#include<iostream.h>

class Grandfather{

public:

Grandfather(){

cout<<"grandfather\n"<<endl;

}

void Sleep(){

cout<<"grangfather is sleeping\n"<<endl;

}

~Grandfather(){

cout<<"destructing grandfather\n"<<endl;

}

};

class Father:public Grandfather{

public:

Father(){

cout<<"father\n"<<endl;

}

void WatchTV(){

cout<<"father is watchtv\n"<<endl;

}

~Father(){

cout<<"destructing father\n"<<endl;

}

};

class Son:public Father{

public:

Son(){

cout<<"son\n"<<endl;

}

void Play(){

cout<<"son is playing lol\n"<<endl;

}

~Son(){

cout<<"destructing son\n"<<endl;

}

};

int main(){

Son s;

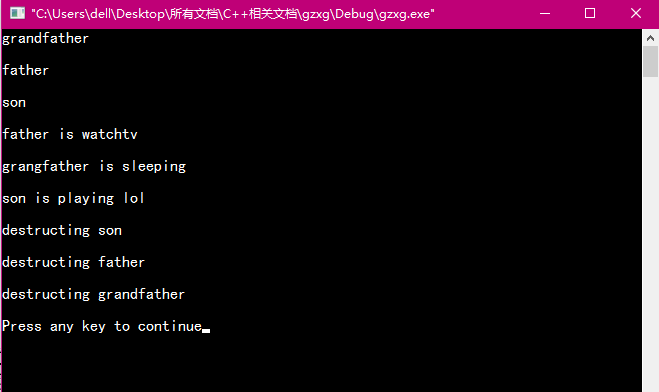
s.WatchTV();

s.Sleep();

s.Play();

return 0;

}



构造函数的执行顺序：先祖先（基类，调用顺序按照他们继承时说明的顺序），再客人（对象成员，调用顺序按照他们在类中说明的顺序），后自己（派生类本身）。

析构函数的执行顺序与构造函数正好严格相反。

上图中方的执行顺序则是用户定义的顺序。