小李打妖怪

/\*\*

\* author: xmmmmmovo

\* generation time: 2019/05/22

\* filename: MR.LiBeatMonstor.cpp

\* language & build version : C++ 11

\*/

#include <iostream>

#include <cstdio>

#include <cstdlib>

#include <algorithm>

#include <string>

#include <vector>

using *namespace* std;

*class* Monstor

{

*private:*

string name;

*int* hp;

*public:*

Monstor(string *n*, *int* *h*)

: name(n), hp(h){}

*void* Hurt(*int* *decrease\_hp*) {

cout << name << " hurt!";

if (decrease\_hp >= hp) {

Killed();

return;

}

this->hp -= decrease\_hp;

cout << name << " now HP : " << hp << endl;

}

*void* Killed() {

cout << name << " was killed !" << endl;

}

*int* GetHp() {

return hp;

}

};

*class* Weapons

{

*private:*

*int* attack\_damage = 1;

*public:*

Weapons(){}

*void* SetAttackDamage(*int* *d*) {

attack\_damage = d;

}

*int* GetAttackDamage() {

return attack\_damage;

}

virtual *void* AttackTarget(Monstor &*mon*) = 0;

};

*class* WoodSword : *public* Weapons

{

*public:*

WoodSword() {

SetAttackDamage(50);

}

*void* AttackTarget(Monstor &*mon*) {

mon.Hurt(GetAttackDamage());

}

};

*class* IronSword: *public* Weapons

{

*public:*

IronSword() {

SetAttackDamage(100);

}

*void* AttackTarget(Monstor &*mon*) {

mon.Hurt(GetAttackDamage());

}

};

*class* MagicSword : *public* Weapons

{

*public:*

MagicSword(){

SetAttackDamage(200);

}

*void* AttackTarget(Monstor &*mon*) {

mon.Hurt(GetAttackDamage());

}

};

/\*\*

\* 小李类

\* 未作线程安全的一个单例设计模式

\*/

*class* LiPlayer

{

*private:*

LiPlayer() {}

LiPlayer(const LiPlayer &);

LiPlayer & operator= (const LiPlayer &);

string name = "xiaoli";

*public:*

Weapons \*weapon;

static LiPlayer \* GetInstance() {

static LiPlayer li\_instance;

return &li\_instance;

}

*void* Attack(Monstor &*mon*) {

weapon->AttackTarget(mon);

}

};

*int* main(*int* *argc*, *char* const \**argv*[]) {

Monstor m1("Slime", 100);

Monstor m2("Slime2", 150);

Monstor m3("demo", 250);

LiPlayer \*player = LiPlayer::GetInstance();

player->weapon = new WoodSword();

player->Attack(m1);

player->Attack(m1);

player->weapon = new IronSword();

player->Attack(m2);

player->Attack(m2);

player->weapon = new MagicSword();

player->Attack(m3);

player->Attack(m3);

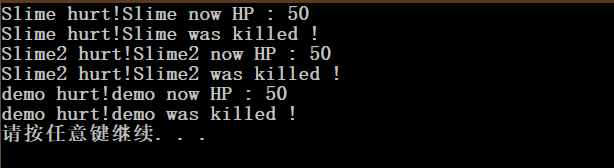
// cout << m1.GetHp() << endl;

// player->Attack(m1);

system("pause");

return 0;

}



老张开车去东北

#include<iostream>

#include<algorithm>

#include<string>

using *namespace* std;

*class* Address {

*public:*

    string address ;

     Address(string *addr*) : address(addr) {

    }

};

*class* Driver {

*public:*

    string name;

    Driver(string *name1*): name(name1) {

    }

};

*class* Vehicle {

*public:*

    virtual *void* go(Address *dest*, Driver *dri*) {}

};

*class* Car : *public* Vehicle{

*public:*

*void* go(Address *dest*, Driver *dri*) { cout << dri.name << "开小汽车去" << dest.address; }

};

*class* Plain : *public* Vehicle {

*public:*

*void* go(Address *dest*, Driver *dri*) { cout << dri.name << "开飞机去" << dest.address; }

};

*class* Train : *public* Vehicle {

*public:*

*void* go(Address *dest*, Driver *dri*) { cout << dri.name << "开火车去" << dest.address; }

};

*int* main() {

    Address address("东北");

    Driver driver("老张");

    Vehicle ve;

    Car car;

    Plain plain;

    Train train;

*int* i;

while (cin >> i) {

if (i == 1) {

car.go(address, driver);

cout << endl;

}

if (i == 2) {

plain.go(address, driver);

cout << endl;

}

if (i == 3) {

train.go(address, driver);

cout << endl;

}

}

    system("pause");

}

