**第六次上机实验**

1. **感想：**

通过本次实验，我掌握了派生类的声明方法和派生类构造函数的定义方法，

并掌握不同方式下，构造函数与析构函数的执行顺序与构造规则。

这让我的编程水平得到了进一步的提升。

这次上机学习写出程序运行的结果。声明一个SortArray继承类MyArray，在该类中定义一个函数，具有将输入的整数从小到大进行排序的功能

出现问题时，考虑将基类Base中数据成员x的访问权限改为private时，会出现哪些错误，为什么？将基类Base中数据成员x的访问权限改为protected时，会出现哪些错误，为什么？在原程序的基础上，将派生类Derived的继承方式改为private时，会出现哪些错误，为什么？在原程序的基础上，将派生类Derived的继承方式改为protected时，会出现哪些错误，为什么？

这三个问题。

1. **代码：**

|  |
| --- |
| //版权声明  //Copyright 2021-2099 ZhantengZhang.  //202030310284 张展腾  #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 px();  SortArray(int leng);  ~SortArray();  };  SortArray::SortArray(int leng):MyArray(leng)  {  cout << "SortArray类对象已创建" << endl;  }  SortArray::~SortArray()  {  cout << "SortArray类对象已撤销" << endl;  }  void SortArray::px()  {  int \*p = alist;  for (int i = 0; i < length - 1; i++)  for (int j = i + 1; j < length; j++)  if (alist[i] < alist[j])  {  int t = p[i];  p[i] = p[j];  p[j] = t;  }  }  int main()  {  SortArray a(5);  a.Input();  a.Display("显示排序以前的");  a.px();  a.Display("显示排序之后的");  return 0;  } |

1. **截图：**

