计算机学院 高级语言程序设计 课程实验报告

实验题目：魔兽世界(二)装备		学号：202200400053
日期：2024-05-25	班级： 2202	姓名： 王宇涵
Email：1941497679@qq.com		

实验目的：

锻炼面向对象综合设计能力.

实验软件和硬件环境：

软件环境 ：Clion

硬件环境 ：Legion Y7000P

实验步骤与内容：

AC 截图如下

状态：Accepted

源代码

```
#include "bits/stdc++.h"
using namespace std;
class weapon{
public:
    int weaponID;//武器编号
    int weaponAttack;//武器攻击力
    weapon(int weaponID_, int weaponAttack_){
        weaponID = weaponID_;
```

基本信息

#: 45275057
题目: 2
提交人: fzzh
内存: 248kB
时间: 2ms
语言: G++
提交时间: 2024-06-14 14:28:53

测试输入

1

20

3 4 5 6 7

测试输出

Case:1

000 red iceman 1 born with strength 5,1 iceman in red headquarter

It has a bomb

000 blue lion 1 born with strength 6,1 lion in blue headquarter

It's loyalty is 14

001 red lion 2 born with strength 6,1 lion in red headquarter

It's loyalty is 9

001 blue dragon 2 born with strength 3,1 dragon in blue headquarter

It has a arrow,and it's morale is 3.67

002 red wolf 3 born with strength 7,1 wolf in red headquarter

002 blue ninja 3 born with strength 4,1 ninja in blue headquarter

It has a sword and a bomb

003 red headquarter stops making warriors

003 blue iceman 4 born with strength 5,1 iceman in blue headquarter

It has a bomb

004 blue headquarter stops making warriors

需求分析

本次实验相比于实验一，主要多了生成武士有了特定的种类，需要根据不同的种类进行对应的信息输出.

设计思路

使用 warrior 作为武士虚基类，派生出 dragon，ninja，iceman，lion，wolf 武士，并实现多态的信息打印.

不同的武士类具体实现如下:

武士基类

```
class warrior{
public:
    int lifeValue;//生命值
    int attack;//攻击力
    int id;//编号
};
```

Dragon

```
class dragon: public warrior{
private:
    double morale; //士气
    int weaponID; //武器编号
    map<int, int> weapons;//编号 -> 数量
public:
    dragon(int id_, int lifeValue_, double remainingLifeValue){
        id = id_;
        weaponID = id % 3;
        if (weaponID == 0) {//sword
            weapons[0] = 1;
        } else if (weaponID == 1) {//bomb
            weapons[1] = 1;
        } else {//arrow
            weapons[2] = 1;
        }
        lifeValue = lifeValue_;
        morale = remainingLifeValue / (double) lifeValue_;
    }
    double getMorale(){
        return morale;
    }
    int getWeaponID(){
        return weaponID;
    }
    void print() {
        string weaponName ;
        if (weaponID == 0) {
            weaponName = "sword";
        } else if (weaponID == 1) {
            weaponName = "bomb";
        } else {
            weaponName = "arrow";
        }
        cout << "It has a " <<weaponName <<",and it's morale is " << fixed << setprecision( n: 2)<< morale << endl;
```

Ninja

```cpp
class ninja: public warrior{
private:
    int weaponID1;
    int weaponID2;
    map<int, int> weapons;//编号 -> 数量
public:
    ninja(int id_, int lifeValue_){
        id = id_;
        lifeValue = lifeValue_;
        weaponID1 = id % 3;
        weaponID2 = (id + 1) % 3;
        if (weaponID1 == 0) {//sword
            weapons[0] = 1;
        } else if (weaponID1 == 1) {//bomb
            weapons[1] = 1;
        } else {//arrow
            weapons[2] = 1;
        }
    }
    int getWeaponID1(){
        return weaponID1;
    }
    int getWeaponID2(){
        return weaponID2;
    }
    void print() {
        string weaponName1, weaponName2;
        if (weaponID1 == 0) {
            weaponName1 = "sword";
        } else if (weaponID1 == 1) {
            weaponName1 = "bomb";
        } else {
            weaponName1 = "arrow";
        }
        if (weaponID2 == 0) {
            weaponName2 = "sword";
        } else if (weaponID2 == 1) {
            weaponName2 = "bomb";
        } else {
            weaponName2 = "arrow";
        }
        cout << "It has a " << weaponName1 << " and a " << weaponName2 << endl;
    }
};
```

Iceman

```cpp
class iceman: public warrior{
private:
    int weaponID;
    map<int, int> weapons;//编号 -> 数量
public:
    iceman(int id_, int lifeValue_){
        id = id_;
        weaponID = id % 3;
        if (weaponID == 0) {//sword
            weapons[0] = 1;
        } else if (weaponID == 1) {//bomb
            weapons[1] = 1;
        } else {//arrow
            weapons[2] = 1;
        }
        lifeValue = lifeValue_;
    }
    int getWeaponID(){
        return weaponID;
    }
    void print() {
        string weaponName ;
        if (weaponID == 0) {
            weaponName = "sword";
        } else if (weaponID == 1) {
            weaponName = "bomb";
        } else {
            weaponName = "arrow";
        }
        cout << "It has a " << weaponName << endl;
    }
};
```

Lion

```
class lion: public warrior{
private:
    int loyalty;
    int weaponID;
    map<int, int> weapons;//编号 -> 数量
public:
    lion(int id_, int lifeValue_, int remainingLifeValue){
        id = id_;
        loyalty = remainingLifeValue;
        lifeValue = lifeValue_;
        weaponID = id % 3;
        if (weaponID == 0) {//sword
            weapons[0] = 1;
        } else if (weaponID == 1) {//bomb
            weapons[1] = 1;
        } else {//arrow
            weapons[2] = 1;
        }
    }
    int getLoyalty(){
        return loyalty;
    }
    void print() {
        cout << "It's loyalty is " << loyalty << endl;
    }
};
```

Wolf

```
,,
class wolf: public warrior{
private:
    vector<weapon> weapons;
public:
    wolf(int id_, int lifeValue_){
        id = id_;
        lifeValue = lifeValue_;
    }
};
```

制造武士函数 makeWarrior 核心代码如下:

```
if (makingSeq[curMakingIndex] == 1) {//如果是dragon
    dragon* curWarrior = new dragon ( id_: curId,  lifeValue_: lifeValue,  remainingLifeValue: totalLifeValue);
    city2Warrior[0][color] = curWarrior;
    curWarrior->print();
} else if (makingSeq[curMakingIndex] == 2) {//如果是ninja
    ninja * curWarrior = new ninja( id_: curId,  lifeValue_: lifeValue);
    city2Warrior[0][color] = curWarrior;
    curWarrior->print();
} else if (makingSeq[curMakingIndex] == 3) {//如果是iceman
    iceman* curWarrior = new iceman( id_: curId,  lifeValue_: lifeValue);
    city2Warrior[0][color] = curWarrior;
    curWarrior->print();
} else if (makingSeq[curMakingIndex] == 4) {//如果是lion
    lion* curWarrior = new lion( id_: curId,  lifeValue_: lifeValue,  remainingLifeValue: totalLifeValue);
    city2Warrior[0][color] = curWarrior;
    curWarrior->print();
} else if (makingSeq[curMakingIndex] == 5) {//如果是wolf
}
```

功能划分

1. headQuarter.h 司令部类的函数定义和实现
2. warrior.h 武士类的定义和打印信息函数实现
3. main.cpp 输入输出和对应测试函数

结论分析与体会：

本次实验我独立设计了武士类，维护了内部变量，完成了生成特定种类武士的功能，深化了自己的面向对象思维，提高了自己的编程能力，锻炼了面向对象综合设计能力，也更加期待以后的挑战！已经装备了，是否要开战了呢？

就实验过程中遇到的问题及解决处理方法，自拟 1－3 道问答题：

1. 如何判断高效实现不同武士类？

答 ： 可以先实现一个武士基类，再通过武士基类的指针来实现多态的信息打印

2. 如何初始化武士不同的信息？

答 ： 通过不同武士类特定的含参构造函数来实现，比如 lion 需要此时司令部剩余的生命元的数目，则需要传入生命元的数目.