**2022 Spring OOP Assignment Report**

과제 번호 : ASSN #2 Prob1

학번 : 20210479

이름 : 이주현

Povis ID : lanthanum14

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. **프로그램 개요**
   * Prob1은 “Vector.h”와 “Vector.cpp” 파일의 코드문을 작성하여 STL에 있는 vector를 직접 구현해보는 것을 목적으로 하는 프로그램이다.
   * Prob1에서 작성한 코드문들은 그 자체로 실행을 위한 것이 아니고, 다른 코드문과 link하여 사용할 수 있는 코드문들이다. 따라서 실행을 위해서는 “Vector.h”를 include한 main.cpp가 필요하다.
   * “Vector.h”를 포함한 main.cpp가 실행되면 main 함수 내에서 호출된 Vector class의 멤버 함수들이 실행된다. Vector class 내부에 구현한 모든 public 멤버 함수는 자유롭게 이용 가능하다.
2. **프로그램의 구조 및 알고리즘**

* 이 프로그램에서는 STL에 있는 vector와 같은 기능을 하는 “Vector”라는 class를 직접 구현해 보았다. Vector class는 배열을 저장할 변수 arr, 실제 데이터 길이를 저장할 변수 realLength, 배열에 할당된 데이터 크기를 저장할 변수 arrCapacity라는 private 변수 세 개를 통해 관리되며, STL vector 내의 함수들과 동일한 기능을 하는 여러 멤버 함수들로 구성되어 있다.
* Vector의 생성자에는 기본 생성자, 초기화할 변수와 배열 사이즈를 입력해주는 생성자, 초기화할 배열과 사이즈를 입력해주는 생성자, 다른 벡터 복사 생성자가 있다. 할당된 배열을 삭제하는 소멸자도 존재한다.
* 다른 멤버 함수들의 경우에는 대부분 STL vector와 같은 알고리즘을 사용한다. 다만 STL에는 존재하지 않는 allocate라는 함수도 구현하였다. 이 함수는 새로 할당해줄 데이터 사이즈를 입력받는 함수인데, 문제에 제시된 데이터 할당 규칙을 먼저 따랐다는 점에서 resize와 어느 정도 차이를 갖는다.
* 보통 함수 내에서 다른 변수를 선언하는 경우는 많지 않았으나, 배열에 새로운 크기의 데이터를 할당해야 하는 경우에는 배열 내의 데이터를 그대로 안전하게 보관하기 위해 공통적으로 임시 배열을 선언하여 데이터를 복사해 두고 새로 배열을 할당하는 방법을 사용하였다.
* 또한 배열을 삽입하는 insert 함수를 구현할 경우에는 삽입할 배열의 크기를 알아내기 위한 arrSize라는 변수를 선언하기도 하였으며, overloading된 insert 함수 두 개 모두에서 position으로 주어진 삽입 위치를 배열의 인덱스로 저장하기 위한 ind라는 변수도 선언하였다.

1. **토론 및 개선**
   * + 이번 과제를 통해 class를 구현하는 방법과 생성자, 소멸자 등 class를 구성하는 요소들에 대해서 잘 알 수 있었다.
     + 또한 직접 class를 구현하고자 할 때 header 파일에 적어야 하는 부분과 cpp에 적어야 하는 부분의 차이점을 잘 알 수 있었다.
2. **참고 문헌**
   * + <https://www.cplusplus.com/reference/vector/vector/>
       - STL vector 설명