|  |  |
| --- | --- |
| **组号:** | **11** |

****

信息科学与工程学院课程实验报告

《面向对象程序设计》

|  |  |
| --- | --- |
| 姓名： | 陈冰 |
| 学号： | 201711010250 |
| 班级： | 计工本1702 |
| 教师： | 张庆科 |
| 时间： | 2018/11/18 |

**面向对象程序设计实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 陈冰 | 班级 | 计工本1702 | 学号 | 201711010250 | 组号 | 11 |
| 时间 | 2018/10/21 | 地点 | E315 | 周次 |  | 页码 |  |
| 源码 | □ 无源码 ☑ 文档源码 □ 托管源码 | | | | | | |
| 报  告  内  容 | **实验报告要求**：请围绕实验目的、实验内容、实验过程及步骤(可添加文字、矢量图)、实验结论与分析进行撰写。凡涉及源代码内容可给出完整源码并附上源码Github托管网址（请务必按照条目书写）。   1. **实验目的**   1. 编程调试教材第4章"数据共享"案例后面的程序思考题:P103(思考与练习),P105(思考与练习). 观察分析结果, 并给出原因；编程调试教材第4章"数据保护"案例后面的程序思考题:P107(思考与练习),P109(思考与练习),P110(思考与练习). 观察分析结果, 并给出原因.  2. 调试运行教材P111-"学生信息管理系统"项目, 理解静态成员数据,静态成员函数 || 常数据,常函数,常对象的本质.  **2.实验问题及解决方式**  103页：（1）能运行，total默认值为0   1. 不能运行，没有对total初始化 2. 不能运行，需要定义一个静态成员函数   105页：（1）能运行，delete析构了new所分配的内存空间，但\*  p在栈上，指向student   1. 能运行，可以直接用类名进行访问   107页：（1）不能运行，PI只能通过初始化列表初始化  （2）同（1）  109页：（1）不能运行。常成员函数不能修改本类的数据成员，普通成员函数可以  （2）不能运行，常成员函数不能调用普通成员函数  （3）能运行，普通成员函数能调用常成员函数  110页：（1）能运行，调用成员函数的输出函数  （2）不能运行，没有常成员函数的输出函数  **3.实验源码及源码地址**  **example4\_01**  **8]LRCQ[8SKPF%{0YD49[I~I**  1 #include <iostream>  2 #include <cstdlib>  3 #include <windows.h>  4 #pragma warning(disable:4996)  5 **using namespace** std;  6  7 **class** Student  8 {  9 **private**:  10 **int** num;  11 **char** name[20];  12 **public**:  13 **static int** total;  14 **Student**() { total++; }  15 ~**Student**() { total--; }  16 **Student**(**int** n, **const char** \*p = "Wang");  17 **char** **GetName**();  18 **int** **Getnum**();  19 };  20 **int** **Student**::total;  21  22 **Student**::**Student**(**int** n, **const char** \*p)  23 {  24 num = n;  25 **strcpy**(name, p);  26 total++;  27 }  28  29 **char** **Student**::**GetName**()  30 {  31 **char** \*p1;  32 p1 = name;  33 **return** \*p1;  34 }  35  36 **int** **Student**::**Getnum**()  37 {  38 **return** num;  39 }  40  41  42  43 */\*----------------------------------\**  44  *Main Function*  45 *\*-----------------------------------\*/*  46  47 **int** **main**()  48 {  49 cout << "The number of all students:" << **Student**::total << endl;  50 Student \*p = **new** **Student**(13);  51 cout << "The number of all students:" << **Student**::total << endl;  52 cout << "The number of all students:" << p->total << endl;  53 **delete** p;  54 cout << "The number of all students:" << **Student**::total << endl;  55 Student s[2];  56 cout << "The number of all students:" << s[0].total << endl;  57 cout << "The number of all students:" << s[2].total << endl;  58  59 **system**("pause");  60 **return** 0;  61 }  **example4\_02**    1 #include <iostream>  2 #include <cstdlib>  3 #include <windows.h>  4 #pragma warning(disable:4996)  5 **using namespace** std;  6  7 **class** Student  8 {  9 **private**:  10 **int** num;  11 **char** name[20];  12 **static int** total;  13 **public**:  14  15 **Student**() { total++; }  16 ~**Student**() { total--; }  17 **Student**(**int** n, **const char** \*p = "Wang");  18 **char** **GetName**();  19 **int** **Getnum**();  20 **static void** **Print**();  21 };  22 **int** **Student**::total ;  23  24 **Student**::**Student**(**int** n, **const char** \*p)  25 {  26 num = n;  27 **strcpy**(name, p);  28 total++;  29 }  30  31 **char** **Student**::**GetName**()  32 {  33 **char** \*p1;  34 p1 = name;  35 **return** \*p1;  36 }  37  38 **int** **Student**::**Getnum**()  39 {  40 **return** num;  41 }  42  43 **void** **Student**::**Print**()  44 {  45 cout << "The number of all students:" << total <<endl;  46 }  47  48  49  50 */\*----------------------------------\**  51  *Main Function*  52 *\*-----------------------------------\*/*  53  54 **int** **main**()  55 {  56 **Student**::**Print**();  57 Student \*p = **new** **Student**(13);  58 **Student**::**Print**();  59 p->**Print**();  60 **delete** p;  61 **Student**::**Print**();  62 Student s[2];  63 s[0].**Print**();  64 s[1].**Print**();  65  66 **system**("pause");  67 **return** 0;  68 }  **example4\_03**    Circle.h  1 #pragma once  2  3 #ifndef \_Circle  4 #define \_Circle  5  6 **class** Circle  7 {  8 **private**:  9 **double** Radius;  10 **const double** PI;  11 **public**:  12 **Circle**(**double** r = 0) :**PI**(3.1415926)  13 {  14 Radius = r;  15 }  16 **double** **Area**();  17 **double** **Circumference**();  18 };  19  20 #endif  Circle.cpp  1 #include"Circle.h"  2  3 **double** **Circle**::**Area**()  4 {  5 **return** PI \* Radius \* Radius;  6 }  7 **double** **Circle**::**Circumference**()  8 {  9 **return** 2 \* PI\*Radius;  10 }  main.cpp  1 #include<iostream>  2 **using namespace** std;  3 #include"Circle.h"  4  5 **int** **main**()  6 {  7 Circle **c1**(3.5), c2;  8 cout << "area of c1=" << c1.**Area**() << ",circumference of c1=" << c1.**Circumference**();  9 cout << "area of c2=" << c2.**Area**() << ",circumference of c2=" << c2.**Circumference**();  10 **system**("pause");  11 **return** 0;  12 }  **example4\_04**    Circle.h  1 #pragma once  2  3 #ifndef \_Circle  4 #define \_Circle  5  6 **class** Circle  7 {  8 **private**:  9 **double** Radius;  10 **const double** PI;  11 **public**:  12 **Circle**(**double** r = 0) :**PI**(3.1415926)  13 {  14 Radius = r;  15 }  16 **double** **Area**();  17 **double** **Circumference**();  18 **double** **GetRadius**()**const**;  19 };  20  21 #endif  Circle.cpp  1 #include"Circle.h"  2  3 **double** **Circle**::**Area**()  4 {  5 **return** PI \* Radius \* Radius;  6 }  7  8 **double** **Circle**::**Circumference**()  9 {  10 **return** 2 \* PI\*Radius;  11 }  12  13 **double** **Circle**::**GetRadius**()**const**  14 {  15 **return** Radius;  16 }  main.cpp  1 #include<iostream>  2 **using namespace** std;  3 #include"Circle.h"  4  5 **int** **main**()  6 {  7 Circle **c1**(3.5), c2;  8 cout << "radius of c1=" << c1.**GetRadius**();  9 cout << ",area of c1=" << c1.**Area**() << ",circumference of c1=" << c1.**Circumference**()  10 <<endl;  11 cout << "radius of c2=" << c2.**GetRadius**();  12 cout << ",area of c2=" << c2.**Area**() << ",circumference of c2=" << c2.**Circumference**()  13 << endl;  14 **system**("pause");  15 **return** 0;  16 }  **example4\_05**    1 #include<iostream>  2 #include<string>  3 **using namespace** std;  4 #pragma warning(disable:4996)  5 **class** Person  6 {  7 **private**:  8 **int** age;  9 **char** \*name;  10 **public**:  11 **Person**(**int** n =1,**const char** \*na="Zhuli" );  12 ~**Person**();  13 **void** **Print**();  14 **void** **Print**()**const**;  15 **void** **ModifyAge**();  16 };  17  18 **Person**::**Person**(**int** n, **const char** \*na)  19 {  20 age = n;  21 name = **new** **char**[**strlen**(na) + 1];  22 **strcpy**(name, na);  23 }  24  25 **Person**::~**Person**()  26 {  27 **delete**[]name;  28 }  29  30 **void** **Person**::**Print**()  31 {  32 cout << "age=" << age << " name=" << name << endl;  33 cout << "This is general Print()." << endl;  34 }  35  36 **void** **Person**::**Print**()**const**  37 {  38 cout << "age=" << age << " name=" << name << endl;  39 cout << "This is general Print()." << endl;  40 }  41  42 **void** **Person**::**ModifyAge**()  43 {  44 age++;  45 }  46  47 **int** **main**()  48 {  49 **char** contentp1[10] = "wu";  50 **const** Person **p1**(17, contentp1);  51 cout << "Output const object p1" << endl;  52 p1.**Print**();  53 **char** contentp2[10] = "zhang";  54 Person **p2**(18, contentp2);  55 cout << "Output const object p2" << endl;  56 p2.**ModifyAge**();  57 p2.**Print**();  58 **system**("pause");  59  60 **return** 0;  61 }  学生管理系统      Student.h  1 #pragma once  2 #ifndef \_STUDENT  3 #define \_STUDENT  4 #include<iostream>  5 #include<string>  6 **using namespace** std;  7 #define SIZE 80  8  9 **class** Student  10 {  11 **char** \*name;*//??*  12 **char** ID[19];*//???????*  13 **char** number[10];*//??*  14 **char** speciality[20];*//????*  15 **int** age;*//???*  16 **static int** count;  17 **public**:  18 **Student**();  19 **Student**(**char** \*na, **char** \*id, **char** \*num, **char**\*spec, **int** ag);  20 **Student**(**const** Student &per);  21 ~**Student**();  22 **char**\***GetName**()**const**;  23 **char**\***GetID**();  24 **char**\***GetNumber**();  25 **char**\***GetSpec**();  26 **int** **GetAge**()**const**;  27 **void** **DisPlay**()**const**;  28 **void** **Input**();  29 **void** **Insert**();  30 **void** **Delete**();  31 **static int** **GetCount**();  32 };  33  34 #endif  Student.cpp  1 #include"student.h"  2 #pragma warning(disable:4996)  3 **int** **Student**::count = 0;  4  5 **Student**::**Student**() *//???????*  6 {  7 name = NULL;  8 age = 0;  9 }  10  11 **Student**::**Student**(**char** \*na, **char** \*id, **char** \*num, **char**\*spec, **int** ag) *//???????*  12 {  13 **if** (na)  14 {  15 name = **new** **char**[**strlen**(na) + 1];  16 **strcpy**(name, na);  17 }  18 **strcpy**(ID, id);  19 **strcpy**(number, num);  20 **strcpy**(speciality, spec);  21 age = ag;  22 count++;  23 }  24  25 **Student**::**Student**(**const** Student &per) *//????????*  26 {  27 **if** (per.name)  28 {  29 name = **new** **char**[**strlen**(per.name) + 1];  30 **strcpy**(name, per.name);  31 }  32 **strcpy**(ID, per.ID);  33 **strcpy**(number, per.number);  34 **strcpy**(speciality, per.speciality);  35 age = per.age;  36 count++;  37 }  38  39 **Student**::~**Student**() *//????*  40 {  41 cout << "disCon" << endl;  42 **if** (name)  43 **delete**[]name;  44 count--;  45 }  46  47 **char**\***Student**::**GetName**()**const**  48 {  49 **return** name;  50 }  51  52 **char**\***Student**::**GetID**()  53 {  54 **return** ID;  55 }  56  57 **int** **Student**::**GetAge**()**const**  58 {  59 **return** age;  60 }  61  62 **char**\***Student**::**GetSpec**()  63 {  64 **return** speciality;  65 }  66  67 **char**\***Student**::**GetNumber**()  68 {  69 **return** number;  70 }  71  72 **void** **Student**::**DisPlay**()**const**  73 {  74 cout << "???" << name << endl;  75 cout << "???????" << ID << endl;  76 cout << "???" << number << endl;  77 cout << "?????" << speciality << endl;  78 cout << "????" << age << endl;  79 }  80  81 **void** **Student**::**Input**()  82 {  83 **char** na[10];  84 cout << "???? ??";  85 cin >> na;  86 **if** (name)  87 **delete**[]name;  88 name = **new** **char**[**strlen**(na) + 1];  89 **strcpy**(name, na);  90 cout << "??????????";  91 cin >> ID;  92 cout << "???????";  93 cin >> age;  94 cout << "????????";  95 cin >> speciality;  96 cout << "???";  97 cin >> number;  98 count++;  99 }  100  101 **void** **Student**::**Insert**()  102 {  103 **if** (!age)  104 {  105 **Input**();  106 }  107 }  108  109 **void** **Student**::**Delete**()  110 {  111 age = 0;  112 count--;  113 }  114  115 **int** **Student**::**GetCount**()*//???????*  116 {  117 **return** count;  118 }  Main.cpp  1 #include<iostream>  2 **using namespace** std;  3 #include"student.h"  4 #pragma warning(disable:4996)  5 **const int** N = 10;  6 **void** **menu**();  7 **void** **OutputStu**(**const** Student \*array);  8 **void** **InputStu**(Student \*array);  9 **int** **SearchStu**(**const** Student \*array, **char** \*na);  10 **bool** **InsertStu**(Student \*array);  11 **bool** **DeleteStu**(Student \*array, **char** \*na);  12 **int** **main**()  13 {  14 Student array[N];  15 **int** choice;  16 **char** na[20];  17 **do**  18 {  19 **menu**();  20 cout << "Please input your choice:";  21 cin >> choice;  22 **if**(choice>=0&&choice<=5)  23 **switch** (choice)  24 {  25 **case** 1:**InputStu**(array);**break**;  26 **case** 2:  27 cout << "Input the name searched:" << endl;  28 cin >> na;  29 **int** i;  30 i = **SearchStu**(array, na);  31 **if** (i == N)  32 cout << "??????\n";  33 **else**  34 array[i].**DisPlay**();  35 **break**;  36 **case** 3:**OutputStu**(array);**break**;  37 **case** 4:**if** (**InsertStu**(array))  38 {  39 cout << "??????????\n";  40 }  41 **else**  42 {  43 cout << "??????\n";  44 }  45 **break**;  46 **case** 5:  47 cout << "Input the name deleted:" << endl;  48 cin >> na;  49 **if** (**DeleteStu**(array,na))  50 {  51 cout << "???????????\n";  52 }  53 **else**  54 {  55 cout << "???????\n";  56 }  57 **break**;  58 **default**:  59 **break**;  60 }  61 } **while** (choice);  62 **system**("pause");  63 **return** 0;  64 }  65  66 **void** **menu**()  67 {  68 cout << "\*\*\*\*\*\*\*\*1.?????\*\*\*\*\*\*\*\*" << endl;  69 cout << "\*\*\*\*\*\*\*\*2.??????\*\*\*\*\*\*\*\*" << endl;  70 cout << "\*\*\*\*\*\*\*\*3.??????\*\*\*\*\*\*\*\*" << endl;  71 cout << "\*\*\*\*\*\*\*\*4.?????\*\*\*\*\*\*\*\*" << endl;  72 cout << "\*\*\*\*\*\*\*\*5.??????\*\*\*\*\*\*\*\*" << endl;  73 cout << "\*\*\*\*\*\*\*\*0.???\*\*\*\*\*\*\*\*" << endl;  74 }  75  76 **void** **OutputStu**(**const** Student \*array)  77 {  78 cout << "??????=" << **Student**::**GetCount**() << endl;  79 **for** (**int** i = 0; i < N; i++)  80 **if** (array[i].**GetAge**())  81 array[i].**DisPlay**();  82 }  83  84 **int** **SearchStu**(**const** Student \*array, **char** \*na)  85 {  86 **int** i, j = N;  87 **for**(i=0;i<N;i++)  88 **if**(array[i].**GetAge**())  89 **if** (**strcmp**(array[i].**GetName**(), na) == 0)  90 {  91 j = i;  92 **break**;  93 }  94 **return** j;  95 }  96  97 **void** **InputStu**(Student \*array)  98 {  99 **char** ch;  100 **int** i = 0;  101 **do**  102 {**if** (**Student**::**GetCount**()==N)  103 {  104 cout << "?????????????" << endl;  105 }  106 **if** (!array[i].**GetAge**())  107 {  108 array[i++].**Input**();  109 }  110 cout << "????????Y or N?" << endl;  111 cin >> ch;  112 } **while** (ch=='Y');  113 }  114  115 **bool** **InsertStu**(Student\*array)  116 {  117 **if** (**Student**::**GetCount**()==N)  118 {  119 cout << "??????????????" << endl;  120 **return false**;  121 }  122 **for** (**int** i=0;array[i].**GetAge**();i++)  123 {  124 array[i].**Insert**();  125 }  126 **return true**;  127 }  128  129 **bool** **DeleteStu**(Student \*array, **char** \*na)  130 {  131 **if** (**Student**::**GetCount**()==0)  132 {  133 cout << "????????????" << endl;  134 **return false**;  135 }  136 **int** i = **SearchStu**(array, na);  137 **if** (i==N)  138 {  139 cout << "????????????\n";  140 **return false**;  141 }  142 array[i].**Delete**();  143 **return true**;  144 } | | | | | | |

：可根据内容自行拓展页面，作业内容尾部尽量不要留有空白