로 구현할 내용은 없다.

2-3. 가장 낮은 배열 index

배열이 0부터 시작하지 않기 때문에 low값을 반환해준다.

2-4. 가장 높은 배열 index

high값을 반환해준다.

2-5. 배열의 원소에 새로운 값을 대입하는left value용도의 operator[] 오버로딩

Array 클래스 생성자를 통해 배열을 만들었기 때문에 배열의 범위는 0~(high-low)이다.

가장 낮은 음수 i를 이용해 정상적으로 배열에 접근하기 위해서는 i가 0이 되야 한다.

따라서 Array 클래스의 Overloading을 사용하기 때문에 인자에는 i-low를 입력해준다.

2-6. 배열의 원소 값을 참조하는 right value용도의 operator[] 오버로딩

2-5 와 마찬가지로 Array 클래스의 Overloading을 사용하고, 인자에는 i-low를 입력해준다

2-7. 배열의 모든 원소 출력

Overloading을 통해 0~len까지 정상적으로 참조가 가능하기 때문에

기존 Array 클래스의 print 함수를 사용한다.