实验报告（第五次上机）

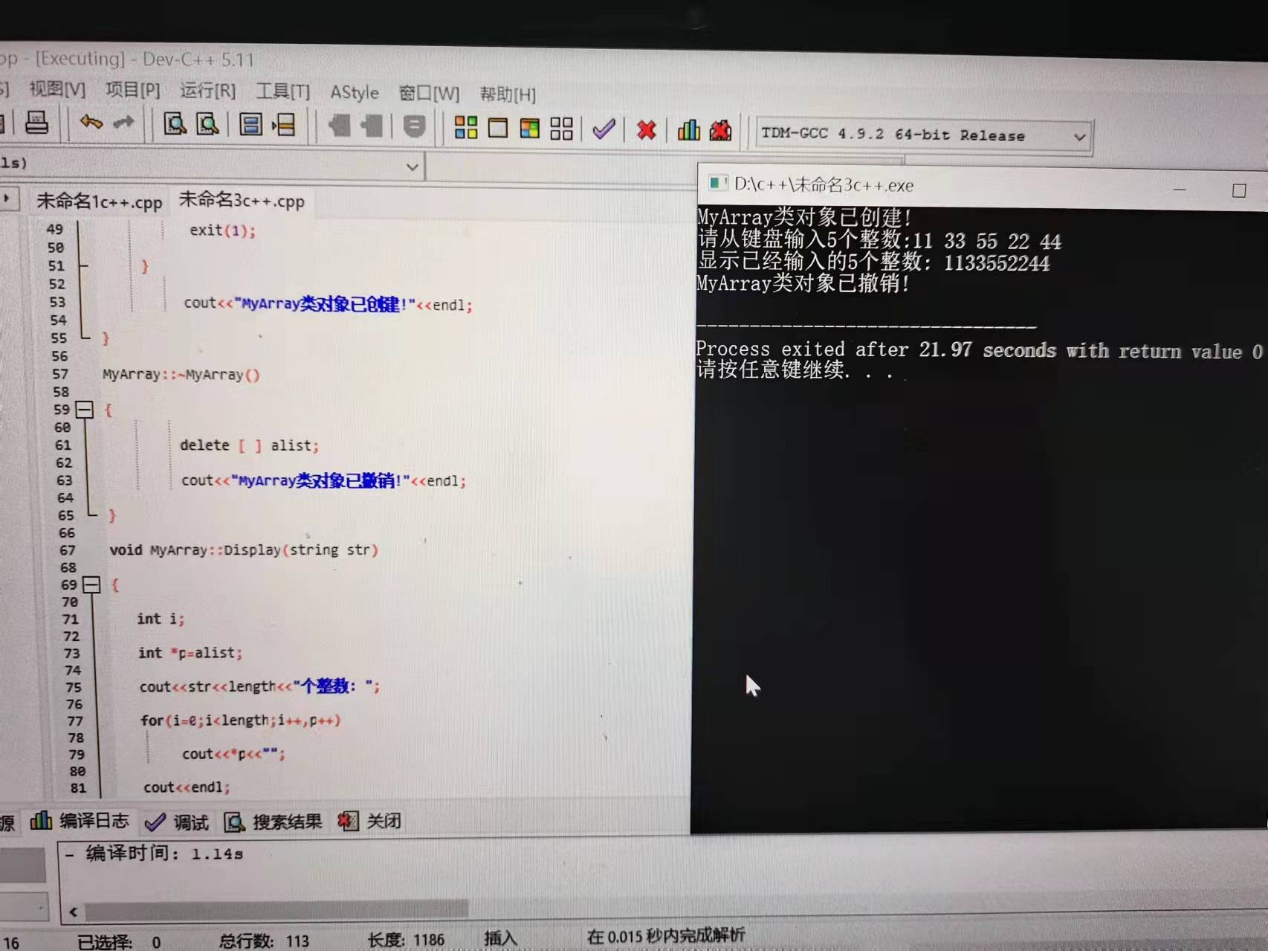
自2001 郭锴

#include<iostream>using namespace std;class Base{public: void setx(int i) { x=i; } int getx() { return x; }public: int x;};class Derived:public Base{public: void sety(int i) { y=i; } int gety() { return y; } void show() { cout<<"Base::x="<<x<<endl; // 语句1 }public: int y;};int main(){ Derived bb; bb.setx(16); bb.sety(25); bb.show(); cout<<"Base::x="<<bb.x<<endl; cout<<"Derived::y="<<bb.y<<endl;

cout<<"Base::x="<<bb.getx()<<endl;

cout<<"Derived::y="<<bb.gety()<<endl;

return 0;}图片：



心得：重要的一点是‌继承中基类与派生类之间的关系，继承虽然好用，利于减少代码量，增强可读性，加强代码之间的联系等，但是稍有不慎就会出现错误，继承出现的错误有的时候可并非容易找，毕竟这是继承的特性，例如基类中save函数与析构函数之间的关系没有把握好，就会产生操作失败的问题，因为派生类操作完毕后进行了一次保存，但是基类析构时这时若再调用save函数，保存的是基类中未修改的数据，直接覆盖掉了之前的数据，相当于没有执行操作。