3주차 예비보고서

전공: 아트&테크놀로지학과 학년: 4학년 학번: 20191048 이름: 김도솔

**1.**

먼저 기본 클래스인 Array 클래스를 구현한다. Array 클래스의 protected 멤버에는 배열을 가리키는 포인터 변수와 배열의 크기를 나타내는 변수를 선언한다. 생성자는 사용자가 입력한 size에 따라 new 연산자를 통해 int 배열을 할당받는다. 소멸자에서는 delete 연산자를 통해 할당된 메모리를 해제하도록 한다. int length() 멤버함수를 통해 배열의 크기를 리턴할 수 있게 한다. 일반적인 배열과 마찬가지로 Array클래스에서도 ‘[]’을 사용해 원소를 참조하거나 값을 대입할 수 있도록 연산자 다중정의를 이용한다. 배열에 원소를 삽입할 수 있는 left value 용도의 int &operator[]를 구현하고, 배열의 원소 값을 반환하는 right value 용도의 int operator []를 구현한다. 두 멤버함수 모두 배열의 인덱스가 범위 밖에 있으면 에러 메시지를 출력하게 한다. 마지막으로 배열의 모든 원소를 출력하는 print() 멤버함수를 구현한다. 다음으로 위에서 구현한 Array 클래스를 상속 받아 RangeArray 클래스를 구현한다. RangeArray 클래스는 인덱스의 범위가 0에서 시작되는 것이 아니라 객체 생성 시 동적으로 정해지므로 protected 멤버에 배열의 인덱스 범위를 나타내는 base와 end 변수를 선언한다. RangeArray 클래스의 생성자는 범위를 입력 받아 배열의 크기를 계산해 상위 클래스인 Array 클래스의 생성자를 통하여 메모리를 할당 받는다. int &operator[], int operator []멤버함수는 배열의 범위에 포함되는 원소에 대해서만 기능하도록 한다. 마지막으로 사용자가 RangeArray 클래스의 배열 범위를 알 수 있게 base값과 end값을 반환하는 baseValue(), endValue() 멤버함수를 추가한다.

**2.**

* **OOP**란 Object-Oriented Programming의 줄임말로, 객체 지향 프로그래밍을 의미한다. 프로그램을 단순히 명령어의 나열로 보는 것이 아니라 프로그래밍 데이터를 추상화해서 상태와 행위로 이루어진 객체라는 기본 단위로 나누고, 그 객체들의 상호작용으로 프로그램 로직을 구성하는 프로그래밍 방법론이다.
* **객체**란 상태를 저장할 변수와 행위를 수행할 메소드를 서로 관련된 것끼리 결합해 만든 것이다. 변수, 함수, 자료 구조 등을 담을 수 있다.
* **클래스**는 객체를 정의하기 위한 속성과 기능의 집합으로 객체지향 프로그램의 기본적인 사용자 정의 데이터형이라고 할 수 있다.
* **인스턴스**는 클래스 정의에 따라 만들어진 객체들을 의미한다.
* **OOP를 쓰는 이유**는 다음과 같다. OOP는 코드 개발과 유지, 보수 측면에서 편리하다. 코드 수정 시 캡슐화라는 특성으로 인해 주변 코드에 미치는 영향이 적다. 또한 추상화를 통한 직관적인 코드 분석이 가능하며 라이브러리나 상속을 통해 코드의 재사용률을 높일 수 있다. 코드의 구조가 유연하고 객체 단위로 설계되기 때문에 업무 분장이 쉬워 대형 프로젝트에 강점을 가진다.
* **상속**은 새로운 클래스가 기존 클래스의 속성이나 기능을 물려받아 사용할 수 있게 해주는 것이다. 이때 기존 클래스를 상위 클래스 또는 부모 클래스라고 하며 상속을 받은 클래스를 파생 클래스 또는 자식 클래스라고 한다.