객체지향언어1 13주차 과제

2071012 이연준

1. 주제 제목

클래스를 이용한 간단한 게임 시뮬레이터

2. 코드 설명

클래스를 이용한 간단한 게임 시뮬레이터 입니다. 먼저 게임캐릭터가 존재하기에 캐릭터 클래스를 만들었고, RPG 게임에는 직업군 이라는 특징이 있기 때문에 캐릭터 클래스를 상속받는 직업군 클래스를 만들었습니다. 또한 플레이어가 캐릭터와 직업을 고르고 이름도 지어주기때문에 이 모든걸 지시할 수 있는 직업군 클래스를 상속받는 플레이어 클래스를 만들었습니다.

캐릭터 클래스에는 체력과 공격력이 들어갑니다. 따라서 공격력값과 체력값을 매개변수로 받는 생성자를 만들었습니다. 시뮬레이터이기 때문에 공격에 대한 체력감소를 표현하기 위해 Attack함수를 클래스 외부에 선언해주고 프렌드함수로 캐릭터 클래스에 넣어줬습니다.

Attack함수에 쓰이는 체력값을 반환해주는 getHealth함수와 캐릭터에 대한 정보를 표시해주는 show함수도 작성하였습니다.

직업군 클래스는 캐릭터 클래스를 상속받습니다. 따라서 직업군, 공격력, 체력을 매개변수로 받는 생성자를 만들어 캐릭터 클래스의 매개변수 있는 생성자를 호출했습니다. 그러나 고유값인 직업정보가 필요하기 때문에 직업내용을 private 변수로 설정하였습니다.

직업을 바꿔주는 changeRole함수를 설정했고, 직업정보를 표현하는동시에 캐릭터 클래스의 캐릭터 정보를 표현해주는 show함수를 포함하는 showRole함수를 작성했습니다.

플레이어 클래스는 직업군 클래스를 상속받습니다. 그래서 플레이어가 설정할 수 있는 고유값인 플레이어 이름을 private변수로 설정해주었습니다. 또한 직업군 클래스를 상속받기때문에 플레이어 이름, 직업군, 공격력, 체력을 매개변수로 받는 생성자를 만들어 직업군 클래스의 매개변수 있는 생성자를 호출했습니다.

직업군 클래스와 마찬가지로 플레이어 이름을 바꿔주는 changeName함수를 설정했고, 플레이어 이름과 상속받는 직업군 클래스의 정보를 표현해주는 showRole함수를 포함하는 showPlayer함수를 작성했습니다.

몬스터 클래스는 캐릭터 클래스를 상속받습니다. 몬스터는 직업이 존재하지 않기 때문에 캐릭터 클래스를 상속시켰습니다. 전과 비슷한 캐릭터 클래스를 상속받는 직업군 클래스와는 다르게 몬스터 클래스에는 고유값인 몬스터 이름의 변수를 private에 생성하였습니다. 몬스터는 보통 개발자가 이름을 정하는 경우가 많으므로 이름변경함수는 생성하지 않고 몬스터의 정보를 표현해주는 showMonster함수를 설정하였습니다.

생성자로는 몬스터 이름, 공격력, 체력을 매개변수로 받아 캐릭터의 매개변수 있는 생성자를 호출하게 했습니다.

공격함수를 클래스 외부에 작성하였는데 이 공격함수를 캐릭터 클래스의 프렌드 함수로 설정해 공격함수에서 캐릭터 클래스의 private 변수값에 접근 할 수 있도록 하였습니다.

메인함수에서는 pDer포인터 변수로 Warrior클래스를 가리키게 하였고 업캐스팅 해주어 Warrior의 클래스인 플레이어 클래스의 상위 클래스인 직업군 클래스에 pBase로 접근이 가능하게 해 주었습니다.

순차적으로 프로그램에 아무숫자나 입력하면 프로그램이 진행되도록 하였습니다.

이름을 변경하는 절차에 들어섰을땐 다운캐스팅을 통하여 타입변환을 해주어 혼동이 오지 않고 정상적으로 컴파일되게 하였습니다.

3. 프로그램 코드

#include <iostream>

#include <string>

using namespace std;

// 캐릭터 클래스

class Character {

private:

int attack, health; // 공격력, 방어력, 체력

public:

Character() { attack = 0; health = 0; } // 아무것도 설정해주지 않았을때 기본 캐릭터

Character(int attack, int health) { this->attack = attack; this->health = health;} // 매개변수를 이용한 캐릭터 생성자

int getHealth() {return health;} //체력값을 출력하는 함수

void show() { cout << " 공격력 : " << attack << " 체력 : " << health << endl; } // 정보 표시 함수

friend void Attack(Character &op1, Character &op2);

};

// 직업군 클래스

class Role : public Character { //캐릭터 클래스 상속

private:

string role; // 직업군

public:

Role() { role = "기본직업"; } // 기본 직업 설정

Role(string role, int attack, int health) : Character(attack, health) { this->role = role; }

void changeRole(string role) {this->role = role;} //직업 변경 함수

void showRole() { cout << " 직업 : [" << role << "]"; show(); } // 직업군 표시 함수

};

// 플레이어 클래스

class Player : public Role { //직업군 클래스 상속

private:

string name; // 플레이어 이름

public:

Player() { name = "Player"; } // 기본 플레이어 생성

Player(string name, string role, int attack, int health) : Role(role, attack, health) { this->name = name; }

// 플레이어 정보들을 입력받아 매개변수 있는 생성자에 정보를 전달

void changeName(string name) {this->name = name;} //플레이어 이름 변경 함수

void showPlayer() { cout << " 이름 : [" << name << "]"; showRole(); } // 플레이어 표시 함수

};

// 몬스터 클래스

class Monster : public Character { //캐릭터 클래스 상속

private:

string name; // 몬스터 이름

public:

Monster() { name = "Monster"; } // 기본 몬스터 생성

Monster(string name, int attack, int health) : Character(attack, health) { this->name = name; }

// 몬스터 정보들을 입력받아 매개변수 있는 생성자에 정보를 전달

void showMonster() { cout << " 이름 : [" << name << "]"; show(); } // 몬스터 표시 함수

};

//공격함수

void Attack(Character &op1, Character &op2) {

op2.health = op2.health - op1.attack;

}

int main() {

cout << "게임 시뮬레이터 시작" << endl;

Player Warrior("김씨", "전사", 10, 50);

Player \*pDer = &Warrior;

Role \*pBase = pDer; // 업 캐스팅

Monster Monster("몬스터", 4, 100);

Warrior.showPlayer();

Monster.showMonster();

int command;

cout << "다음으로 진행하려면 아무 숫자나 입력하시오." << endl;

cin >> command;

if(command != NULL) {

cout << "플레이어가 몬스터를 공격하게 하기" << endl;

Attack(Warrior, Monster);

cout << "플레이어가 몬스터를 공격했습니다." << endl;

pDer->showPlayer();

Monster.showMonster();

}

cout << "다음으로 진행하려면 아무 숫자나 입력하시오." << endl;

cin >> command;

if (command != NULL) {

cout << "몬스터가 플레이어를 공격하게 하기" << endl;

Attack(Monster, Warrior);

cout << "몬스터가 플레이어를 공격했습니다." << endl;

pDer->showPlayer();

Monster.showMonster();

}

cout << "다음으로 진행하려면 아무 숫자나 입력하시오." << endl;

cin >> command;

if (command != NULL) {

cout << "플레이어의 직업을 변경하기" << endl;

string changeRole;

cout << "변경할 직업을 입력하시오." << endl;

cin >> changeRole;

pBase->changeRole(changeRole); //업캐스팅한 포인터로 직업군 클래스를 가리켜 직업을 변경하는 함수 사용

cout << "플레이어 직업을 변경하였습니다." << endl;

Warrior.showPlayer();

Monster.showMonster();

}

cout << "다음으로 진행하려면 아무 숫자나 입력하시오." << endl;

cin >> command;

if (command != NULL) {

cout << "플레이어의 이름을 변경하기" << endl;

pDer = (Player \*)pBase; //다운 캐스팅

string changeName;

cout << "변경할 이름을 입력하시오." << endl;

cin >> changeName;

pDer->changeName(changeName);

cout << "플레이어의 이름을 변경하였습니다." << endl;

Warrior.showPlayer();

Monster.showMonster();

}

cout << endl;

cout << "게임 시뮬레이터가 종료되었습니다." << endl;

}

4. 실행화면

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명