

HW1. Simple class implementation

부산대학교 정보컴퓨터공학부

202055645

신세환

제출일: 2024-03-19

1. 문제정의 -

- Student 클래스를 정의하고 내부 멤버는 getter와 setter로 접근가능하도록 한다.
이 때 대입 연산자 오버로딩도 같이 구현한다.
- **CourseRegistration** 클래스를 정의하고 수업을 듣는 학생 목록을 저장하는 멤버변수를 관리한다.
- main함수에서 student.cpp와 CourseRegistration.cpp를 참조해서 학생 15명 이상, 강좌 3개 이상인 상태를 보여준다. 이 때 makefile파일도 구현해서 사용한다.

2. 클래스 정의 설명

student클래스는 ID, 이름, 주 소, 첫 수강 날짜, 이수한 학점의 수를 멤버로 가지고 getter, setter, 연산자 오버로딩함수를 추가한다.

courseRegistration클래스는 강좌ID, 등록된 학생ID, 수업 시수, 성적을 멤버로 가지고 getter, setter를 추가한다.

3. 클래스의 주요 함수 설명

- Main 함수

```
for(int i = 0; i < students.size(); i++)
{
    int ci = i % courses.size();
    courses[ci].addStudent(students[i]);
    courses[ci].setCourseGrade(students[i].getID(), grade[(i*3)%5]);

    courses[(ci + 1) % 3].addStudent(students[i]);
    courses[(ci + 1) % 3].setCourseGrade(students[i].getID(), grade[(i*2)%5]);
}
```

인덱스 i를 활용해서 비교적 랜덤하게 강의에 학생을 등록하고 성적도 등록한다.

1) Student 클래스 대입 연산자 오버로딩 함수

```
Student& Student::operator=(const Student& rvalue)
{
    if(this != &rvalue)
    {
        id = rvalue.getID();
        name = rvalue.getName();
        address = rvalue.getAddress();
        firstDate.setDate(rvalue.getFirstDate().getDate());
        firstDate.setMonth(rvalue.getFirstDate().getMonth());
    }

    return *this;
}
```

우측 값 레퍼런스를 활용해서 Student클래스의 대입 연산자 오버로딩을 설계한다. 값을 단순히 복사하기만 하고 변경하지 않기 때문에 const로 선언해야 한다. Rvalue 자체가 const형태이기 때문에 Student클래스 자체에서도 const함수를 포함해서 작성한다. 미리 구현된 getter와 setter 함수를 통해 값을 다루고, 연속 대입이 가능하도록 마지막에 객체를 반환해준다.

4. 프로그램 실행

1) 프로그램 실행 환경

- 리눅스는 윈도우 환경에서 WSL을 설치하여 사용함. 우분투 버전은 Ubuntu 22.04.3 LTS를 사용하였음.

2) 프로그램 실행 방법

- Powershell로 wsl 리눅스 환경으로 접근해서 hw1_simple_class 폴더를 새로 만들고 git clone을 수행한다. vi편집기로 프로그램을 작성하고 컴파일한다.

```
sk3456@DESKTOP-PQL1ORV:~/git_project$ mkdir hw1_simple_class
sk3456@DESKTOP-PQL1ORV:~/git_project$ cd hw1_simple_class
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class$ git clone https://github.com/datalab-pnu/fs24hw1-sinsehwan.git
Cloning into 'fs24hw1-sinsehwan'...
Username for 'https://github.com': sinsehwan
Password for 'https://sinsehwan@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/datalab-pnu/fs24hw1-sinsehwan.git/'
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class$ git clone https://github.com/datalab-pnu/fs24hw1-sinsehwan.git
Cloning into 'fs24hw1-sinsehwan'...
Username for 'https://github.com': sinsehwan
Password for 'https://sinsehwan@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class$ ls
fs24hw1-sinsehwan
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class$ cd fs24hw1-sinsehwan/
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class/fs24hw1-sinsehwan$ ls
README.md
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class/fs24hw1-sinsehwan$
```

Makefile 파일을 만들어서 파일이 변경되었을 때 재실행이 용이하게 만든다.

```
CC=g++
all=main
OBJS = main.o student.o courseRegistration.o

%.o : %.cpp
    $(CC) -c -o $@ $<
main : $(OBJS)
    $(CC) -o main $(OBJS)
clean:
    rm -f main $(OBJS)
```

```
ID : 12
Name : student12
Address : address12
number of credit hours : 22
ID : 13
Name : student13
Address : address13
number of credit hours : 23
ID : 14
Name : student14
Address : address14
number of credit hours : 24
```

course Data :

```
course 0
course ID : 100
course Credit hours : 1
students :
student ID : 0, Grade : A+
student ID : 2, Grade : F
student ID : 3, Grade : F
student ID : 5, Grade : A+
student ID : 6, Grade : C+
student ID : 8, Grade : A0
student ID : 9, Grade : B0
student ID : 11, Grade : B0
student ID : 12, Grade : A0
student ID : 14, Grade : C+
```

```
-----
course 1
course ID : 101
course Credit hours : 2
students :
student ID : 0, Grade : A+
student ID : 1, Grade : C+
student ID : 3, Grade : A0
student ID : 4, Grade : B0
student ID : 6, Grade : B0
```

5. Github 화면

```
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class/fs24hw1-sinsehwan$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .vscode/
    Makefile
    courseRegistration.cpp
    courseRegistration.h
    courseRegistration.o
    main
    main.c
    main.cpp
    main.o
    student
    student.cpp
    student.h
    student.o
    test
```

```
nothing added to commit but untracked files present (use "git add" to track)
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class/fs24hw1-sinsehwan$
```

```
nothing added to commit but untracked files present (use "git add" to track)
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class/fs24hw1-sinsehwan$ git add .
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class/fs24hw1-sinsehwan$ git commit -m "add hw1 source code [Add]"
[main 388bd13] add hw1 source code [Add]
14 files changed, 225 insertions(+)
 create mode 100644 .vscode/settings.json
 create mode 100644 Makefile
 create mode 100644 courseRegistration.cpp
 create mode 100644 courseRegistration.h
 create mode 100644 courseRegistration.o
 create mode 100755 main
 create mode 100644 main.c
 create mode 100644 main.cpp
 create mode 100644 main.o
 create mode 100755 student
 create mode 100644 student.cpp
 create mode 100644 student.h
 create mode 100644 student.o
 create mode 100755 test
sk3456@DESKTOP-PQL1ORV:~/git_project/hw1_simple_class/fs24hw1-sinsehwan$ git push
Username for 'https://github.com': sinsehwan
Password for 'https://sinsehwan@github.com':
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 16 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (16/16), 67.50 KiB | 2.70 MiB/s, done.
Total 16 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/data-lab-pnu/fs24hw1-sinsehwan.git
   ce55e78..388bd13  main -> main
```

fs24hw1-sinsehan
Private
Watch 1

main
1 Branch
0 Tags
Go to file
Add file
Code

sinsehan
add hw1 source code [Add]
388bd13 · 12 hours ago
2 Commits

.vscode	add hw1 source code [Add]	12 hours ago
Makefile	add hw1 source code [Add]	12 hours ago
README.md	add deadline	yesterday
courseRegistration.cpp	add hw1 source code [Add]	12 hours ago
courseRegistration.h	add hw1 source code [Add]	12 hours ago
courseRegistration.o	add hw1 source code [Add]	12 hours ago
main	add hw1 source code [Add]	12 hours ago
main.c	add hw1 source code [Add]	12 hours ago
main.cpp	add hw1 source code [Add]	12 hours ago
main.o	add hw1 source code [Add]	12 hours ago
student	add hw1 source code [Add]	12 hours ago
student.cpp	add hw1 source code [Add]	12 hours ago
student.h	add hw1 source code [Add]	12 hours ago
student.o	add hw1 source code [Add]	12 hours ago
test	add hw1 source code [Add]	12 hours ago

6. 논의 사항

- student 클래스 내에 날짜를 표현하기 위해 Date클래스를 student.h에 포함시켰는데 date.cpp, date.h파일로 분리해서 관리하는 게 더 깔끔했을 것 같다.
- ifndef을 활용해서 순환참조를 막는 작업에 좀 더 익숙해졌다.