**实验报告三**

1. **程序设计思路**

析构函数是成员函数的一种，没有参数和返回值。析构函数在对象消亡时自动被调用。在本次实验中，我们在头文件中建立了一个名为“shuxing”的类，相当于建立了一个英雄的模板，这样以后无论建立何种英雄只要改变类中参数即可，不需要再重新编写每个英雄函数，在“shuxing”类下建立了显示英雄状态的“show”函数，完成击打动作的“gongji”函数和显示英雄被击败的析构函数“~gongji”。

1. **程序代码**
2. main.cpp

#include <iostream>

#include "gongji.h"

using namespace std;

int main()

{

cout<<"》 亲爱的召唤师：你好，我是你的助手妲己！！"<<endl;

cout<<"》 欢迎来到王者峡谷！！"<<endl;

cout<<"》 敌军还有五秒钟到达战场！！"<<endl;

cout<<"》 让我们一起战斗吧！！"<<endl;

cout<<"》 敌方英雄血量："<<endl;

cout<<"》 100"<<endl;

cout<<"》 敌方英雄魔抗:"<<endl;

cout<<"》 100"<<endl;

cout<<"》 敌方英雄物抗："<<endl;

cout<<"》 100"<<endl;

cout<<"》 英雄蓝量："<<endl;

cout<<"》 100"<<endl;

cout<<"》 英雄红量："<<endl;

cout<<"》 100"<<endl;

cout<<"》 让我们按“1”键释放一技能击溃敌人吧！！"<<endl;

int q;

cin>>q;

shuxing direnjie(100,100,100,100,100);

while(q>0)

{

switch (q)

{

case 1:if (direnjie.b1>0)

{

direnjie.gongji();

direnjie .show();

if (direnjie .b1<=0)

{

direnjie.~shuxing ();

}

};

break ;

default:cout<<"无效键"<<endl;break;

}

cin>>q;

}

return 0;

}

1. gongji.cpp

#include <iostream>

#include "gongji.h"

using namespace std;

shuxing::shuxing( int c1,int c2,int c3,int c4,int c5)

{

b1=c1;

b2=c2;

b3=c3;

b4=c4;

b5=c5;

}

1. gongji.h

#include <iostream>

using namespace std;

class shuxing

{

public:

shuxing();

shuxing( int a1,int a2,int a3,int a4,int a5);//血量：a1 魔抗：a2 物抗：a3 蓝耗：a4 红耗：a5

int b1; int b2;int b3;int b4;int b5;

void show()

{

cout<<"敌方英雄血量:";

cout<<b1 <<endl;

cout<<"敌方英雄魔抗:";

cout<< b2 <<endl;

cout<<"敌方英雄物抗:";

cout<< b3<<endl;

cout<<"英雄蓝量:";

cout<< b4<<endl;

cout<<"英雄红量:";

cout<< b5 <<endl;

}

void gongji()

{

b1 =b1-25;

b2 =b2-15;

b3 =b3-10;

b4 =b4-5;

b5 =b5-10;

};

~shuxing()

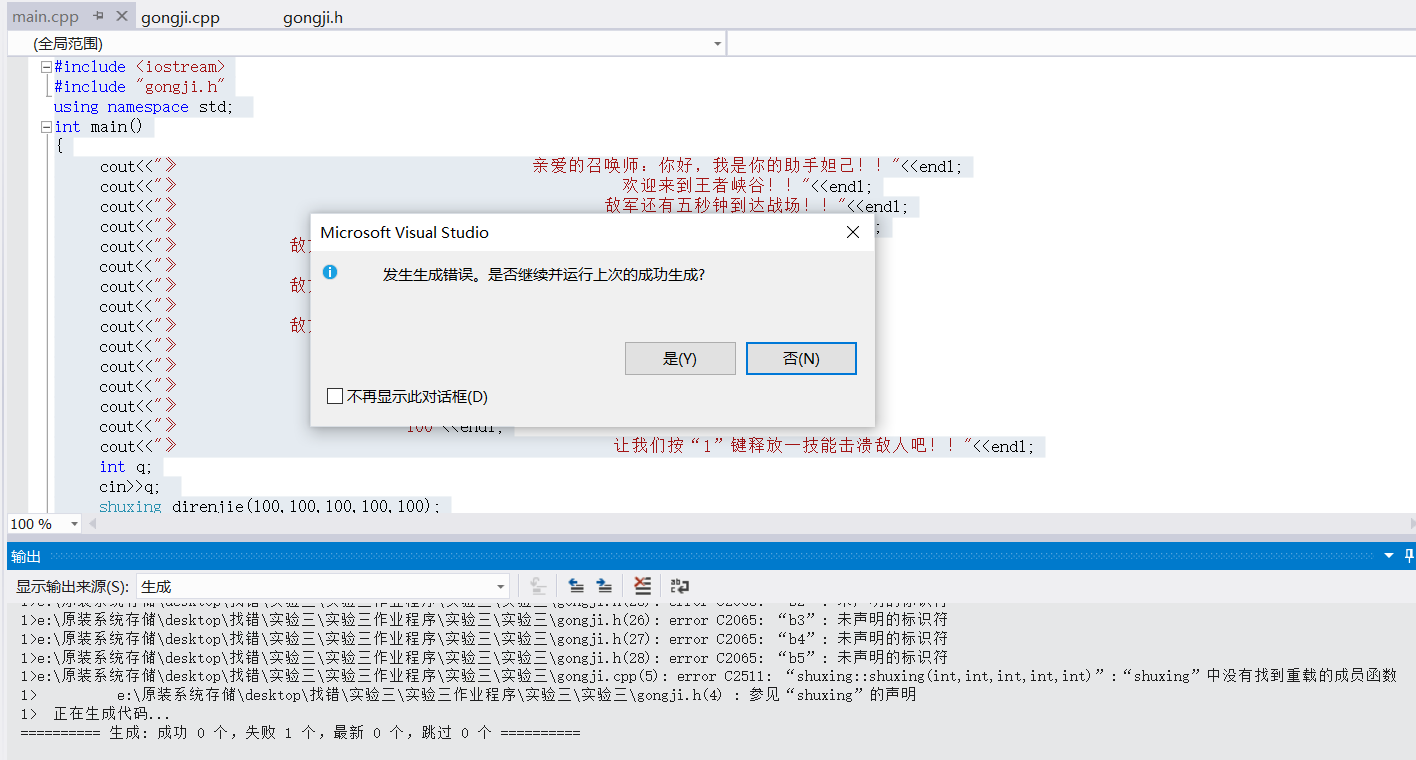
{

cout<<"》 ###敌方英雄被击败！！###"<<endl ;

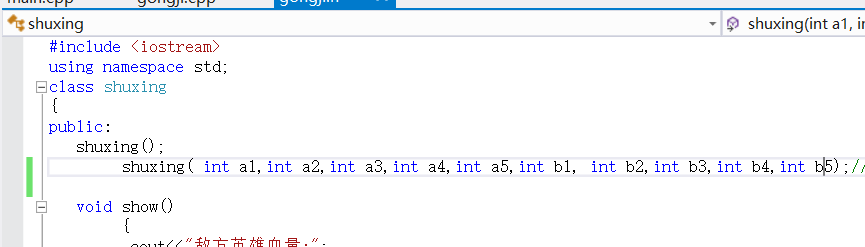
};

};

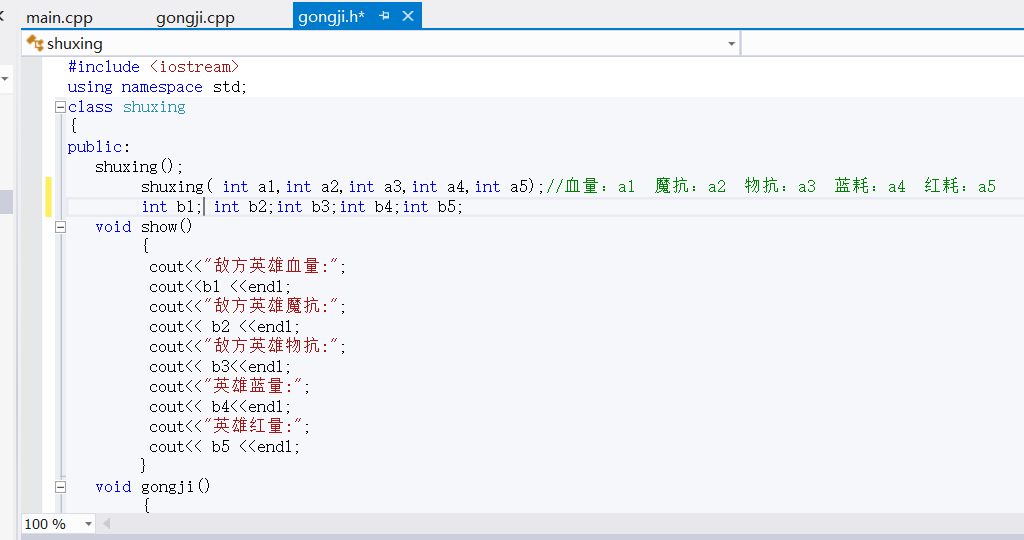
1. **程序错误及解决方法**
2. 程序错误：编译过程中发现程序无法编译，现象如下：



错误更改：通过系统提示发现b类参数未声明，通过检查发现将b定义在了构造函数中致使b无法被调用，我们将b放到构造函数外定义即可，错误如下：



改正程序如下：



1. **程序运行结果**

