

개요

제출

편집

코딩 결과

실습 082 - 템플릿 클래스

제출 마감일: 2023-05-14 23:59

업로드 가능한 파일 수: 2

제출 방식: 개인

목적

이 실습은 template class 를 정의하는 연습을 합니다.

설명

지금까지 실습 및 과제를 진행하면서 STL 라이브러리에서 제공하는 std::vector 클래스의 사용 빈도가 높았습니다.

벡터 클래스를 사용할 때는 항상 벡터에 저장할 타입을 함께 제공해 줍니다.

예를 들어, std::vector<int>, std::vector<Student>, std::vector<std::string> 등 입니다.

맞습니다. std::vector 는 템플릿 클래스입니다. 아마도 다음과 비슷하게 정의되어 있을 것입니다.

```
template<typename T>
class vector {
public:
    vector(int n);
    T& operator[ ] (int index);
    ...
private:
    int _size;
    T* _data;
}
```

벡터 타입의 변수 std::vector<int> vec; 를 선언하면, 벡터 템플릿 클래스는 다음과 같이 인스턴스화 됩니다.

```
template<>
class vector {
public:
    vector(int n);
    int& operator[ ] (int index);
    ...
private:
    int _size;
    int* _data;
}
```

std::vector<Student> vec2; 변수를 정의하면, 다음과 같이 인스턴스화 됩니다.

```
template<typename T>
class vector {
public:
    vector(int n);
    Student& operator[ ] (int index);
    ...
private:
    int _size;
    Student* _data;
}
```

문제

my_vector 템플릿 클래스를 정의하시오.

입력

없음

출력

주어진 main.cpp 파일의 표준 출력은 다음과 같다.

```
1 2 3 4 10
```

제출파일

my_vector.h
82.csv

참고

main.cpp -----

```
#include "my_vector.h"
#include <iostream>

int main() {
    using namespace std;
    my_vector<int> vec = {1, 2, 3, 4, 5};
    my_vector<int> vec2(vec); vec2[4] = 10;
    for (int i=0; i<vec2.size(); ++i)
        cout << vec2[i] << ' ';
    cout << '\n';
}
```

my_vector.h -----

my_vector.h

```
#ifndef TEMPLATE_MY_VECTOR_H
#define TEMPLATE_MY_VECTOR_H

#include <algorithm>
#include <initializer_list>
#include <memory>

template<typename T>
class my_vector {
public:
    explicit my_vector(size_t N=0) noexcept ; // 생성자를 구현하시오

    my_vector(std::initializer_list<int> lst) noexcept ; // 생성자를 구현하시오

    my_vector(const my_vector<T>& other) noexcept ; // 복사 생성자를 구현하시오 (deep copy)

    T& operator[](int i) ; // subscript operator, 오퍼레이터 오버로딩 하시오

    size_t size() const // 멤버 함수를 구현하시오

private:
    size_t _size;
    std::unique_ptr<T[]> _data;
};

#endif //TEMPLATE_MY_VECTOR_H
```