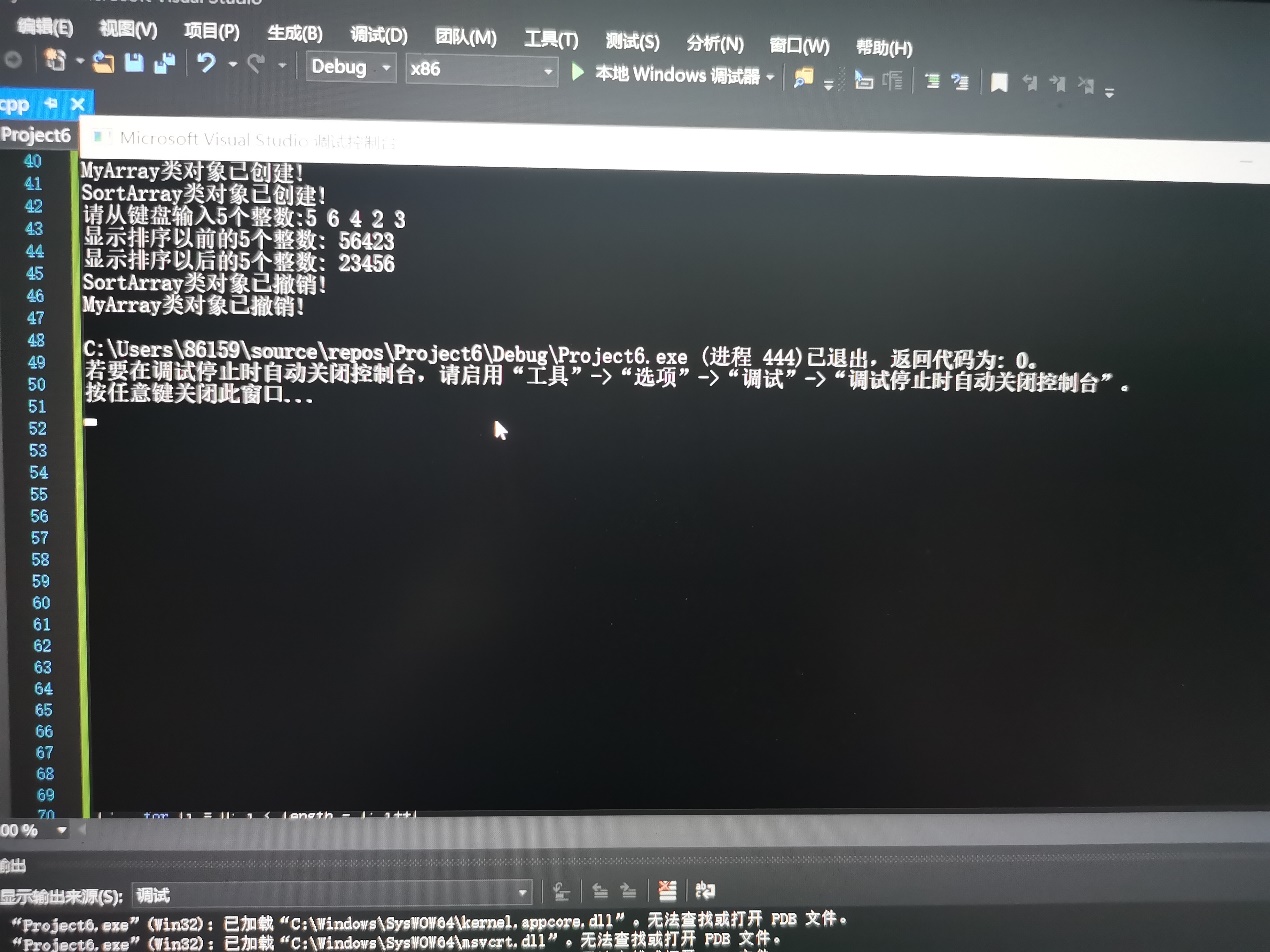
第六次实验报告：继承下的构造函数与析构函数

1. 程序代码
2. #include<iostream>
3. #include<string>
4. using namespace std;
5. class MyArray{
6. public:
7. MyArray(int length);
8. ~MyArray();
9. void Input();
10. void Display(string);
11. protected:
12. int \*alist;
13. int length;
14. };
15. MyArray::MyArray(int leng)
16. {
17. if(leng<=0)
18. {
19. cout<<"error length";
20. exit(1);
21. }
22. length = leng;
23. alist = new int [length];
24. if(alist == NULL)
25. { cout<<"assign failure";
26. exit(1);
27. }
28. cout<<"MyArray类对象已创建!"<<endl;
29. }
30. MyArray::~MyArray()
31. {
32. delete [ ] alist;
33. cout<<"MyArray类对象已撤销!"<<endl;
34. }
35. void MyArray::Display(string str)
36. {
37. int i;
38. int \*p=alist;
39. cout<<str<<length<<"个整数: ";
40. for(i=0;i<length;i++,p++)
41. cout<<\*p<<"";
42. cout<<endl;
43. }
44. void MyArray::Input()
45. {
46. cout<<"请从键盘输入"<<length<<"个整数:";
47. int i;
48. int \*p=alist;
49. for(i=0;i<length;i++,p++)
50. cin>>\*p;
51. }
52. int main()
53. {
54. MyArray a(5);
55. a.Input();
56. a.Display("显示已经输入的");
57. return 0;
58. }
59. class SortArray: public MyArray{
60. public:
61. void Sort();
62. SortArray(int leng):MyArray(leng)
63. {
64. cout<<"SortArray类对象已创建!"<<endl;
65. }
66. ~SortArray();
67. };
68. SortArray::~SortArray()
69. {
70. cout<<"SortArray类对象已撤销!"<<endl;
71. }
72. void SortArray::Sort()
73. {
74. int i,j,temp;
75. for(i=0;i<length-1;i++)
76. for(j=0;j<length-i-1;j++)
77. {
78. if(alist[j]>alist[j+1])
79. {
80. temp=alist[j];
81. alist[j]=alist[j+1];
82. alist[j+1]=temp;
83. }
84. }
85. }
86. int main()
87. {
88. SortArray s(5);
89. s.Input();
90. s.Display("显示排序以前的");
91. s.Sort();
92. s.Display("显示排序以后的");
93. return 0;
94. }
95. 2.程序结果
96. 
97. 3.心得体会
98. 学习本章内容，最重要的是搞清楚继承的法则，这是最重要的。不能犯继承错误的问题。要清晰地明白当基类中为公有继承、私有继承、保护继承时，在其派生类中的继承形式。尤其也要明白构造函数与析构函数的调用顺序。构造函数的调用顺序为：
99. 基类中的构造函数、对象成员的构造函数、派生类本身的构造函数，而析构函数的顺序与构造函数正好相反。