实验一 CPU类

#include<iostream>

using namespace std;

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

class CPU {

public:

CPU(CPU\_Rank inputR, int inputF, double inputV) {

rank = (CPU\_Rank)inputR;

frequency = inputF;

voltage = inputV;

cout << "The constructor is called." << endl;

}

void run() {

cout << "CPU is running." << endl;

cout << rank <<endl<< frequency <<" MHZ"<<endl<< voltage<<" V"<<endl;

}

void stop() {

cout << "CPU has stopped." << endl;

}

~CPU() {

cout << "The deconstructor is called." << endl;

}

private:

CPU\_Rank rank;

int frequency;

double voltage;

};

int main() {

CPU cpu(P2, 1200, 100.0);

cpu.run();

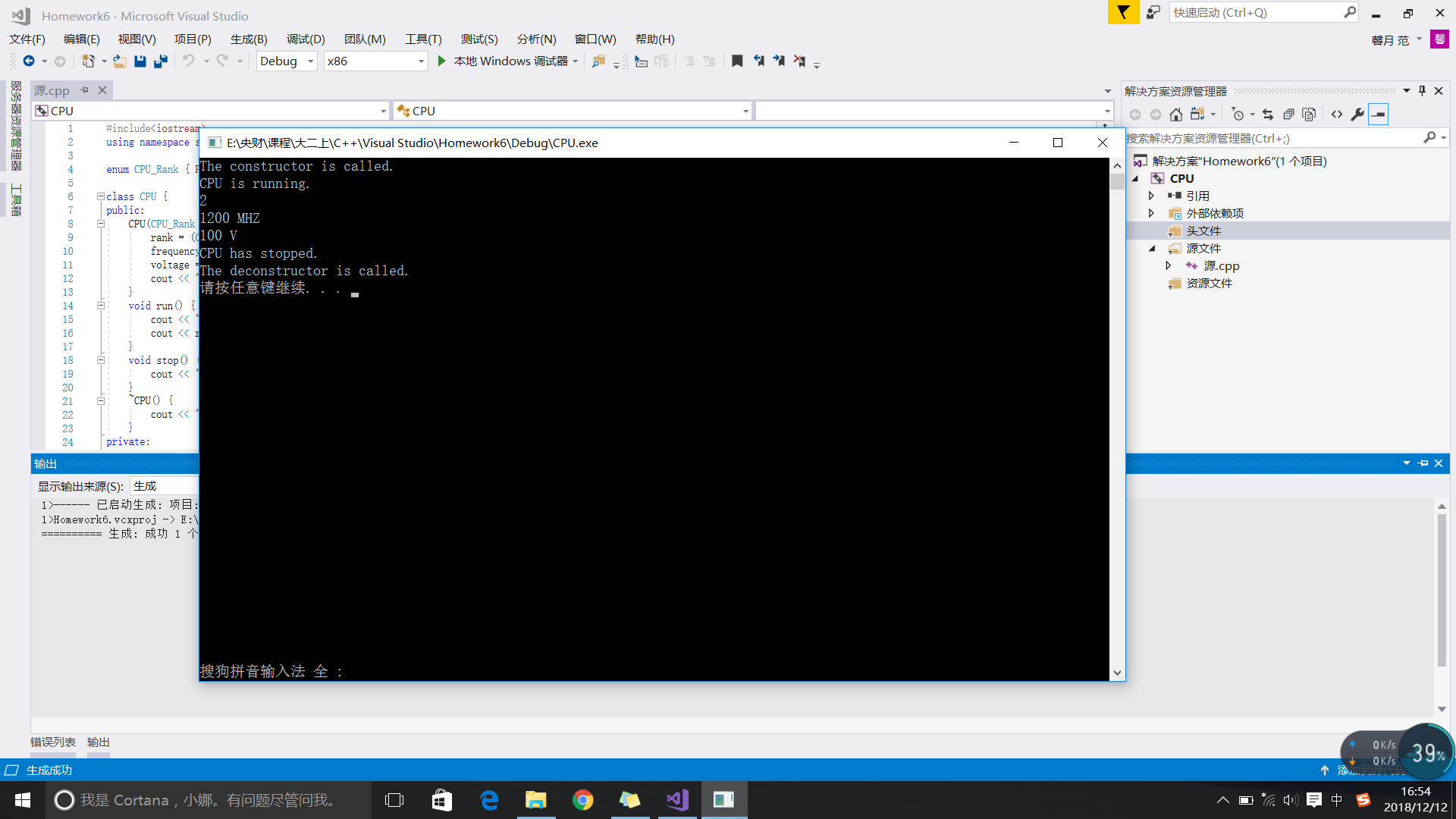
cpu.stop();

cpu.~CPU();

system("pause");

return 0;

}



实验二 Compute类

#include<iostream>

using namespace std;

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

class CPU {

public:

CPU(CPU\_Rank inputR = P1, int inputF=1500, double inputV=200.0) {

rank = (CPU\_Rank)inputR;

frequency = inputF;

voltage = inputV;

cout << "The constructor is called." << endl;

}

void run() {

cout << "CPU is running." << endl;

cout << rank << endl << frequency << " MHZ" << endl << voltage << " V" << endl;

}

void stop() {

cout << "CPU has stopped." << endl;

}

~CPU() {

cout << "The deconstructor is called." << endl;

}

private:

CPU\_Rank rank;

int frequency;

double voltage;

};

class RAM {

private:

double size;

public:

RAM(double inputSize=4) {

size = inputSize;

}

};

class CDROM {

private:

double size;

public:

CDROM(double inputSize=100) {

size = inputSize;

}

};

class Computer {

private:

CPU cpu;

RAM ram;

CDROM cdrom;

public:

Computer(CPU\_Rank inputR, int inputF, double inputV, double sizeRam, double sizeCdrom) :cpu(inputR, inputF, inputV), ram(sizeRam), cdrom(sizeCdrom) {

cout << "The constructoe is called" << endl;

}

void run() {

cout << "The computer is running" << endl;

cpu.run();

}

void stop() {

cout << "The computer has stopped" << endl;

cpu.stop();

}

};

int main() {

Computer com(P3, 1600, 300.0, 2.0, 400.0);

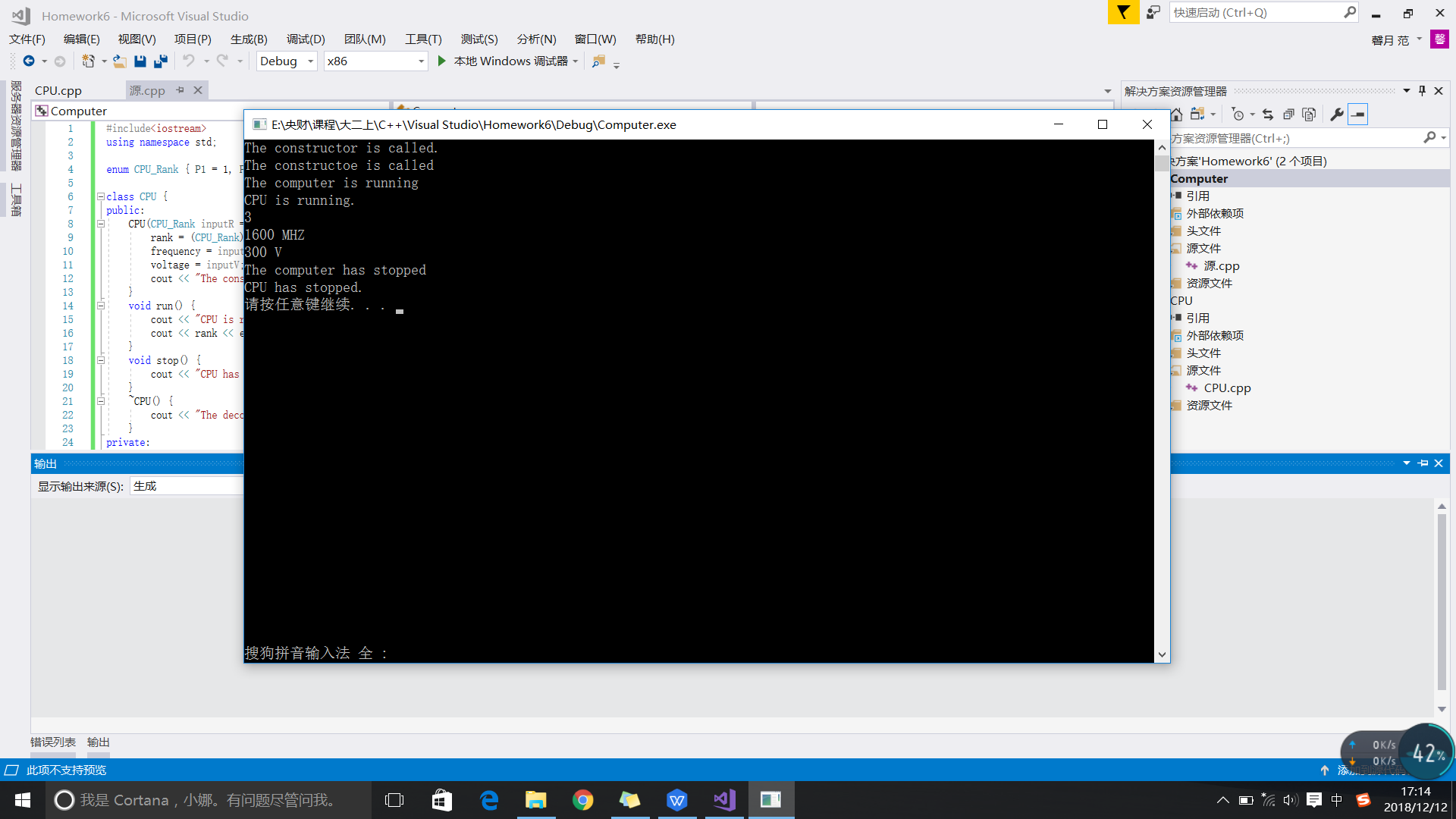
com.run();

com.stop();

system("pause");

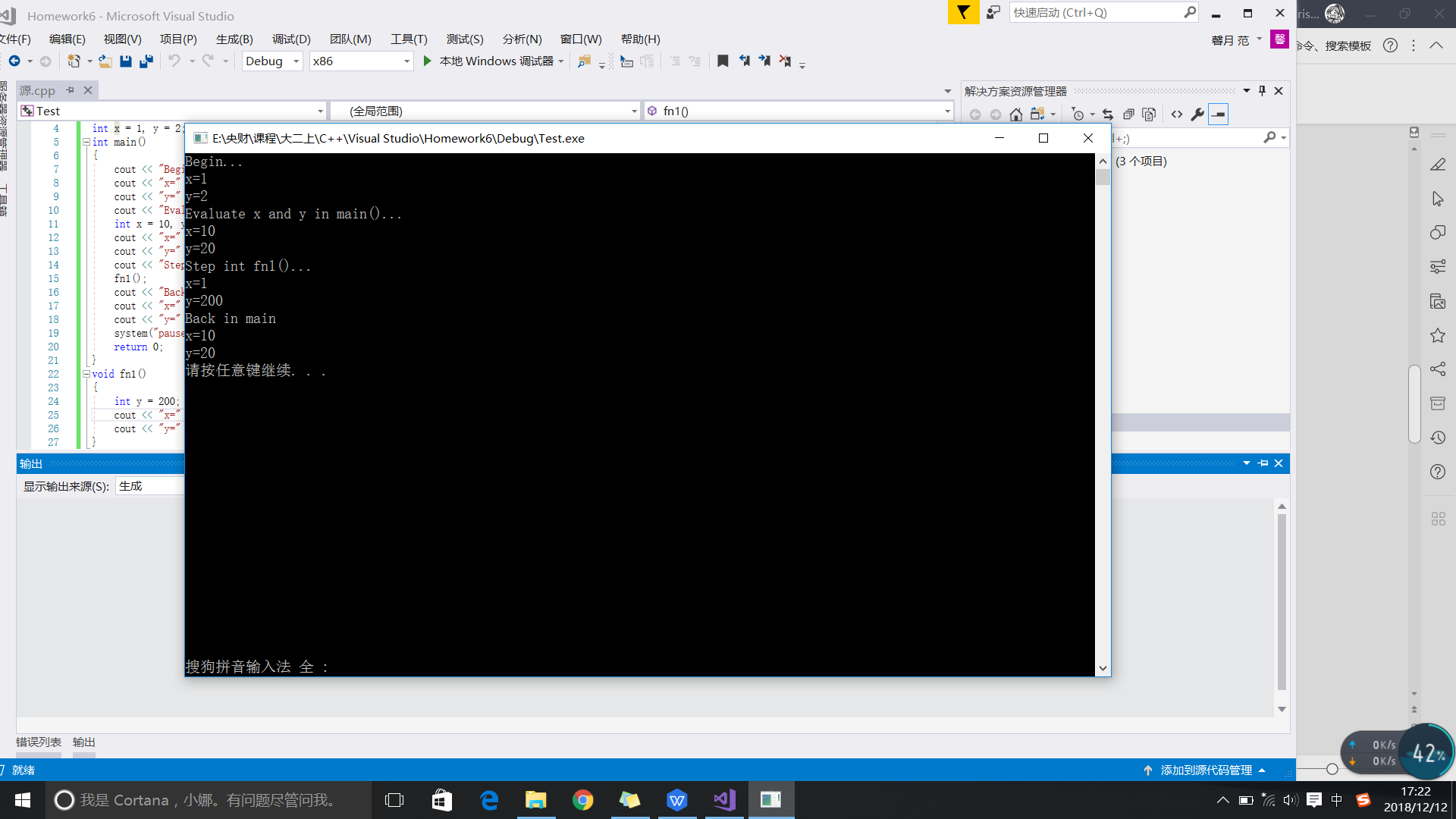
return 0;

}



实验三

1. Test



1. Lab

client.h

#pragma once

class CLIENT {

private:

int number = 0;

int key;

public:

static int totalNumber;

CLIENT(int inputNo, int inputKey);

int getNumber();

int getKey();

};

client.cpp

#include"client.h"

#include<iostream>

using namespace std;

CLIENT::CLIENT(int inputNo, int inputKey) {

number = inputNo;

key = inputKey;

CLIENT::totalNumber++;

cout << "An client is built." << endl;

}

int CLIENT::getNumber() {

return number;

}

int CLIENT::getKey() {

return key;

}

int CLIENT::totalNumber = 0;

lab5\_2.cpp

#include"client.h"

#include<iostream>

using namespace std;

int main() {

CLIENT client1(26, 123456);

cout << "The client No. " << client1.getNumber() << endl;

cout << "The client's key is: " << client1.getKey() << endl;

cout << "The total number of clients is " << CLIENT::totalNumber << endl;

CLIENT client2(30, 234567);

CLIENT client3(0,000000);

cout << "The total number of clients is " << CLIENT::totalNumber << endl;

system("pause");

return 0;

}

