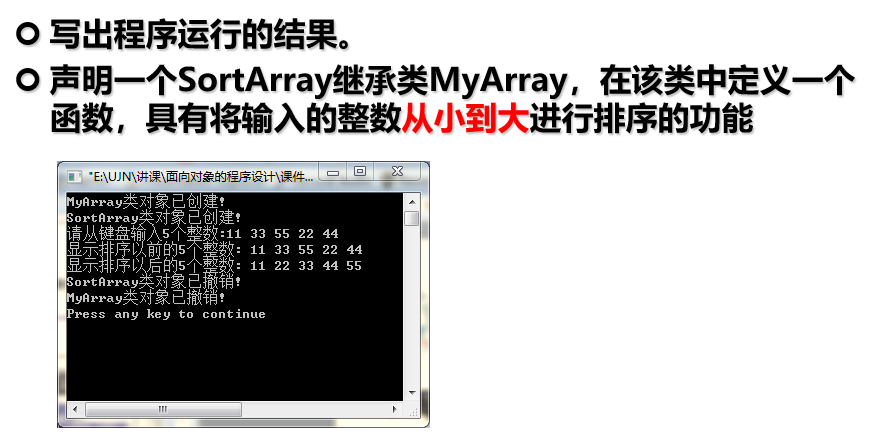
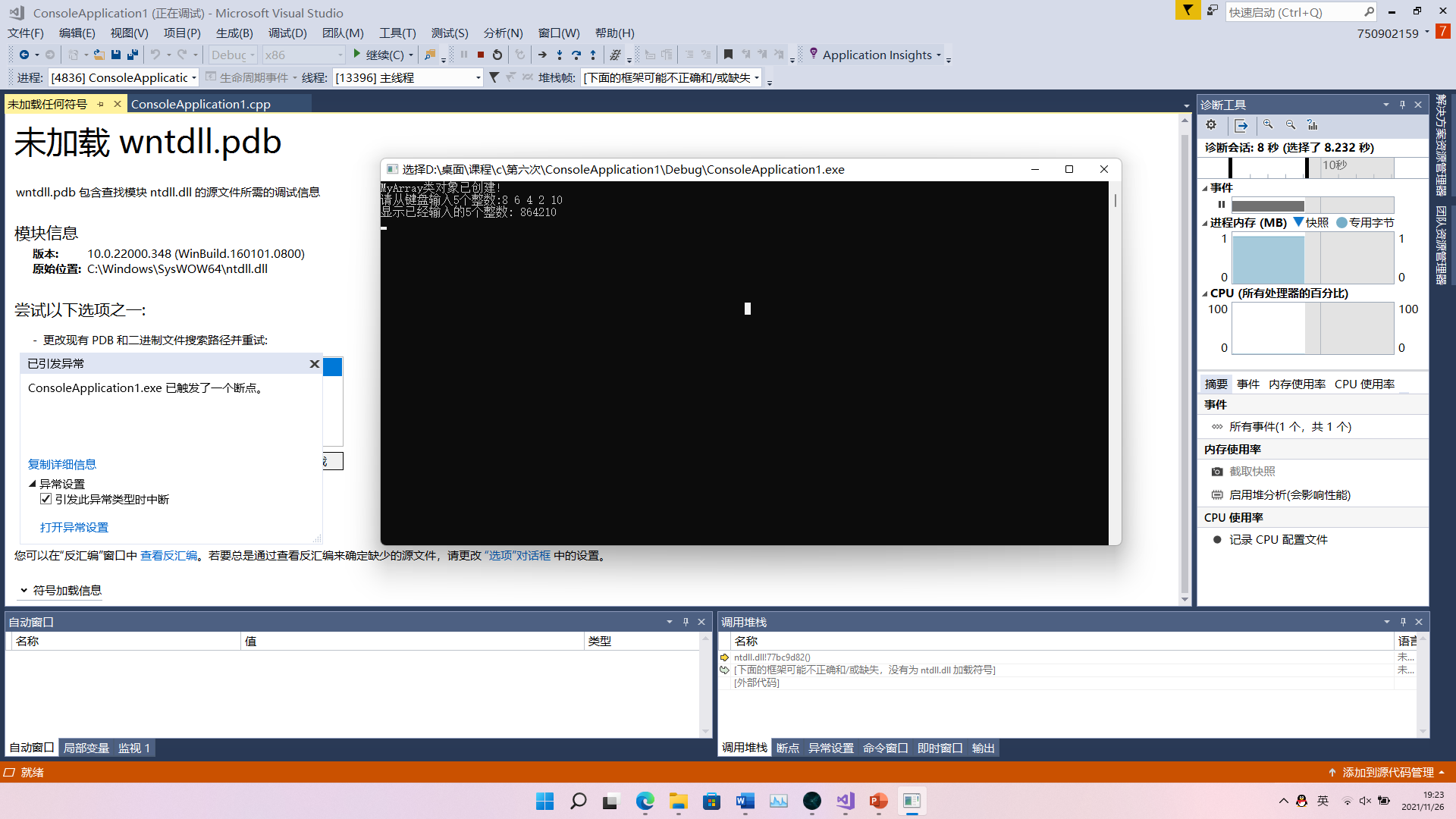
**第六次上机实验报告**

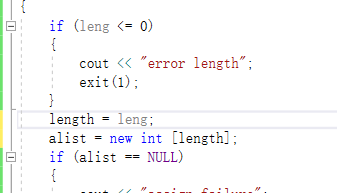
实验要求



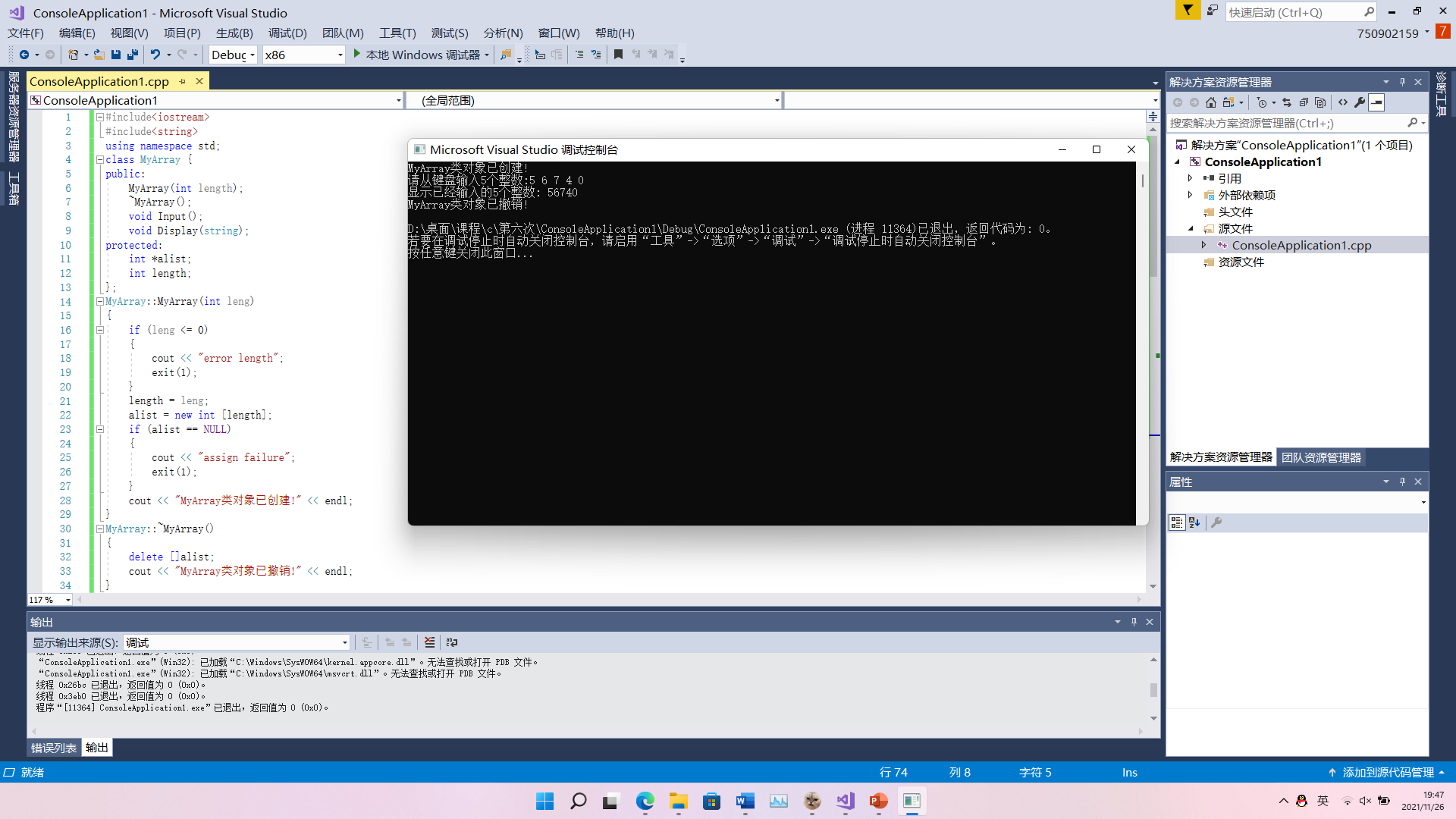
心得体会



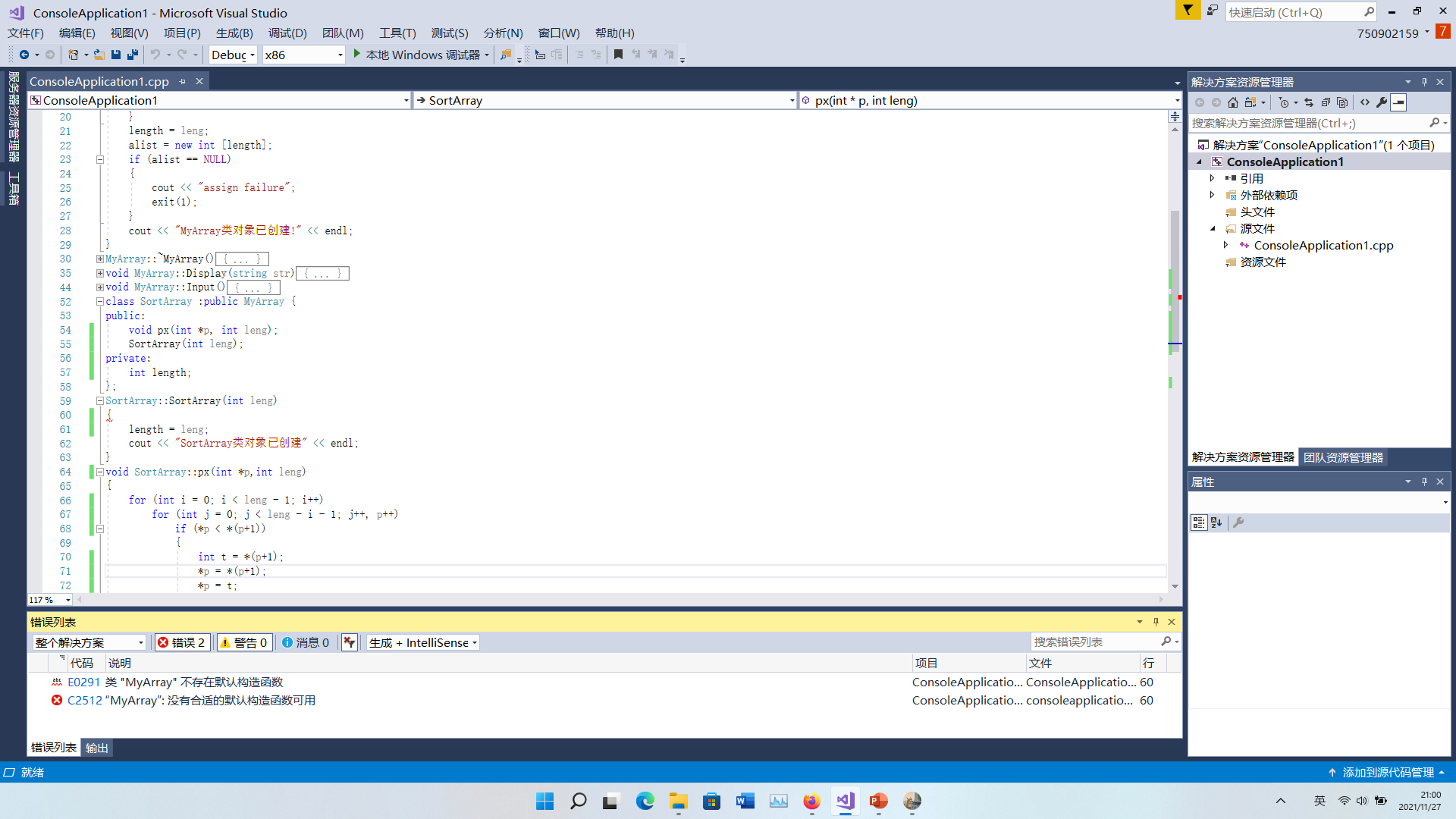
复制黏贴了ppt上的的代码，出现了未知的问题（经过百度查询应该是内存泄露的缘故）



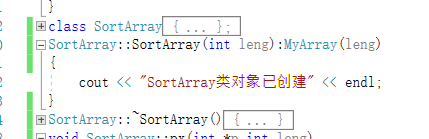
换了一下语句的顺序，应该是没有规定动态分配的大小



后来成功运行了此代码



在源代码的基础上做了部分修改，在自己敲代码的时候遇到了这个问题



查了一下继承方式，又改了一下

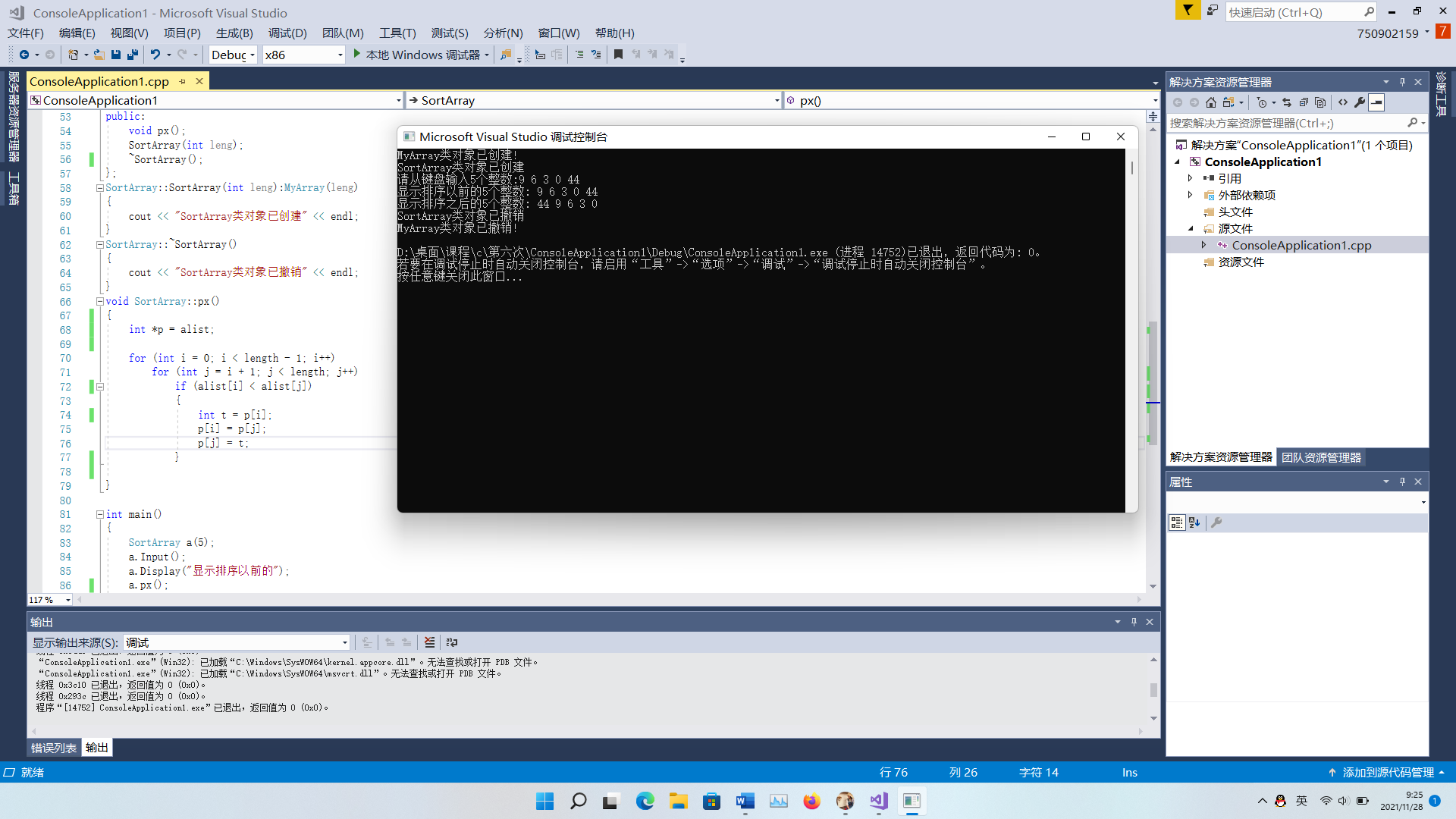
最终成功了

这次实验我发现自己在继承问题上的构造和析构函数的只是还有待加强，

SortArray:: SortArray (int leng):Myarray(leng)，这个含参的构造函数的继承我就是上网搜的。

这次实验还用到了new 和delete ，老师在ppt 上犯了一个错误，导致了内存泄漏的问题，原因是没给动态分配定义大小，当然在不同编译器上可能程序能运行。New 是我们在堆区（内存四区中的一区）自行调用的内存空间，new 之后一定要用delete 释放掉我们调用的内存空间

实验代码与程序结果



|  |
| --- |
| //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;  } |

版权声明

Copyright ©2021-2099 ZhantengZhang.

202030310284 张展腾