实验报告

实验一

#include "stdafx.h"

#include <iostream>

using namespace std;

enum CPU\_Rank { P1 = 1, P2, P3, P4, P5, P6, P7 };

class CPU {

public:

CPU(CPU\_Rank r, float vol, int fre)

{

rank = r;

voltage = vol;

frequency = fre;

cout << "调用构造函数" << endl;

};

~CPU() { cout << "调用析构函数" << endl; };

CPU\_Rank getrank() const { return rank; };

float getvol() const { return voltage; };

int getfre() const { return frequency; };

void setrank(CPU\_Rank r) { rank = r; };

void setvol(float vol) { voltage = vol; };

void setfre(int fre) { frequency = fre; };

void run() { cout << "运行CPU" << endl; };

void stop() { cout << "停止CPU" << endl; };

private:

CPU\_Rank rank;

float voltage;

int frequency;

};

int main() {

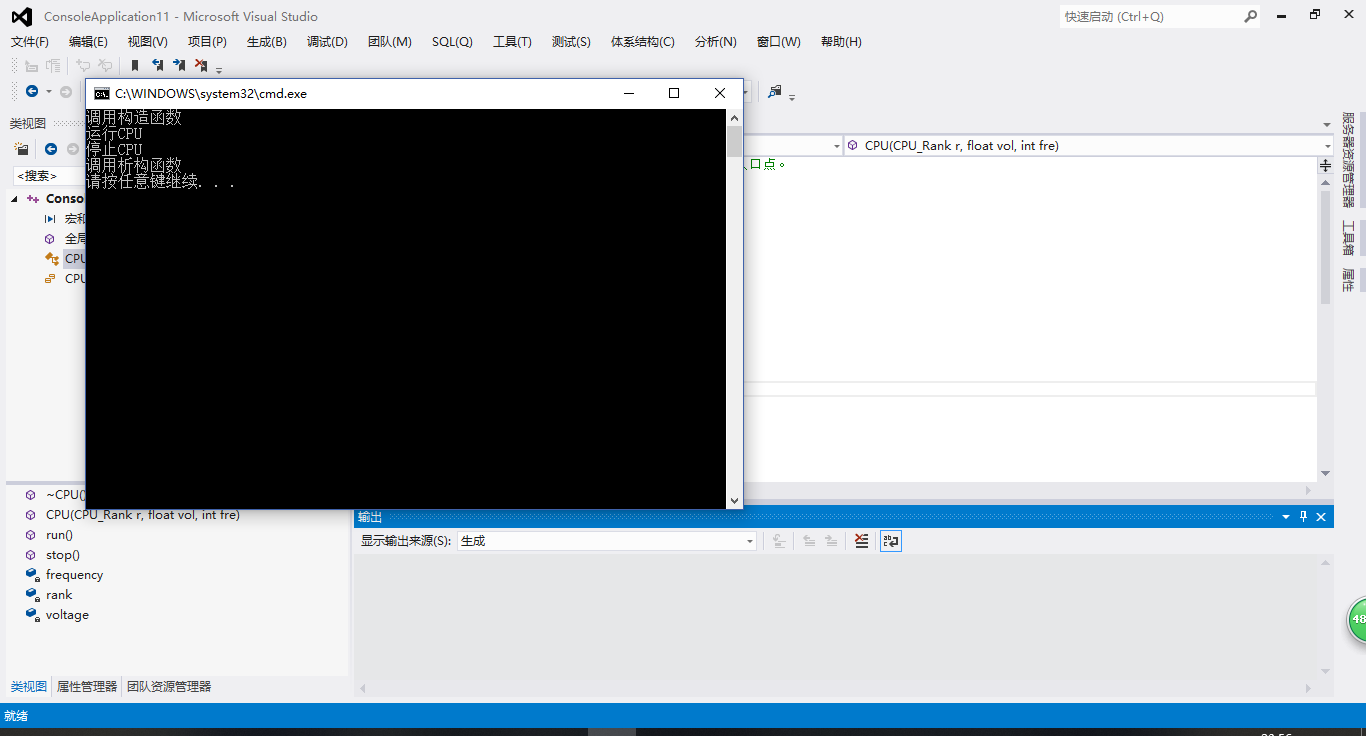
CPU cpu(P4, 2.8, 300);

cpu.run();

cpu.stop();

return 0;

}



先调用构造函数，在依次顺序运行完构造函数和类中两个共有成员函数后，调用析构函数以释放内存。

实验二

#include "stdafx.h"

#include <iostream>

using namespace std;

enum CPU\_Rank { P1 = 1, P2, P3, P4, P5, P6, P7 };

class CPU {

public:

CPU(CPU\_Rank r, float vol, int fre)

{

rank = r;

voltage = vol;

frequency = fre;

cout << "调用CPU构造函数" << endl;

};

~CPU() { cout << "调用CPU析构函数" << endl; };

CPU\_Rank getrank() const { return rank; };

float getvol() const { return voltage; };

int getfre() const { return frequency; };

void setrank(CPU\_Rank r) { rank = r; };

void setvol(float vol) { voltage = vol; };

void setfre(int fre) { frequency = fre; };

void run() { cout << "运行CPU" << endl; };

void stop() { cout << "停止CPU" << endl; };

private:

CPU\_Rank rank;

float voltage;

int frequency;

};

class RAM {

public:

RAM(int x) {

cout << "调用RAM构造函数" << endl;

};

~RAM() { cout << "调用RAM析构函数" << endl; };

private:

int x;

};

class CDROM {

public:

CDROM(int y) {

cout << "调用CDROM构造函数" << endl;

};

~CDROM() { cout << "调用CDROM析构函数" << endl; };

private:

int y;

};

class Computer {

public:

Computer(CPU\_Rank r,float v,int f,int x,char y):cpu(r,v,f),ram(x),cdrom(y)

{

cout << "调用Computer构造函数" << endl;

};

~Computer() { cout << "调用Computer的析构函数" << endl; };

void run() { cout << "运行Computer" << endl; };

void stop() { cout << "停止Computer" << endl; };

private:

CPU cpu;

RAM ram;

CDROM cdrom;

};

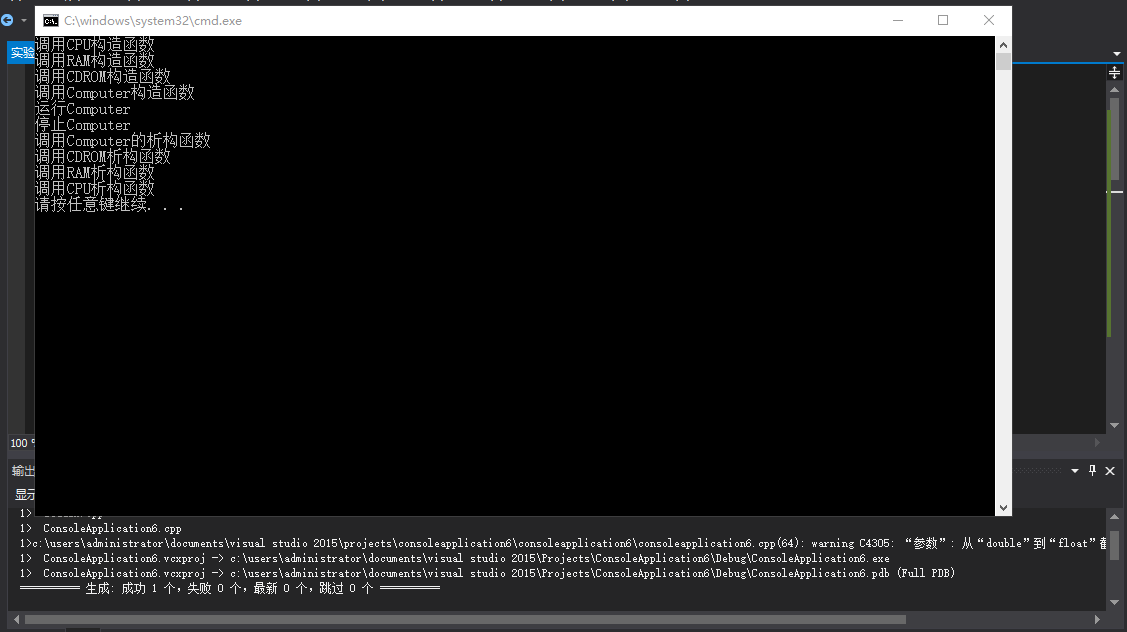
int main() {

Computer computer(P3, 2.8, 300, 4, 'G');

computer.run();

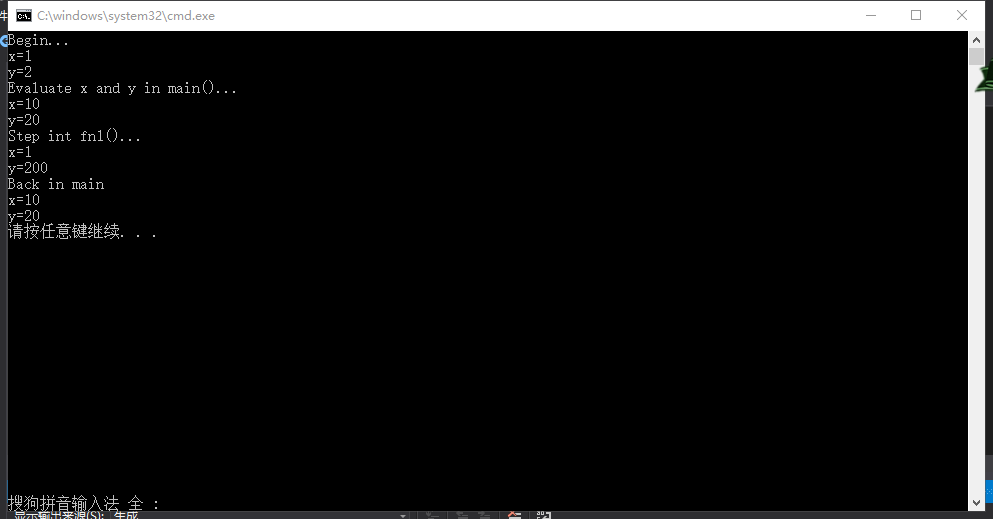
computer.stop();

}



实验三

（1）程序输出



（2）

//client.h

#pragma once

#ifndef CLIENT\_H

#define CLIENT\_H

class CLIENT {

public:

CLIENT();

~CLIENT();

static void changeServername(char sn);

static void showServername();

static void showClientnum();

private:

static char Servername;

static int Clientnum;

};

#endif

//client.cpp

#include "client.h"

#include <iostream>

using namespace std;

char CLIENT::Servername = 'A';

int CLIENT::Clientnum = 0;

CLIENT::CLIENT() { Clientnum++; };

CLIENT::~CLIENT() { Clientnum--; };

void CLIENT::changeServername(char sn) { Servername = sn; };

void CLIENT::showServername() { cout << "服务器名" << Servername << endl; };

void CLIENT::showClientnum() { cout << "客户总数" << Clientnum << endl; };

//lab5\_2.cpp

#include "client.h"

#include <iostream>

using namespace std;

int main() {

CLIENT::showServername();

CLIENT::showClientnum();

CLIENT::changeServername('Jin');

CLIENT x;

x.showServername();

x.showClientnum();

{

CLIENT y;

y.showServername();

y.showClientnum();

};

CLIENT::showServername;

CLIENT::showClientnum;

}

