

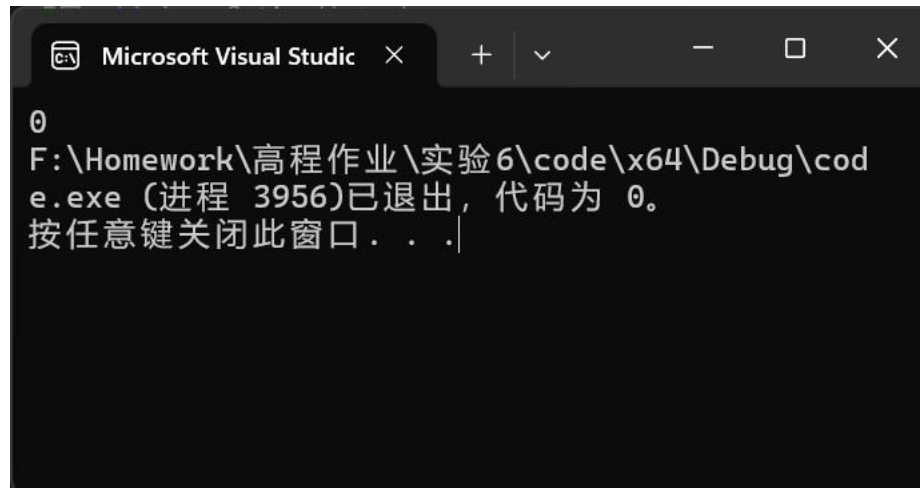
计算机学院高级语言程序设计课程实验报告

实验题目：实验六、类的继承		学号：202300130183
日期：2024 年 3 月 4 日	班级：2023 级智能班	姓名：宋浩宇
Email：202300130183@mail.sdu.edu.cn		
<p>实验目的：</p> <ol style="list-style-type: none">1. 学习使用和使用类的继承关系，声明派生类。2. 熟悉不同继承方式下对基类成员的访问控制。3. 学习利用虚基类解决二义性问题。		
<p>实验软件和硬件环境：</p> <p>实验软件：Windows 11 家庭中文版(x64) Visual Studio 2022</p> <p>硬件环境：处理器：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz</p> <p>RAM 32.0 GB (31.6 GB 可用)</p>		

实验步骤与内容:

第 7 章 1. (1)

```
1>----- 已启动生成: 项目: code, 配置: Debug x64 -----
1>lab7_1.cpp
1>F:\Homework\高程作业\实验6\code\lab7_1.cpp(18,6): error C2248: "Animal::age": 无法访问 private 成员(在 "Animal" 类中声明)
1>F:\Homework\高程作业\实验6\code\lab7_1.cpp(7,6): message : 参见 "Animal::age" 的声明
1>F:\Homework\高程作业\实验6\code\lab7_1.cpp(4,7): message : 参见 "Animal" 的声明
1>已完成生成项目 "code.vcxproj" 的操作 - 失败。
```



对于基类中的成员,只有在以公有的方式继承,并且对应成员为公有时,在派生类中这个成员才是可见的,其余情况都是不可见的。

第 7 章 1. (2)



第 7 章 1. (3)

```

1>----- 已启动生成: 项目: code, 配置: Debug x64 -----
1>lab7_3.cpp
1>F:\Homework\高程作业\实验6\code\lab7_3.cpp(54,38): error C2385: 对“MaxSpeed”的访问不明确
1>F:\Homework\高程作业\实验6\code\lab7_3.cpp(54,38): message : 可以是基“vehicle”中的“MaxSpeed”
1>F:\Homework\高程作业\实验6\code\lab7_3.cpp(54,38): message : 也可以是基“vehicle”中的“MaxSpeed”
1>F:\Homework\高程作业\实验6\code\lab7_3.cpp(54,55): error C2679: 二元“<<”: 没有找到接受“motorcycle”类型的右操作数的运算符(或没有可接受的转换)

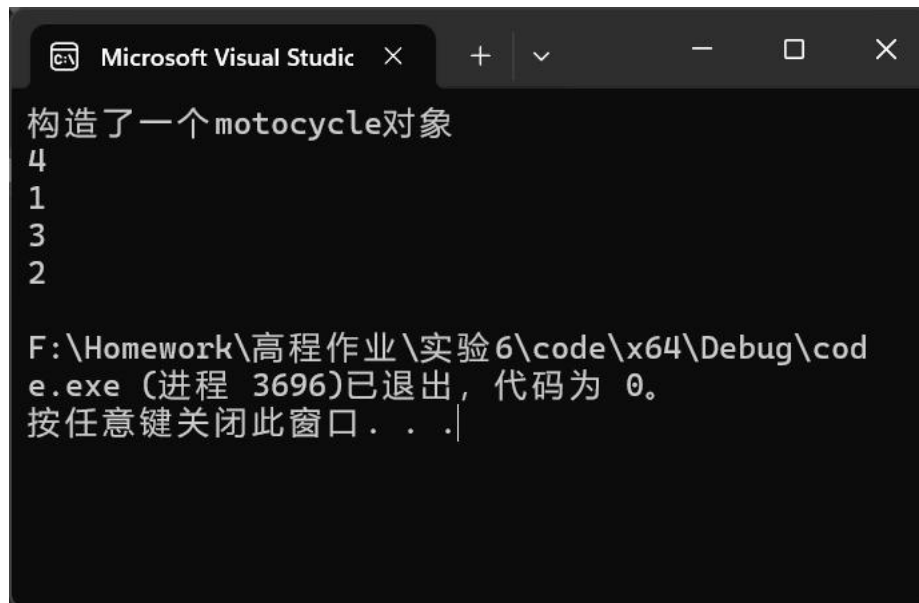
```

```

1>F:\Homework\高程作业\实验6\code\lab7_3.cpp(54,95): error C2385: 对“Weight”的访问不明确
1>F:\Homework\高程作业\实验6\code\lab7_3.cpp(54,95): message : 可以是基“vehicle”中的“Weight”
1>F:\Homework\高程作业\实验6\code\lab7_3.cpp(54,95): message : 也可以是基“vehicle”中的“Weight”

```

如果不把 vehicle 设置成虚基类会出现二元性错误，即对 MaxSpeed 和 Weight 的访问不明确，如果设置成虚基类就不会出现这种错误



```

Microsoft Visual Studio
构造了一个motorcycle对象
4
1
3
2

F:\Homework\高程作业\实验6\code\x64\Debug\code.exe (进程 3696)已退出, 代码为 0。
按任意键关闭此窗口...

```

第 7 章 2. 不能运行，想要运行的话需要为 Point 类添加对应参数的构造函数和默认构造函数，并且为 Rectangle 类添加默认构造函数。

第 7 章 3. move(), getX() 和 getY() 这三个函数是函数隐藏，它与函数重载的区别是，因为在派生类中声明了与基类同名的函数，所以基类中的同名函数在派生类中变得不可见。函数重载是定义多个参数列表不同的同名函数，在使用的函数同名而输入的参数不同的时候调用不同的函数。

第 7 章 4. 会报错，Derived 没有合适的默认构造函数，这是因为在显式地定义了含参构造函数之后，编译器就不会隐式地自动生成的默认构造函数了，这个时候是不存在默认构造函数的，所以如果以没有参数的形式来声明一个 Derived 对象，没有可用的默认构造函数就会出现这种问题。若想解决的话就需要在原本的 Derived 类中添加默认构造函数。

第 7 章 6. 无法通过编译，即 Derived 类中的 fun() 确实隐藏了父类中形参不同的同名函数。解释如下：因为在派生类中声明了与基类同名的函数，因此基类中这个函数名就无法直接通过函数名调用了，此时是进行了函数隐藏，即基类中的函数在派生类中不可见了，若想调用只能通过派生类中的函数通过::运算符来调用。

第 7 章 7. 在补充定义了含参构造函数之后

```
Derived(int d) { var0 = d; }
```

运行结果如下

```
Microsoft Visual Studio  X + - □ X
1 1 1
Member of Base0
2 2 2
3 3 3

F:\Homework\高程作业\实验6\code\x64\Debug\code.exe (进程 28136)已退出, 代码为 0。
按任意键关闭此窗口. . .
```

输出解释如下:首先根据补充定义的构造函数此时var0被赋值为1,因为Derived类的基类是以虚基类的方式继承的Base0的,因此此处的var0并不会产生二义性错误,而是有一个确定的var0,因此可以直接d.var0进行访问,而使用d.Base1::var0和d.Base2::var0访问的也是同一个成员,因此第一行输出的是三个1,而经由d.var0=2更改后输出的是三个2,经过d.Base2::var0=3更改后输出的是三个3;而关于虚基类的成员函数,因为使用的是Base1和Base2使用的是虚基类来继承Base0,因此对其中的函数fun0()调用也不会出现二义性错误,因此第二行成功执行fun0()输出Member of Base0

第七章 8. 继承和派生在该例子中的作用为:让LinearEqu类获得了除了构造函数和析构函数以外的所有Matrix类的成员,实现了代码的复用;而且由于基类Matrix类的成员是公有和保护类型,也控制了LinearEqu类对基类中成员的访问权限,实现了数据的保护。

第七章 9. 在新的代码中,涉及到与类的继承相关的是:Account类作为基类,派生出SavingsAccount和CreditAccount类,而这两个派生类都将基类的成员原样地继承过来,因此不需要为每个派生类重复定义数据成员另外,基类将record函数设为protected函数,使得派生类处理每一笔具体账目时可以调用该函数来改变余额并输出账目信息。基类的show函数也被派生类继承了下来,但由于CreditAccount在show函数中需要输出额外的信息,因此该类重新定义了一个同名的函数,该函数首先通过Account::show的方式调用了基类的show函数,然后再将额外的信息输出。

结论分析与体会：

类的继承对于代码的复用有着重要意义，也是面向对象编程对于编程效率提高的体现之一，而且相较于类的组合，类的继承对于基类的数据成员的访问权限要求更严格，对于代码的封装性和安全性也有重要作用。另外虚基类的设置也是给了类的继承更广的应用范围。

就实验过程中遇到的问题及解决处理方法（如有）：

未遇到问题