

类与对象（二）——构造函数、析构函数及对象的应用

张林鹏_2021032449

一、实验目的

1. 理解构造函数、析构函数的意义及作用, 掌握构造函数、析构函数的定义及调用时间, 熟悉构造函数的种类.
2. 理解this指针及使用方法, 熟悉对象数组、对象指针、对象引用的定义及使用方法, 熟悉对象作为函数参数的使用方法.
3. 熟悉类与对象的应用及编程.

二、实验内容

程序1: exp_201.cpp

1. 程序的运行输出结果为:

```
constructing!  
destructing!
```

2. 该输出结果说明构造函数 `Myclass()` 是在 构造函数时 执行的,

而析构函数 `~Myclass()` 是在 对象销毁时 执行的.

3. 将 `main()` 函数中的 `Myclass ob` 改为: `Myclass ob[2]` 后, 运行程序的输出结果为:

```
constructing!  
constructing!  
destructing!  
destructing!
```

4. 将 `main()` 中的 `Myclass ob[2]` 改为 `Myclass *ob; ob = new Myclass[2]` 后, 运行程序的输出结果为:

```
constructing!
```

```
constructing!
```

5. 在4. 的基础上, 在程序的末尾加入 `delete []ob` 后运行, 程序的输出结果为:

```
destructing!  
destructing!
```

6. 比较3 - 5的输出结果, 说明:

使用new创建的对象需要手动删除,否则会导致内存泄漏

程序2: exp_202.cpp

7. 运行该程序输出的结果为:

8. 程序中的成员函数 `A(A &ob)` 成为 拷贝构造函数.

该函数的执行时间时在执行 创建对象时 被调用的.

9. 将 `main()` 中的 `A ob2(ob1);` 改为 `A ob2=ob1;` , 重新运行程序, 观察输出结果, 说明拷贝构造函数也可以在 赋值操作 时调用.

10. 将 `main()` 函数中加注释的语句去掉前面的 `//` , 重新运行程序, 观察输出结果, 说明执行 `ob = ob1;` 时 会 调用拷贝构造函数, 原因是 `ob = ob1;` 只是对对象的 赋值操作.

程序3: exp_203.cpp

1. 你分析的输出结果是:

```
程序的实际输出结果是:  
调用func1:  
拷贝构造函数被调用!  
func1: a=10 b=10  
析构函数被调用!  
调用func2:  
func2: a=10 b=10  
调用func3:  
func3: a=10 b=10  
main: a=10 b=10  
析构函数被调用!
```

2. 程序实际输出结果是:

```
程序的实际输出结果是:
```

```
调用func1:
拷贝构造函数被调用!
func1: a=10 b=10
析构函数被调用!
调用func2:
func2: a=10 b=10
调用func3:
func3: a=10 b=10
main: a=10 b=10
析构函数被调用!
```

程序4: exp_204

11. 程序中:

- i. 处应为: `person;`
- ii. 处应为: `name = new char[strlen(pn) + 1];`
- iii. 处应为: `if(name!=NULL) strcpy(name,pn);`
- iv. 处应为: `new char[strlen(ob.name)+1];`
- v. 处应为: `if(name!=NULL) strcpy(name,ob.name);`

程序5: exp_205

12. 完善类的定义, 程序中:

- i. 处的定义应为: `for(i=0;i<m;i++) {sc[i]=x[i];sum+=sc[i];}`
- ii. 处的定义应为: `for(i=0;i<m;i++) {sc[i]=x[i];sum+=sc[i];}`

13. 完善 `main()` 函数, 程序中:

- i. 处的定义应为: `new score[n];`
- ii. 处的定义应为: `p,n,m;`
- iii. 处的定义应为: `delete []p;`
- iv. 处的定义应为: `p[i].set_score(x,m);`
- v. 处的定义应为: `p[i].print_score();`
- vi. 处的定义应为: `aver.set_score(s,m);`
- vii. 处的定义应为: `{k=j;a=p[j].get_aver();}`
- viii. 处的定义应为: `{t=p[i];p[i]=p[k];p[k]=t;}`

程序设计实验

1. result.h:

```

#ifndef RESULT_H
#define RESULT_H

#include <bits/stdc++.h>
#define UMPIRE 5 // 裁判数量
using namespace std;

class result
{
private:
    struct player
    {
        int number; //运动员编号
        string name; //运动员姓名
        vector<double> score; //运动员成绩
        double finalResult; //运动员最终成绩

        player() : number(0), name(""), finalResult(0) {}
    };

    int playerNumber; //运动员数量
    int judgeNumber = UMPIRE; //裁判数量
    vector<player> players; //运动员信息

public:
    result(): playerNumber(0) {}
    result(int p, int j) : playerNumber(p), judgeNumber(j) {}
    void inputPlayer();
    void inputScore();
    void calculateResult();
    void outputResult();
    void outputRank();
};

void result::inputPlayer()
{
    player temp;
    cout << "请输入运动员数量: " << endl;
    cin >> playerNumber;
    for (int i = 0; i < playerNumber; i++)
    {
        cout << "请输入第" << i + 1 << "个运动员的编号、姓名: ";
        cin >> temp.number >> temp.name;
        players.push_back(temp);
    }
    cout << endl;
}

void result::inputScore()
{
    for (int i = 0; i < playerNumber; i++)
    {

```

```

        cout << "请输入第" << i + 1 << "个运动员的成绩: " << endl;
        for (int j = 0; j < judgeNumber; j++)
        {
            double temp;
            cout << "第" << j + 1 << "个裁判的成绩: ";
            cin >> temp;
            players[i].score.push_back(temp);
        }
    }
    cout << endl;
}

void result::calculateResult()
{
    for (int i = 0; i < playerNumber; i++)
    {
        double sum = 0;
        sort(players[i].score.begin(), players[i].score.end());
        for (int j = 1; j < judgeNumber - 1; j++)
        {
            sum += players[i].score[j];
        }
        players[i].finalResult = sum / (judgeNumber - 2);
    }
}

void result::outputResult()
{
    cout << "运动员编号\t运动员姓名\t最终成绩" << endl;
    for (int i = 0; i < playerNumber; i++)
    {
        cout << players[i].number << "\t\t" << players[i].name << "\t\t" <<
players[i].finalResult << endl;
    }
}

void result::outputRank()
{
    sort(players.begin(), players.end(), [](player a, player b) { return
a.finalResult > b.finalResult; });
    cout << "运动员编号\t运动员姓名\t最终成绩" << endl;
    for (int i = 0; i < playerNumber; i++)
    {
        cout << players[i].number << "\t\t" << players[i].name << "\t\t" <<
players[i].finalResult << endl;
    }
}

#endif

```

2. exp_206.cpp:

```
#include <bits/stdc++.h>
#include "result.h"
using namespace std;

int main()
{
    result r1;
    r1.inputPlayer();
    r1.inputScore();
    r1.calculateResult();
    r1.outputResult();
    r1.outputRank();
}
```