实验一：

#include<iostream>

using namespace std;

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

class CPU

{

public:

CPU(CPU\_Rank r, int x, float y)

{ rank = r;

frequency = x;

voltage = y;

cout << "成功创建CPU！" << endl;

}

~CPU() { cout << "成功析构CPU！"; }

CPU\_Rank getrank() { return rank; }

int getfrequency() { return frequency; }

float fetvoltage() { return voltage; }

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

void setFrequency(int f) { frequency = f; }

void setVoltage(float v) { voltage = v; }

void run();

void stop();

private:

CPU\_Rank rank;

int frequency;

float voltage;

};

void CPU::run()

{

cout << "程序开始执行" << endl;

}

void CPU::stop()

{

cout << "程序结束" << endl;

}

int main()

{

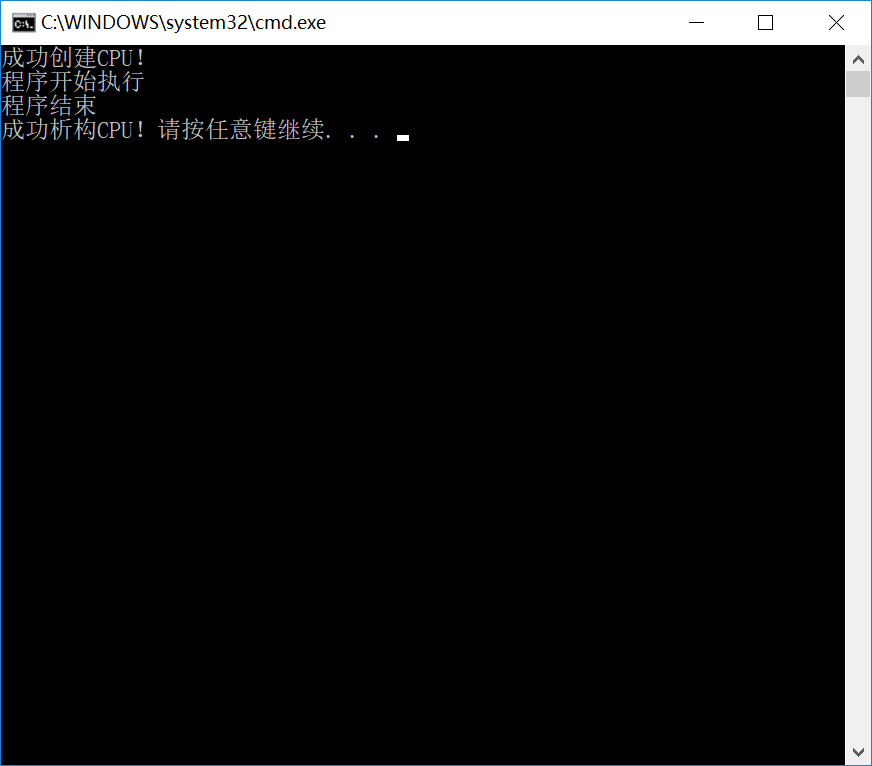
CPU C\_1(P1, 30, 220);

C\_1.run();

C\_1.stop();

return 0;

}

}

实验二：

#include <iostream>

using namespace std;

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

class CPU

{private:

CPU\_Rank rank;

int frequency;

float voltage;

public:

CPU(CPU\_Rank r, int f, float v)

{

rank = r;

frequency = f;

voltage = v;

cout << "构造了一个CPU!" << endl;

}

CPU()

{

cout << "构造了一个CPU!" << endl;

};

~CPU()

{

cout << "析构了一个CPU!" << endl;

}

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

int GetFrequency() const { return frequency; }

float GetVoltage() const { return voltage; }

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

void SetFrequency(int f) { frequency = f; }

void SetVoltage(float v) { voltage = v; }

void Run() { cout << "CPU开始运行!" << endl; }

void Stop() { cout << "CPU停止运行!" << endl; }

};

class RAM

{public:

RAM()

{

cout << "构造了一个RAM!" << endl;

}

~RAM()

{

cout << "析构了一个RAM!" << endl;

}

void Run() { cout << "RAM开始运行!" << endl; }

void Stop() { cout << "RAM停止运行!" << endl; }

};

class CDROM

{public:

CDROM() { cout << "构造了一个CDROM!" << endl; }

~CDROM()

{

cout << "析构了一个CDROM!" << endl;

}

void Run() { cout << "CDROM开始运行!" << endl; }

void Stop() { cout << "CDROM停止运行!" << endl; }

};

class COMPUTER

{

private:

CPU cpu;

RAM ram;

CDROM cdrom;

public:

COMPUTER()

{

cout << "构造了一个COMPUTER!" << endl;

}

~COMPUTER()

{

cout << "析构了一个COMPUTER!" << endl;

}

void Run()

{

cout << "COMPUTER开始运行!" << endl;

cpu.Run();

ram.Run();

}

void Stop()

{

ram.Stop();

cpu.Stop();

cout << "COMPUTER停止运行!" << endl;

}

};

int main()

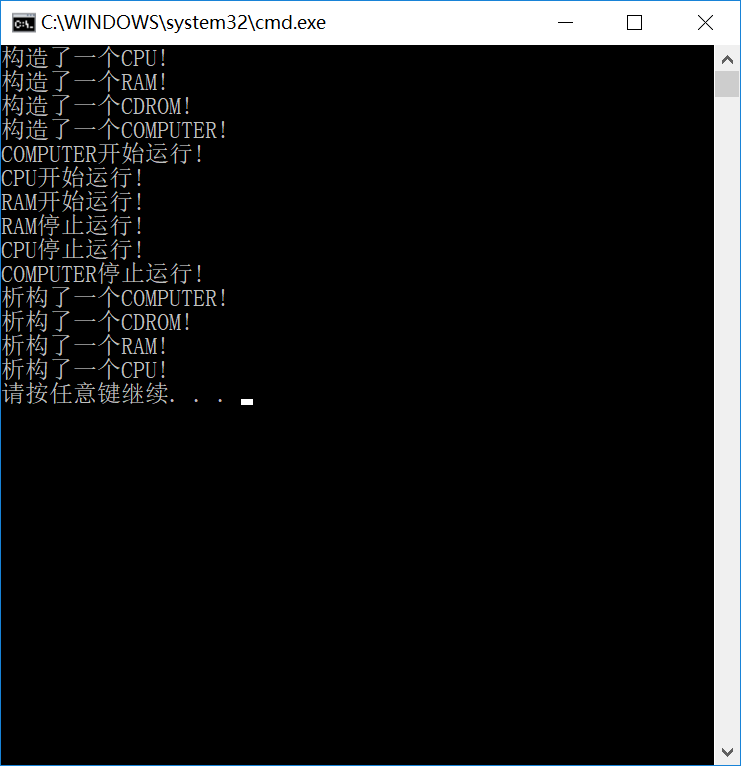
{

COMPUTER a;

a.Run();

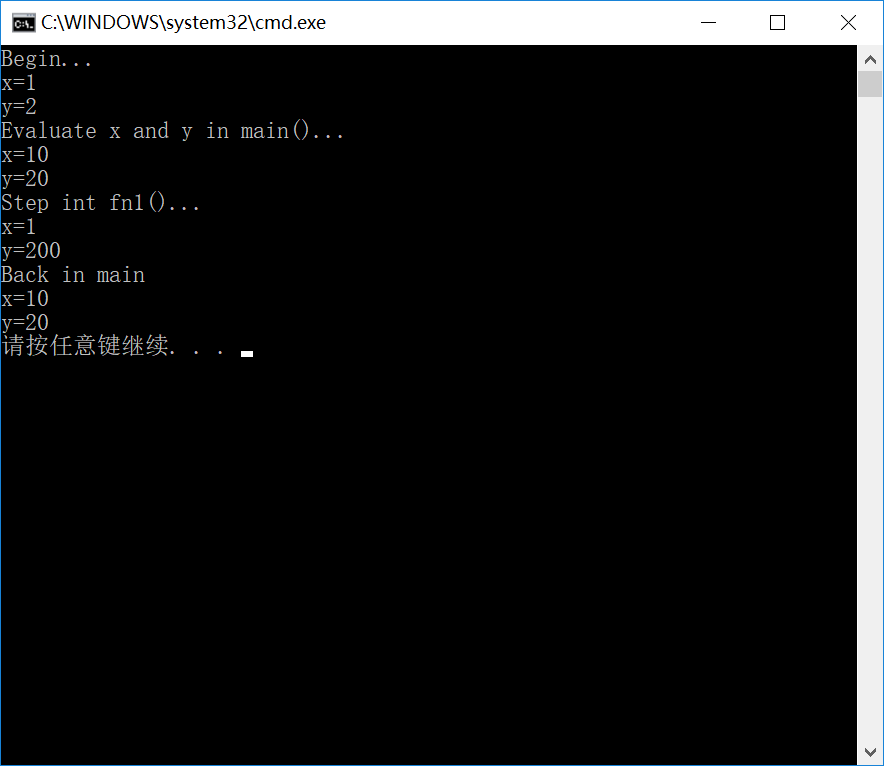
a.Stop();

}



实验三：

（1）



（2）

#ifndef CLIENT\_H

#define CLIENT\_H

class client {

public:

client();

~client();

static void ChangeServerName(char rw);

static void showServerName();

static void showClientNum();

private:

static char ServerName;

static int ClientNum;

};

#endif

#include<iostream>

#include "client.h"

using namespace std;

client::client() { ClientNum++; }

client::~client() { ClientNum--; }

void client::ChangeServerName(char rng) { ServerName =rng; }

void client::showServerName() { cout << "服务器名：" << ServerName << endl; }

void client::showClientNum() { cout << "客户数量：" << ClientNum<<endl; }

char client::ServerName = 'iboy';

int client::ClientNum = 0;

#include<iostream>

#include "client.h"

using namespace std;

void main() {

client::showServerName();

client::showClientNum();

client::ChangeServerName('Mlxg');

client EDG;

EDG.showServerName();

EDG.showClientNum();

{client IG;

IG.showServerName();

IG.showClientNum();

}

client::showServerName();

client::showClientNum();

}

