**C++实验报告三**

1. 设计程序的逻辑思路

C++中的类也是一种构造类型，但是进行了一些扩展，类的成员不但可以是变量，还可以是函数；通过类定义出来的变量也有特定的称呼，叫做“对象”。通过结构体定义出来的变量还是叫变量，而通过类定义出来的变量有了新的名称，叫做对象（Object）。

在C语言中，我们会把重复使用或具有某项功能的代码封装成一个函数，将拥有相关功能的多个函数放在一个源文件，再提供一个对应的头文件，这就是一个模块。使用模块时，引入对应的头文件就可以。

而在C++中，多了一层封装，就是类（Class）。类由一组相关联的函数、变量组成，你可以将一个类或多个类放在一个源文件，使用时引入对应的类就可以。

  面向对象编程是针对开发中大规模的程序而提出来的，目的是提高软件开发的效率。不要把面向对象和面向过程对立起来，面向对象和面向过程不是矛盾的，而是各有用途、互为补充的。

本次上机实验通过构建一个简单有趣的王者荣耀英雄的类，对于类的构建，对于类中的默认构造函数，重载构造函数，复制构造函数，析构函数，成员函数有了初步的了解

1. 程序主体代码

***HERO.h***

#include<iostream>

using namespace std;

class HERO

{

public:

HERO(); //默认构造函数

HERO(int x1,int x2,int x3); //重载构造函数

HERO(const HERO & name); //复制构造函数

~HERO(); //析构函数

HERO &operator=(const HERO &rhs);

void showskill();

private:

int skill1;

int skill2;

int skill3;

};

***HERO.cpp***

#include"HERO.h"

HERO::HERO()

{

cout<<"This is a default constructor!"<<endl;

skill1 = 0;

skill2 = 0;

skill3 = 0;

}

HERO::HERO(int x1,int x2,int x3)

{

cout<<"This is a overloaded constructor!"<<endl;

skill1 = x1;

skill2 = x2;

skill3 = x3;

}

HERO::HERO(const HERO & name)

{

cout<<"This is a copy constructor!"<<endl;

skill1 = name.skill1;

skill2 = name.skill2;

skill3 = name.skill3;

}

HERO::~HERO()

{

cout<<"This is a destructor!"<<endl;

}

HERO & HERO::operator=(const HERO &rhs)

{

if( this == &rhs)

{

return \*this;

}

this->skill1 = rhs.skill1;

this->skill2 = rhs.skill2;

this->skill3 = rhs.skill3;

return \*this;

}

void HERO::showskill()

{

cout <<skill1 << endl;

cout <<skill2 << endl;

cout <<skill3 << endl;

}

***main.cpp***

#include<iostream>

#include"HERO.h"

using namespace std;

int main()

HERO libai;

cout<<"libai skills:"<<endl;

libai.showskill();

HERO sunwukong = HERO(10, 10, 10);

cout<<"libaiskills:"<<endl;

sunwukong.showskill();

HERO hanxin(sunwukong);

cout<<"hanxin skills:"<<endl;

hanxin.showskill();

libai = sunwukong;

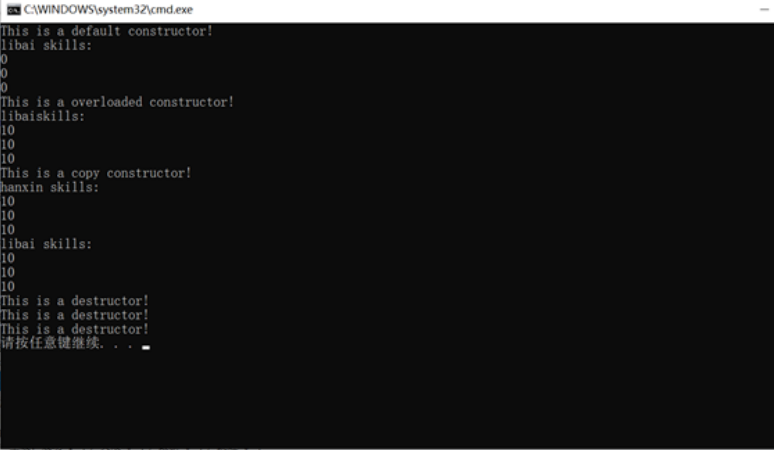
cout<<"libai skills:"<<endl;

libai.showskill();

return 0;

}

1. 运行结果



1. 实验小结

**（1）C语言与C++的区别：**

C语言中的 struct 只能包含变量，而 C++ 中的 class 除了可以包含变量，还可以包含函数。display() 是用来处理成员变量的函数，在C语言中，我们将它放在了 struct Student 外面，它和成员变量是分离的；而在 C++ 中，我们将它放在了 class Student 内部，使它和成员变量聚集在一起，看起来更像一个整体。  
 结构体和类都可以看做一种由用户自己定义的复杂数据类型，在C语言中可以通过结构体名来定义变量，在 C++ 中可以通过类名来定义变量。不同的是，通过结构体定义出来的变量还是叫变量，而通过类定义出来的变量有了新的名称，叫做对象（Object）。

**（2）定义类的对象：**

对象是根据类来创建的。声明类的对象，就像声明基本类型的变量一样。

**(3). 成员访问运算符 (.)：**类的对象的公共数据成员可以使用直接成员访问运算符 (.) 来访问,私有的成员和受保护的成员不能使用直接成员访问运算符 (.) 来直接访问。

**(4).** **构造函数和析构函数:** 二者的区别是：1.析构函数的类名不同，前面必须加上~

2.析构函数不返回任何值 3.析构函数没有参数，因此它无法被重载 4.撤销对象时，系统会自动调用析构函数。

**(5).** **作用域运算符：**想要在局部变量的作用域内使用同名的全局变量，可以在该变量前加上“::” ，“::”称为作用域运算符。

1. 作业3程序代码：

***J.h***

#include <iostream>

using namespace std;

class J

{

public:

J();

J( int bl,int ll,int at,int sd,int gs,int ap,int mk,int hj,int hl,int hx);

void showd();

void jiaxue();

void at();

~J();

int b1;

int b2;

int b3;

int b4;

int b5;

int b6;

int b7;

int b8;

int b9;

int b10;

};

***J.CPP***

cout<<"英雄魔抗:";cout<<b7<<endl;

cout<<"英雄物抗:";cout<<b8<<endl;

cout<<"英雄回蓝:";cout<< b9 <<endl;

cout<<"英雄回血:";cout<< b10 <<endl;

}

void J::jiaxue()

{

b1 =b1+10;

b2 =b2-10;

}

void J::at()

{

b1= b1-50;

}

J::~J ()

{

cout<<"关羽已经阵亡\n函数消亡"<<endl ;

};

***K.CPP***

#include <iostream>

#include "J.h"

using namespace std;

int main()

{

cout<<"开始游戏，您的英雄为杨玉环\n您正在我方泉水。"<<endl;

int xz;

cout<<"1.长恨歌："<<endl;

cout<<"2.杨玉环状态："<<endl;

cout<<"3.攻击关羽："<<endl;

cout<<"4.return 0："<<endl;

cin>>xz;

J guanyu(100,100,0,0,0,0,0,0,0,0);//关羽初始化

J yangyuhuan(100,100,100,100,100,100,100,100,1,1);//杨玉环初始化

while(xz!=4)

{

if(xz==3)

{ yangyuhuan.showd ();}

else

switch (xz)

{

case 1: {

yangyuhuan .jiaxue ();yangyuhuan.showd ();

if (yangyuhuan.b2 ==0)

{cout <<"法力值不足"<<endl;

}}

break ;

case 2: guanyu.at(); guanyu .showd ();

if (guanyu .b1==0)

{cout<<"关羽死亡"<<endl ;

guanyu.~J ();}break ;

default :cout<<"无效按键"<<endl;break;

}

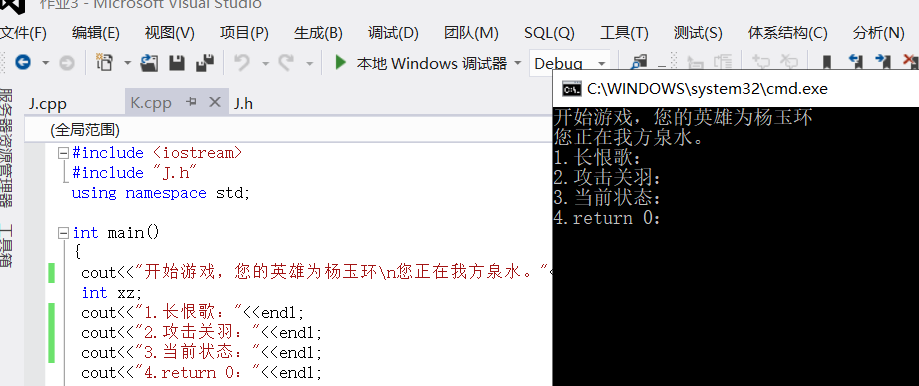
cin>>xz;

}

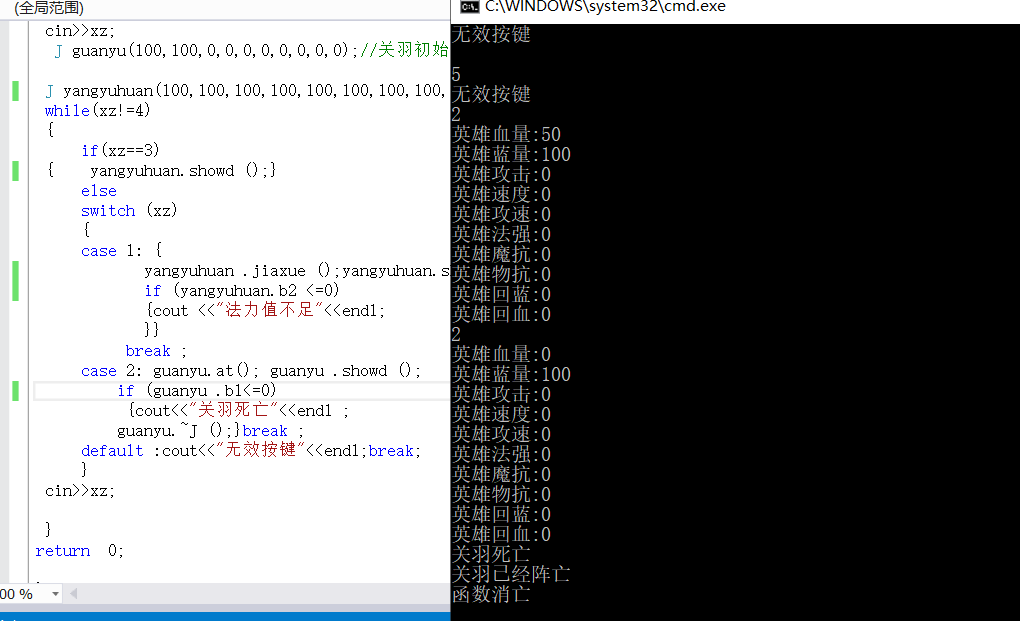
return 0;

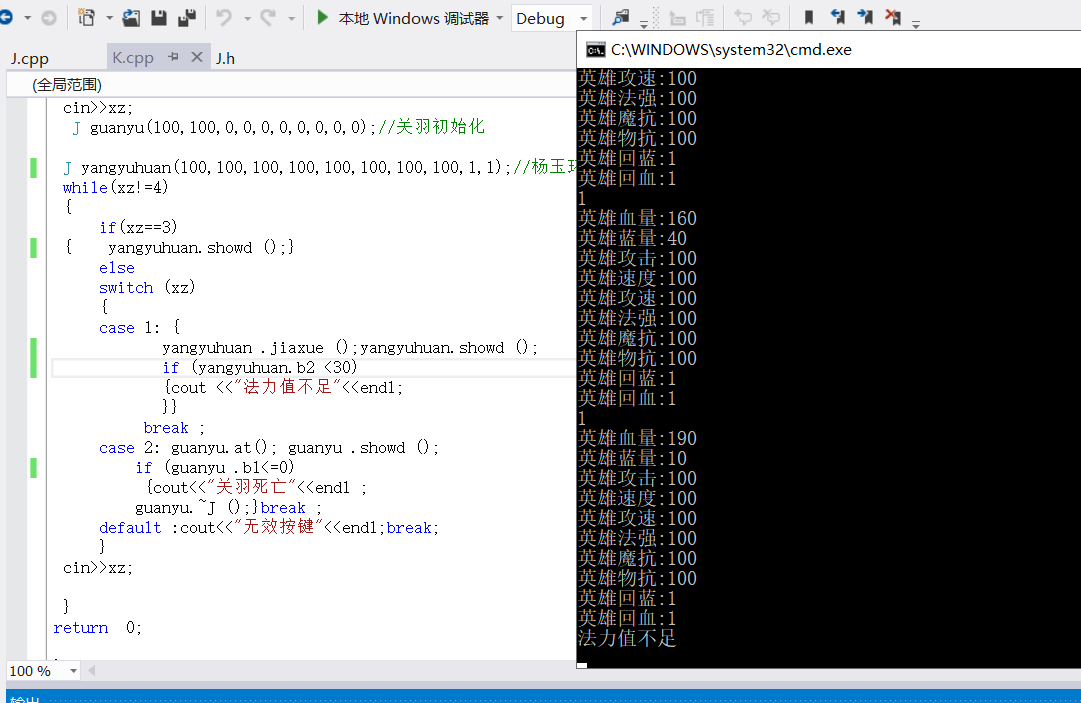
}

实验结果：

初始化界面：

除了1234，其他按键为无效按键，程序会给予提示，下面这段程序是我来攻击英雄关羽，当定义bl为英雄血量值，当关羽的boold小于等于0时，关羽会被击杀，程序会出现提示，然后关羽的数据消亡。



这里实现了杨玉环用大招恢复血量值，我设定杨玉环大招法力值消耗为30，所以当我法力值小于30时他会出现法力值不足。

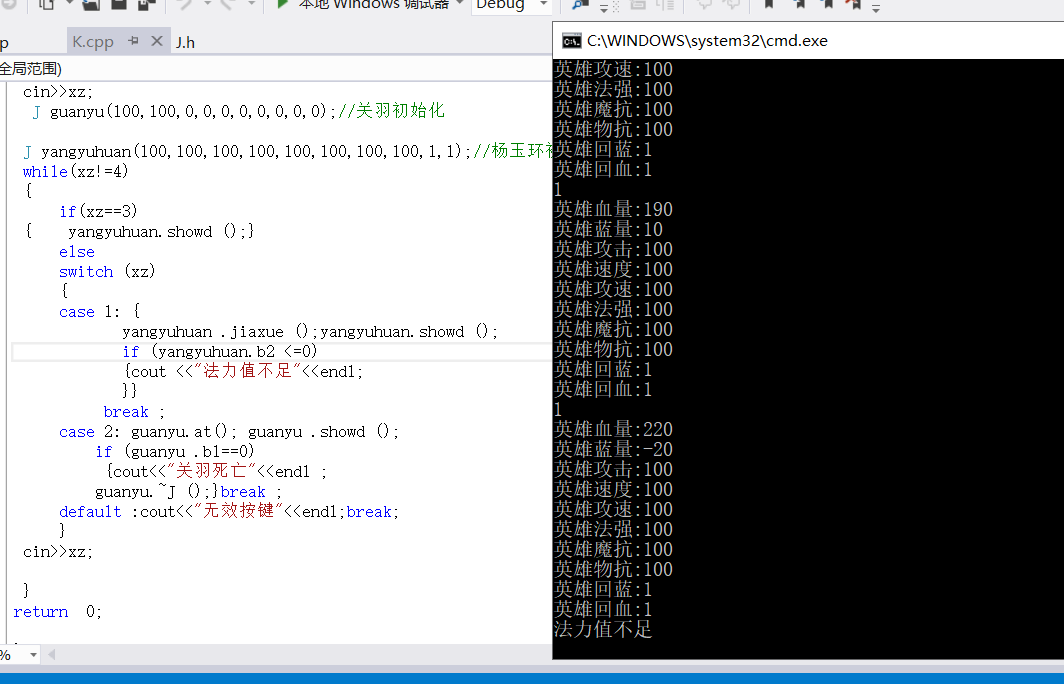
反思：

在本次作业中出现了不少麻烦与错误，在这里大体说几个吧

1.首先没有加break，他将不会跳出。

2.在击杀设计时，应计算溢出伤害，所以我取bl<=0时才报击杀

3.法力值消耗的时候，一开始，设计的是b2<=0时，报法力值不足，但是这样显然不对，该程序我只设计了大招的法力值消耗，所以当所剩法力值低于技能消耗时，应报“法力值不足”



4.将攻击目标对象关羽设计成析构函数，使其先创建关羽这个对象，当他bl（血量值）不足时报击杀，同时执行析构函数，将刚才创建的关羽对象消亡，节省了内存空间，虽然这是个小程序。