

Lec. 12 Inheritance

이름: _____

학번: _____

1. 광물을 채집하여 Gold를 모으고, 병사를 고용하여 적군을 물리치는 turn제 게임을 만드시오.
(총 50 turn 내에 적군의 체력(500)을 0이하로 만들면 승리, **[시작코드]**는 변경하지 말 것)

[요구사항]

- Unit Class는 abstract class이며 Attacker(공격형 유닛) 및 Miner(채집형 유닛) Class의 Base Class임. (action() 및 print() 순수 가상 함수를 가짐)
- Unit Class의 멤버변수는 type(0=공격형 유닛, 1=채집형 유닛), name(유닛의 이름), price(유닛의 가격)이며, 각 멤버 변수를 반환하는 get_type(), get_name(), get_price() 멤버함수를 가짐
- Attacker Class는 attack_point(공격력) 멤버 변수를 가지며, action()은 attack_point를 반환
- Miner Class는 mining_point(채집력) 멤버 변수를 가지며, action()은 mining_point를 반환
- Attacker Class와 Miner Class의 생성자 입력변수는(type, name, price, attack_point(또는 mining point)임.
- Attack Class와 Miner Class의 print() 결과 예제는 아래와 같음

Miner Class의 print()	[Miner] Name:farmer (Mining point: 20, Price: 200)
Attack Class의 print()	[Attacker] Name: pikeman (Attack point: 4, Price: 60)

- ApplicationType Class의 멤버변수 unit_list_in_barracks는 사용자(게이머)가 고용할 수 있는 unit목록을 나타내며(총 4종류), unit_list는 현재 사용자가 고용한 unit들의 목록, curr_turn은 게임 중 현재 turn(시작시 1), MAX_turn은 게임이 끝날 때 까지 사용가능한 총 turn(=50), gold는 현재 사용가능한(채집한) 금의 양(시작시 1000), enemy_hp는 적군의 체력값(시작시 500), flag_status는 현재 게임의 상태(0:진행중, 1:패배, 2:승리)
- ApplicationType의 소멸자에서는 unit_list_in_barracks 및 unit_list의 모든 element를 할당 해제함.
- Incrase_turn()은 curr_turn을 1 증가시킨 다음, 만약 curr_turn이 MAX_turn과 같아지면 flag_status를 1(패배)로 바꾸고, "You lose"를 화면에 출력
- print_unit_list_in_barracks() 및 print_unit_list()는 각각 unit_list_in_barracks 및 unit_list의 element들을 화면에 출력 (반드시 simplified for문과 각 객체의 print()함수를 사용할 것)
- get_command()는 아래의 정보를 화면에 출력하고, 키보드로부터 command를 입력으로 받아 반환

```
Turn: 1/50
Gold: 1000
Enemy HP: 500

1. Hire Unit
2. Attack to Enemy
3. Gather Money
4. Print Units
Input>1
```

- hire_unit()은 1) print_unit_list_in_barracks() 함수를 호출하여 고용 가능한 전체 유닛을 화면에 출력, 2) 고용하고 싶은 unit의 이름을 키보드를 통해 입력 받고, 3) 입력받은 이름이 unit_list_in_barracks에 존재하고, 현재 보유한 gold가 unit의 price보다 작은 경우 unit을 구매(unit_list에 구매한 unit을 추가, unit은 1 turn에 1명만 고용 가능), 4-1) 성공 시 "Successfully hired!"를 화면에 출력하고 increase_turn()을 이용해서 turn을 1 증가시킨 다음 true를 return, 4-2)실패 시 "Error: neither wrong name nor enough gold"를 화면에 출력하고 false를 return.
- attack_to_enemy(): 1) 현재 보유한 unit들 중 공격형 유닛의 총 공격력(attack_point들의 합)만큼 적군의 체력(enemy_hp)을 감소, 2) incrase_turn()을 통해 1턴을 증가시킴, 3) 만약 enemy_hp가 0보다 작으면 flag_status는 2(승리)로 바뀌고 "You win"이 화면에 출력.
- gather_money(): 1) 현재 보유한 unit의 mining_point 총 합만큼 gold를 증가시킴. 2) incrase_turn()을 이용해서 turn을 1 증가
- 모든 클래스는 .h와 .cpp파일로 분리되어야 함(Unit Class는 .cpp필요 없음), main()을 포함하는 파일에는 ApplicationType클래스의 header 파일만 추가할 것

[시작코드]

```
class Unit {
protected:
    int type;
    string name;
    int price;
public:
    /* 메소드 구현 */
};

class Attacker: public Unit {
protected:
    int attack_point;
public:
    /* 메소드 구현 */
};

class Miner : public Unit {
protected:
    int mining_point;
public:
    /* 메소드 구현 */
};

class ApplicationType {
private:
    vector<Unit*> unit_list_in_barracks; // 고용가능한 unit list
    vector<Unit*> unit_list; // 고용된 unit list
    int curr_turn; // 현재 turn
    int MAX_turn; // 최대 turn
    int gold; // 유저가 사용가능한 금의 양
    int enemy_hp; // 적군의 체력
    int flag_status; // 게임 상태
};
```

```

public:
    ApplicationType();
    ~ApplicationType();

    void increase_turn();
    void print_unit_list_in_barracks();
    void print_unit_list();

    void run();
    string get_command();
    bool hire_unit();
    void attack_to_enemy();
    void gather_money();
};

ApplicationType::ApplicationType() {
    /* unit_list_in_barracks에 Attacker(1, "swordsman", 300, 10) 추가*/
    /* unit_list_in_barracks에 Attacker(1, "champion", 1200, 16) 추가*/
    /* unit_list_in_barracks에 Miner(0, "farmer", 200, 20) 추가*/
    /* unit_list_in_barracks에 Miner(0, "landlord", 500, 50) 추가*/
    curr_turn = 1; // 현재 turn (1턴부터 시작)
    MAX_turn = 50; // 종료될 때 까지의 총 turn
    gold = 1000; // 유저의 금의 양
    enemy_hp = 500; // 적군의 체력
    flag_status = 0;
}

void ApplicationType::ApplicationType::run() {
    string sOption;
    while (flag_status == 0) {
        sOption = get_command();
        if (sOption == "1") hire_unit();
        else if (sOption == "2") attack_to_enemy();
        else if (sOption == "3") gather_money();
        else if (sOption == "4") print_unit_list();
        else if (sOption == "5") break;
    }
    cout << "Exit the program..." << endl;
}

int main() {
    ApplicationType appType;
    appType.run();
    return 0;
}

```

[실행화면]

(1-page)	(2-page)
<p>Turn: 1/50 Gold: 1000 Enemy HP: 500</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input>1 [Attacker] Name: swordsman (Attack point: 10, Price: 300) [Attacker] Name: champion (Attack point: 16, Price: 1200) [Miner] Name:farmer (Mining point: 20, Price: 200) [Miner] Name:landlord (Mining point: 50, Price: 500)</p> <p>Please enter the name: swordsman Successfully hired!</p> <p>Turn: 2/50 Gold: 700 Enemy HP: 500</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input>1 [Attacker] Name: swordsman (Attack point: 10, Price: 300) [Attacker] Name: champion (Attack point: 16, Price: 1200) [Miner] Name:farmer (Mining point: 20, Price: 200) [Miner] Name:landlord (Mining point: 50, Price: 500)</p> <p>Please enter the name: swordsman Successfully hired!</p> <p>Turn: 3/50 Gold: 400 Enemy HP: 500</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input>1 [Attacker] Name: swordsman (Attack point: 10, Price: 300) [Attacker] Name: champion (Attack point: 16, Price: 1200) [Miner] Name:farmer (Mining point: 20, Price: 200) [Miner] Name:landlord (Mining point: 50, Price: 500)</p> <p>Please enter the name: farmer Successfully hired!</p> <p>Turn: 4/50 Gold: 200 Enemy HP: 500</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input>1 [Attacker] Name: swordsman (Attack point: 10, Price: 300) [Attacker] Name: champion (Attack point: 16, Price: 1200) [Miner] Name:farmer (Mining point: 20, Price: 200) [Miner] Name:landlord (Mining point: 50, Price: 500)</p> <p>Please enter the name: asdfasdf Error: neither wrong name nor enough gold</p>	<p>Turn: 4/50 Gold: 200 Enemy HP: 500</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input>4 [Attacker] Name: swordsman (Attack point: 10, Price: 300) [Attacker] Name: swordsman (Attack point: 10, Price: 300) [Miner] Name:farmer (Mining point: 20, Price: 200)</p> <p>Turn: 4/50 Gold: 200 Enemy HP: 500</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input>2</p> <p>Turn: 5/50 Gold: 200 Enemy HP: 480</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input>3</p> <p>Turn: 6/50 Gold: 220 Enemy HP: 480</p> <p>1. Hire Unit 2. Attack to Enemy 3. Gather Money 4. Print Units Input></p>