

창의적 소프트웨어 프로그래밍

실습 과제_04

2-4. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)

- 댓글을 입력하고, 인덱스와 명령어를 통해 삭제하는 프로그램 작성
 - 프로그램을 실행한 뒤, 문자열을 입력하여 댓글을 추가할 수 있다.
 - (개발의 편의를 위해, code skeleton에 임의로 추가된 댓글이 존재한다)
 - 추가된 댓글에는 추가한 순서대로 0부터 번호를 매긴다.
 - 모든 명령어는 '#' 문자로 시작한다.
 - #remove 명령어를 통해 댓글을 삭제할 수 있다.
 - #remove [number]: 특정 번호의 댓글을 삭제한다. (e.g. #remove 3) 번호가 존재하지 않을 경우 아무런 내용도 출력하지 않는다.
 - #remove [number]-[number]: 영역 내의 댓글을 모두 삭제한다. (e.g. #remove 2-5) 영역이 존재하는 댓글을 초과할 경우, 영역 내 존재하는 댓글만 삭제한다.
 - #remove [number],[number], ... ,[number]: 선택된 댓글을 삭제한다. (e.g. #remove 0,1,3) / 존재하는 번호만 삭제한다.
 - #remove 명령어가 성공적으로 실행된 이후에는, 남아있는 댓글 목록을 출력하며, 남아있는 댓글의 번호는 순서대로 다시 매겨진다.
 - 복합 명령어는 구현하지 않으며, 오류로 처리된다. (e.g. #remove 1,2-4)
 - #remove asdf 등 명령어에 부적절한 인수가 입력됐을 때는 아무 내용도 출력하지 않는다(이 경우 일반 문자열로 취급하지 않는다)
- #quit 명령어를 입력하면 프로그램이 종료된다.
- 잘못된 명령어(#abc 등)는 무시한다.

2-4. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)

- 파일명 : replyAdmin (replyAdmin.cpp)
- **입력** : 단순 문자열 or 명령어가 포함된 문자열 (#remove, #quit)
- **출력** : 명령어 입력 후 변화된 댓글 목록을 출력

```
$/replyAdmin
Hello, World!
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 I think male TA looks cool.
6 Hello, World!
#remove 5
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 Hello, World!
```

2-4. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)

```
#remove 0,5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
C++ is so hard
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
Yesterday was too bad.
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
5 Yesterday was too bad.
#remove 4-5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
#quit
$
```

2-4. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)

- Code skeleton

```
#include <iostream>
#include <string>
#include <stdio.h>
using namespace std;

const int NUM_OF_CHAT = 200;

int getChatCount(string *_chatList){
    int i;

    for(i=0; i<NUM_OF_CHAT; ++i){
        string s = _chatList[i];

        if(s.empty() == true)
            break;
    }
    return i;
}

void printChat(string *_chatList){
    int count = getChatCount(_chatList);

    for(int i=0; i<count; ++i){
        cout << i << " " << _chatList[i] << endl;
    }
}
```

3-1. Minimal Fighter

- MinimalFighter class 작성
 - private 멤버 변수
 - int mHp (체력)
 - int mPower (공격력)
 - FighterStatus mStatus
(상태: Invalid, Alive, Dead – defined by enum)
 - 생성자
 - MinimalFighter() : 체력 0, 공격력 0, 상태 Invalid로 초기화
 - MinimalFighter(int _hp, int _power) : 체력과 공격력을 주어진 값으로, 상태 Alive나 Dead로 초기화 (이어지는 규칙 참조)

3-1. Minimal Fighter

- 멤버 함수

- int hp() const
- int power() const
- FighterStatus status() const : 체력, 공격력, 상태를 리턴
- void setHp(int _hp) : 체력을 변경
- void hit(MinimalFighter *_enemy) : Hit 명령 수행 (아래의 규칙 참조)
- void attack(MinimalFighter *_target) : Attack 명령 수행 (아래의 규칙 참조)
- void fight(MinimalFighter *_enemy) : Fight 명령 수행 (아래의 규칙 참조)

3-1. Minimal Fighter

- Minimal Fighter 규칙

- 1. 모든 Fighter는 체력이 0 이하로 떨어지면 죽으며 (status = Dead), 입력된 체력이 0 이하일 경우는 이미 죽어있는 상태라 아래의 모든 행동을 할 수 없다
- 2. Hit - Fighter가 enemy와 공격을 동시에 한 번씩 교환한다
공격을 하면 공격력만큼 상대방의 체력이 감소된다
- 3. Attack - Fighter가 target을 일방적으로 공격한다. target의 체력이 공격력만큼 감소하며, Fighter의 공격력이 0이 된다 Fighter의 체력은 변화가 없다
- 4. Fight - Fighter와 enemy가 둘 중 하나가 죽을 때까지 공격을 교환한다
동시에 죽을 경우 둘 다 죽은 것으로 처리한다
- 5. 예시
H2, P1 hit H1, P1 → H1, P1 / DEAD (H0, P1)
H3, P1 hit H3, P1 → H2, P1 / H2, P1
H1, P3 attack H3, P1 → H1, P0 / DEAD
H4, P1 fight H3, P1 → H3, P1 / H2, P1 → H2, P1 / H1, P1 → H1, P1 / DEAD

3-1. Minimal Fighter

- 작성 규칙
 - 주어진 skeleton에서 MinimalFighter 클래스를 구현한다 (인터페이스, 메인 함수 부분은 수정하지 않음)
 - 위의 규칙을 모두 반영할 수 있도록 구현한다
 - 올바르지 않은 입력이 들어온 경우 종료
- 파일명 : minimal_fighter
 - (minimal_fighter.cc, minimal_fighter.h, minimal_fighter_main.cc)
- **입력** : 양의 정수나 0인 Fighter1의 체력과 공격력, 액션(H, A, F), 양의 정수나 0인 Fighter2의 체력과 공격력
- **출력** : 액션 이후 두 Fighter의 상태를 출력 (fight의 경우 최종 결과만 출력)

3-1. Minimal Fighter

```
./minimal_fighter
2 1 H 1 1
H1, P1 / DEAD
3 1 H 3 1
H2, P1 / H2, P1
1 3 A 3 1
H1, P0 / DEAD
4 1 F 3 1
H1, P1 / DEAD
q
$
```

제출 양식 및 제출 기한

- 제출 양식

- 학번_파일명.cpp파일을 압축해서 학번.zip으로 제출
(ex. 2018120511.zip)
- 각각 파일명
 - 2018120511_replyAdmin.cpp
 - 2018120511_minimal_fighter.cpp

- 제출 기한

- 6/28 19:00까지
(추가제출기한 6/28 19:01~6/29 12:00까지)