第四章实验说明

清华大学计算机科学与技术系
程序设计基础 郑莉

类与对象

- 实验目的
  - 掌握类的声明和使用。
  - 掌握类的声明和对象的声明。
  - 复习具有不同访问属性的成员的访问方式。
  - 观察构造函数和析构函数的执行过程。
  - 学习类的组合使用方法。
  - 使用VS2008以及Eclipse的debug调试功能观察程序流程，跟踪观察类的构造函数、析构函数、成员函数的执行顺序。

类与对象

- 实验要求

- 声明一个CPU类，包含等级（rank）、频率（frequency）、电压（voltage）等属性，有两个公有成员函数run、stop。其中，rank为枚举类型CPU_Rank，声明为enum CPU_Rank {P1=1,P2,P3,P4,P5,P6,P7}，frequency为单位是MHz的整型数，voltage为浮点型的电压值。观察构造函数和析构函数的调用顺序。
- 声明一个简单的Computer类，有数据成员芯片（cpu）、内存（ram）、光驱（cdrom）等等，有两个公有成员函数run、stop。cpu为CPU类的一个对象，ram为RAM类的一个对象，cdrom为CDROM类的一个对象，声明并实现这个类。
- （选做）设计一个用于人事管理的People（人员）类。考虑到通用性，这里只抽象出所有类型人员都具有的属性：number（编号）、sex（性别）、birthday（出生日期）、id（身份证号）等等。其中“出生日期”声明为一个“日期”类内嵌子对象。用成员函数实现对人员信息的录入和显示。要求包括：构造函数和析构函数、拷贝构造函数、内联成员函数、组合。

类与对象

- 实验内容（第一题）
- 声明枚举类型CPU_Rank，例如enum CPU_Rank {P1=1,P2,P3,P4,P5,P6,P7}，再声明CPU类，包含等级（rank）、频率（frequency）、电压（voltage）等私有数据成员，声明成员函数run、stop，用来输出提示信息，在构造函数和析构函数中也可以输出提示信息。在主程序中声明一个CPU的对象，调用其成员函数，观察类对象的构造与析构顺序，以及成员函数的调用。

```
#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 << "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; }
  };

  int main()
  {
            CPU a(P6,300,2.8);
            a.Run();
            a.Stop();
  }
```

类与对象

- 实验内容（第二题）
- 下面的程序定义了一个computer类，包含CPU类的对象、RAM类的对象和CDROM类的对象。主函数中定义了两个computer类的对象。主要是为了观察程序运行过程中构造函数和析构函数的调用过程。
  - 构造函数：对象被创建时自动调用。
  - 析构函数：对象生存期即将结束时被调用。调用完成后，对象消失，相应内存空间释放。


```cpp
#include<iostream>
using namespace std;

enum CPU_rank{p1=1,p2,p3,p4,p5,p6,p7};
class CPU
{
public://有五个公有内联成员函数
    CPU (CPU_rank r=p2,int f=2,float v=3):rank(r),frequency(f),voltage(v){
        cout<<"调用CPU构造函数"<<endl;
    }
    CPU(CPU &p):rank(p.rank),frequency(p.frequency),voltage(p.voltage){
        cout<<"调用CPU复制构造函数"<<endl;
    }
    ~CPU(){
        cout<<"调用CPU析构函数"<<endl;
    }
    CPU_rank getrank() const{//常成员函数 不改变对象的数据成员
        return rank;
    }
    int getfrequency() const{
        return frequency;
    }
private://三个私有数据成员
    CPU_rank rank;
    int frequency;
    float voltage;
};
```

```
class RAM
{
public:
    RAM(int r){
        cout<<"调用RAM构造函数"<<endl;
    }
    RAM(){
        ram=0;
        cout<<"调用RAM默认构造函数"<<endl;
    }
    RAM(RAM &p):ram(p.ram){
        cout<<"调用RAM复制构造函数"<<endl;
    }
    ~RAM(){
        cout<<"调用RAM析构函数"<<endl;
    }
private:
    int ram;
};
```

RAM类

构造函数与一般成员函数的不同：构造函数函数名与类名相同，没有返回值。

```
class CDROM
{
public:
CDROM(int cd){
    cdrom=cd;
    cout<<"调用CDROM构造函数"<<endl;
}
CDROM(){
    cout<<"调用CDROM默认构造函数"<<endl;
}
CDROM(CDROM &p):cdrom(p.cdrom){
    cout<<"调用CDROM复制构造函数"<<endl;
}
~CDROM(){
    cout<<"调用CDROM析构函数"<<endl;
}
private:
    int cdrom;
};
```

CDROM类

析构函数的函数名由类名前面加“~”构成，没有返回值，不接收任何参数。

```
69  class computer
70  {
71  public://把函数定义全写在了类外,这里更简洁一些
72      computer(CPU c,RAM r,CDROM cd);//构造函数
73      computer();//默认构造函数
74      ~computer();//析构函数函数
75      computer(computer &p);//复制构造函数
76      int show_cpu_fre();//函数功能：显示computer里的cpu对象的frequency成员
77  private:
78      CPU cpu;RAM ram;CDROM cdrom;
79  };
80
81  computer::computer():cpu(p3,100,2.3),ram(20),cdrom(30){
82      cout<<"调用computer默认构造函数"<<endl;
83  }
84  computer::computer(CPU c,RAM r,CDROM cd):cpu(c),ram(r),cdrom(cd){
85      cout<<"调用computer构造函数"<<endl;
86  }
87  computer::computer(computer &p): cpu(p.cpu),ram(p.ram),cdrom(p.cdrom){
88      cout<<"调用了复制构造函数"<<endl;
89  }
90  computer::~computer(){
91      cout<<"调用computer析构函数"<<endl;
92  }
93  int computer::show_cpu_fre(){
94      cout<<"cpu frequency is "<<cpu.getfrequency()<<endl;
95      return cpu.getfrequency();
96  }
```

computer类

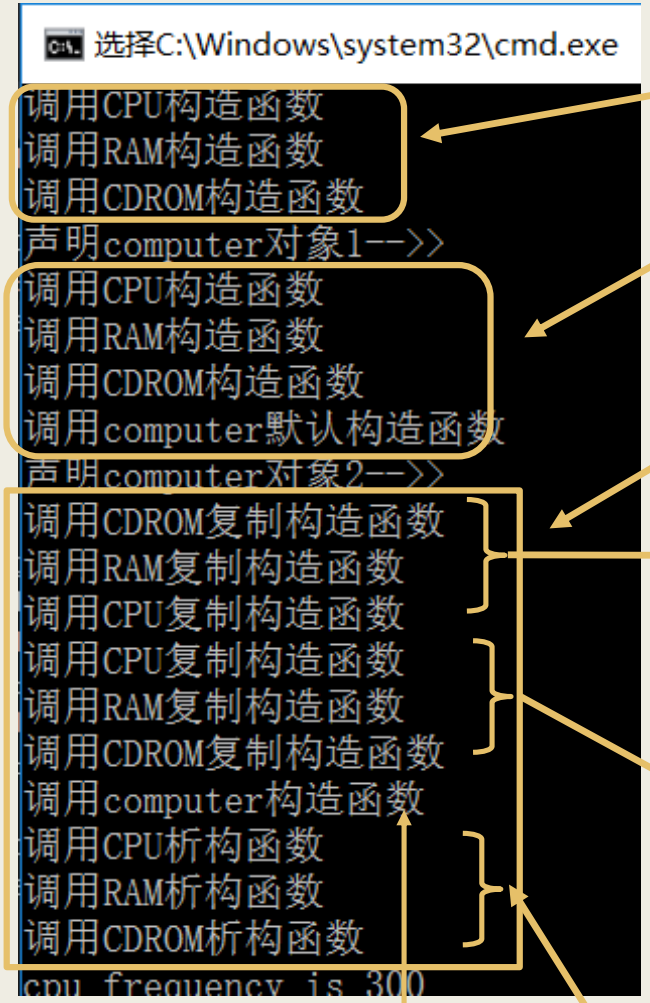
```
98  int main()
99  {
100     CPU CPUobject(p6,300,2.8);
101     RAM RAMobject(1);
102     CDROM CDROMobject(2);
103     cout<<"声明computer对象1-->>"<<endl;
104     computer computer_1;
105     cout<<"声明computer对象2-->>"<<endl;
106     computer computer_2(CPUobject,RAMobject,CDROMobject);
107
108     computer_2.show_cpu_fre();
109     cout<<"结束！"<<endl;
110     return 0;
111 }
```

main函数

复制构造函数调用的三种情形：

1. 当用类的一个对象去初始化该类的另一个对象时
2. 函数的形参是类的对象，调用函数进行形实结合时
3. 函数的返回值是类的对象，函数执行完返回时

下面分析结果》》



Line 100 101 102 分别创建各自类型的对象

调用computer类默认构造函数初始化对象computer_1
1. 调用内嵌对象 cpu ram cdrom 的构造函数初始化这些对象（调用顺序按照其在computer类中的定义顺序，cpu 先定义，所以先初始化cpu）
2. 执行computer类的构造函数的函数体

调用computer类带形参的构造函数初始化computer_2

See Line 84. 形实结合，调用复制构造函数初始化参数CPU类对象c RAM类对象r CDROM类对象cd.（注意这三个对象的作用域只在这个函数内，动态生存期）从右往左进行形实结合值传递（cd在最右边，先调用CDROM类的复制构造函数）

Line 84 初始化列表中，调用复制构造函数利用 c r cd 三个对象分别初始化computer类里的对象 cpu ram cdrom。（调用顺序按照其在computer类中的定义顺序）

执行computer类的构造函数函数体

c r cd 三个对象的生存期结束，要调用其析构函数。调用顺序为先创建的对象，后析构。

运行到main函数右大括号处，computer_2,computer_1,CDROMobject,RAMobject,CPUobject对象的生存期即将结束，调用析构函数。先创建的，后析构。

computer_2

computer_1

CDROMobject

RAMobject

CPUobject

组合类构造函数类外定义的一般形式：

类名::类名(形参表):内嵌对象1(), 内嵌对象2(), …

{语句;}

Thanks!