

06. 생성자와 소멸자 그리고 분할 컴파일

실습내용

- 실습#6 : 성적처리프로그램#5 개선
 - 파일명 : 실습과제#6_2022199002_홍길동
 - 구조체를 클래스로 구현
 - Subject 클래스에 함수 추가
 - 생성자와 소멸자 추가
 - 분할 컴파일

실습#6-1 : 클래스화(1)

■ 다음 그림과 같이 클래스화를 진행한다

- Subject의 멤버변수: string SubName, int Hakjum, string Grade, float GPA
- 추가되는 Subject의 멤버함수는 다음 그림과 같음

```
main.cpp
├─ class Subject
│   │   └─ 멤버 변수
│   │   └─ 멤버 함수
├─ class Student
│   │   └─ 멤버 변수
│   │   └─ 멤버 함수
├─ 전역(외부) 함수
│   │   └─ PrintOneData
│   │   └─ PrintAllData
│   │   └─ StdSearch
│   │   └─ PrintMenu
│   │   └─ PrintAllStdList
│   │   └─ ModifyStdInfo
└─ main
```

함수명	설명
Subject()	기본 생성자, 멤버 변수 초기화
void InputData()	과목명, 학점, 등급을 사용자로부터 입력받음
void CalcGPA()	등급에 따라 GPA 계산
void PrintData() const	과목 정보를 한 줄로 출력 (const = 수정 X)
float GetGPA() const	GPA 값을 반환 (const = 수정 X)

- Subject(): Subject 클래스의 생성자, 멤버변수 초기화(원하는 값으로)
- InputData(): 과목명, 학점, 등급을 입력받는 함수(추가할 것)
- CalcGPA(): Student 클래스의 CalcGPA(Subject& sub)의 기능을 대체 **[CalcGPA(Subject& sub)함수는 지우지 말것!]**
- GetGPA(): Subject 클래스의 멤버 변수 GPA값을 반환하는 함수

실습#6-1 : 클래스화(2)

■ 다음 그림과 같이 클래스화를 진행한다

- Student의 멤버변수: string StdName, int Hakbun, int SubNum, Subject* Sub, float AveGPA
- Student의 멤버함수는 그림과 같음

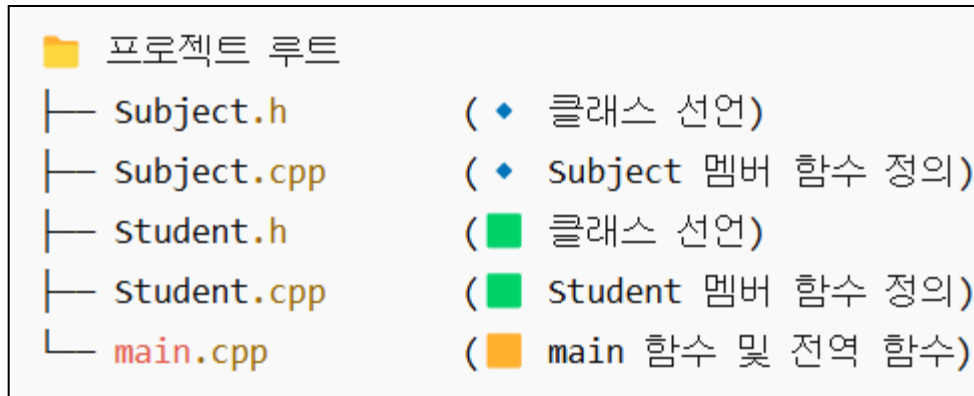
```
main.cpp
├─ class Subject
│   │   └─ 멤버 변수
│   │   └─ 멤버 함수
├─ class Student
│   │   └─ 멤버 변수
│   │   └─ 멤버 함수
├─ 전역(외부) 함수
│   │   └─ PrintOneData
│   │   └─ PrintAllData
│   │   └─ StdSearch
│   │   └─ PrintMenu
│   │   └─ PrintAllStdList
│   │   └─ ModifyStdInfo
└─ main
```

함수명	설명
Student()	기본 생성자, 멤버 변수 초기화
~Student()	소멸자, Sub 배열 메모리 해제
void CalcGPA(Subject& sub)	Subject 객체의 CalcGPA() 호출
float CalcAveGPA(Subject* sub, int SubNum)	모든 과목 GPA 평균 계산
void InputData(Student* pSt, int Student_Num)	학생 데이터 일괄 입력
inline void InputValue(int& i)	사용자로부터 int 입력받고 cin.ignore() 처리
inline void InputValue(string& str)	사용자로부터 string 한 줄 입력받음

- Student(): 생성자로, 멤버변수 초기화(원하는 값으로)
- ~Student(): 소멸자로, 동적 할당된 Subject의 객체를 delete 하도록 구현

실습#6-2 : 분할컴파일

- 다음 그림과 같이 분할한다



- 오른쪽 그림의 포함관계(#include)를 꼭 확인할 것
- 헤더파일 중복방지 꼭 할 것!!
(#pragma once or #ifndef SUBJECT_H... #endif)

