|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | C++ | | | | |
| 实验名称 | 实验六：继承下的构造函数与析构函数 | | | | |
| 实验时间 | 2021-11-26 | | 实验地点 | 1j705 | |
| 学 院 | 自动化与电气工程学院 | | 专 业 | 电气工程及其自动化 | |
| 姓 名 | 白志伟 | 班 级 | 电自2003 | 学 号 | 202030310228 |
| **实验报告内容基本要求参考格式**   1. 实验程序 2. 实验结果 3. 心得体会 | | | | | |
| 1. 实验程序   #include<iostream>  #include<string>  using namespace std;  class MyArray {  public:  MyArray(int length);  ~MyArray();  void Input();  void Display(string);  protected:  int\* alist;  int length;  };  MyArray::MyArray(int leng)  {  if (leng <= 0)  {  cout << "error length";  exit(1);  }  length = leng;  alist = new int[length];  if (alist == NULL)  {  cout << "assign failure";  exit(1);  }  cout << "MyArray类对象已创建!" << endl;  }  MyArray::~MyArray()  {  delete[] alist;  cout << "MyArray类对象已撤销!" << endl;  }  void MyArray::Display(string str)  {  int i;  int\* p = alist;  cout << str << length << "个整数: ";  for (i = 0; i < length; i++, p++)  cout << \*p << "";  cout << endl;  }  void MyArray::Input()  {  cout << "请从键盘输入" << length << "个整数:";  int i;  int\* p = alist;  for (i = 0; i < length; i++, p++)  cin >> \*p;  }  class SortArray : public MyArray {  public:  void Sort();  SortArray(int leng) :MyArray(leng)  {  cout << "SortArray类对象已创建!" << endl;  }  ~SortArray();  };  SortArray::~SortArray()  {  cout << "SortArray类对象已撤销!" << endl;  }  void SortArray::Sort()  {  int i, j, temp;  for (i = 0; i < length - 1; i++)  for (j = 0; j < length - i - 1; j++)  {  if (alist[j] > alist[j + 1])  {  temp = alist[j];  alist[j] = alist[j + 1];  alist[j + 1] = temp;  }  }  }  int main()  {  SortArray s(5);  s.Input();  s.Display("显示排序以前的");  s.Sort();  s.Display("显示排序以后的");  return 0;  }  二、实验结果    2021-12-15 (7)  三、心得体会  上机实验5仅仅是为了理解派生类的继承方式对基类各种声明属性的改变，而本次上机实验6就是对继承与派生类的一次实战，通过本次实战也是切身体会到了C++学科的实践难度，光有理论纸上谈兵显然是不行的，何况在学习第四章时就明显感到了一些吃力，希望通过耐心的复习和将来更多的学习后能更好的掌握这方面知识。还是之前说过的那个表格教会了我很多东西。还是要记得熟记于心！  2021-12-15 (8)  Copyright ©2021-2099 电自2003班 白志伟202030310228. All rights reserved | | | | | |