클래스 헤더 파일

StandardManager.h

파일명 : 단어 시작 문자는 대문자

```
#pragma once

class StandardManager
{
public:
    StandardManager(void) {}
    ~StandardManager(void) {}
    int showStandard(std::string standard);

protected:

private:
    void _showPrivateStandard1(std::string standard);
    void _showPrivateStandard2(int standard);
    int mMemberVariable1;
    float mMemberVariable2;
};
```

#pragma once
헤더파일 처음에 삽입. 중복 포함 방지

클래스명은 파일명과 동일한 형식
권한 순서는 public, protected, private 의 순서로 작성한다.

public 함수는 첫 단어는 소문자
다음 단어는 시작 문자만 대문자로 작성

private 함수는 public함수와 동일하게 명명하되 앞에 _를 붙인다.

멤버 변수는 앞에 m을 붙이고 명명.

클래스 소스 파일 (1)

StandardManager.cpp

소스파일의 파일명도 헤더파일과 같은 명명 규칙을 따른다. (단어의 시작문자 대문자)

```
#include "StandardManager.h“

#define STANDARD_MANAGER 777
```

#define 정의할 때는 대문자 사용
단어와 단어 사이의 공백은 _로 구분

함수 시작전에 //----------로 구분자를 두어 함수의 시작부를 눈에 잘 띄게 한다.

```
//-------------------------------------------------------------
int StandardManager::showStandard(std::string standardName)
{
    int sumValue = 0;

    for (int i = 0; i < 100; i++)
    {
        sumValue += i;
    }

    return sumValue;
}
```

{ 는 함수이름이나 for/if문에서 줄바꿈 뒤에 열도록 한다.

변수는 첫 단어는 소문자. 이후 단어의 시작문자는 대문자.

클래스 소스 파일 (2)

```
//-------------------------------------------------------------------
void StandardManager::_showPrivateStandard1(std::string standard)
{
    if ("Standard“ == standard)
    {
        std::cout << "Standard" << std::endl;
    }
    else if ("STANDARD“ == standard)
    {
        std::cout << "STANDARD" << std::endl;
    }
    else
    {
        std::count << "standard" << std::endl;
    }
}
```

if문 비교시 상수와 변수가 비교된다면
상수를 앞에 작성하도록 한다.

탭사이즈는 4칸으로 공백으로 설정한다.

클래스 소스 파일 (3)

```
//-------------------------------------------------------------------
void StandardManager::_showPrivateStandard2(int standard)
{
    switch (standard)
    {
        case '1':
            std::cout << "Standard1\n";
            break;
        case '2':
            std::cout << "Standard2\n";
            break;
        case '3':
            std::cout << "Standard3\n";
            break;
        default:
            std::cout << "Standard4\n";
            break;
    }
}
```

switch문 사용시에도
case문을 한탭 들여쓰기 하는것을
원칙으로 한다.