2020 年 OOP 上机实验(6)

(Ver. 0.69 2020/05/14 wangxp@fudan.edu.cn)

一、实验目标

1） 编写完整的类，熟悉类的构造函数、析构函数、拷贝构造函数；熟悉操作符重载（含输入输出操作符）；熟悉成员函数与非成员函数的区别；熟悉友员函数；熟悉类的静态属性；

2）理解类中的 copy 操作，以及如何定义 copy 行为，包括复制构造函数和复制赋值操作；理解类中的 move 操作，以及如何定义 move 行为，包括移动构造函数和移动赋值操作；

3）理解类对象的创建和释放过程，理解并熟悉操作符 new/delete, new[]/delete[]的重载；

4）理解基于测试的程序开发过程。

二、实验内容

本次实验所给的文件包括：vec.h 和 vec_test.cpp 文件。只需要修改 vec.h，最后上传时也只需要上传 vec.h。

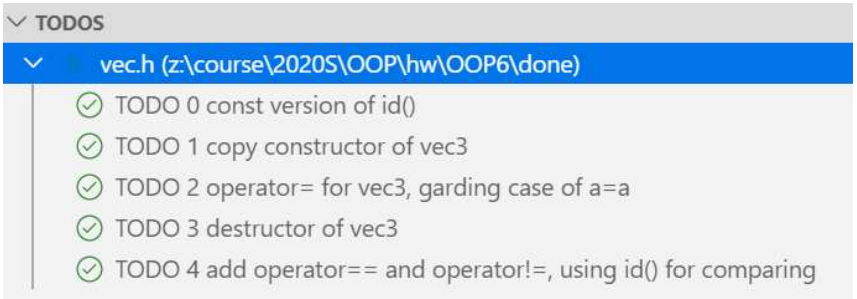
vec.h 中定义了三个模板类，分别是 vec1、vec2 和 vec3。

1）其中 vec1 没有定义任何成员方法和属性。但是在 test_vec1_ctor()，我们可以很清楚地看出，对于这样的类，编译器隐含地生成了很多函数；

2）其中 vec2 我们定义为资源类，包含一个 T*的动态数据。在 T2 中，我们定义了构造函数和析构函数；并且在析构函数中，对删除的资源进行了 nullptr 保护。但是很不幸，由于 C++编译器的默认行为生成了很多函数（看 test_vec2_ctor），包括复制/移动行为的函数，在执行赋值后，出现了两个指针指向同一个对象，并且一个指针丢失的现象(见代码 test_vec2_copy)，导致了程序崩溃(segment fault)；

3）在 vec3 中，通过增加控制 copy 行为的函数（复制构造函数和复制赋值操作符），精确地控制复制行为，保证资源的安全复制。

修改的代码集中在 vec3 中，在最终的测试中，需要把 main 函数中的 test_vec2_copy()这一行注释掉：因为 vec2 的实现会导致该函数直接崩溃。在源代码 vec.h 中，我们增加了 5 个 TODO，请大家根据程序的功能增加相应的代码。



参考资料

[1] 默认构造函数。 https://zh.cppreference.com/w/cpp/language/default_constructor

[2] 复制构造函数。 https://zh.cppreference.com/w/cpp/language/copy_constructor

[3] 移动构造函数。 https://zh.cppreference.com/w/cpp/language/move_constructor

[4] 复制赋值运算符。 https://zh.cppreference.com/w/cpp/language/copy_assignment

[5] 移动赋值运算符。 https://zh.cppreference.com/w/cpp/language/move_assignment