

창의적 소프트웨어 설계 실습 문제 06 - hw6-3

제출 기한

- 10 월 28 일 일 23:59 까지
- I. hw6-3(mkdir hw6-3)라는 폴더를 만들고 GitLab 에 push
- II. hw6-3 디렉토리에 Makefile 과 각 문제에서 요구하는 파일들을 작성
- III. **'make'명령을 수행하여 숙제가 빌드**
- IV. 최종 버전을 GitLab 에 commit
- V. **시간과 파일명**, 입력과 출력 방식 반드시 지키기

과제 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 Invalid 로 초기화

MinimalFighter(int _hp, int _power) : 체력과 공격력을 주어진 값으로, 상태 Alive 나

Dead 로

초기화 (아래의 규칙 참조)

- 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)

- Minimal Fighter 규칙

1. 모든 Fighter 는 체력이 0 이하로 떨어지면 죽으며 (status = Dead), 입력된 체력이 0 이하일 경우는 이미 죽어있는 상태라 아래의 모든 행동을 할 수 없다.
2. Hit - Fighter 가 enemy 와 공격을 동시에 한 번씩 교환한다. 공격을 하면 공격력만큼 상대방의 체력이 감소된다. 동시에 죽을 경우 둘 다 죽은 것으로 처리한다.
3. Attack - Fighter 가 target 을 일방적으로 공격한다. target 의 체력이 공격력만큼 감소하며, Fighter 의 공격력이 0 이 된다. Fighter 의 체력은 변화가 없다.
4. Fight - Fighter 와 enemy 가 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

작성 규칙

- MinimalFighter 클래스를 구현한다.
- 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 0 인 Fighter1 의 체력과 공격력, 액션(H, A, F), 양의 정수나 0 인 Fighter2 의 체력과 공격력

출력 : 액션 이후 두 Fighter 의 상태를 출력 (fight 의 경우 최종 결과만 출력)

```
$ ./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
\$