河南工業大學

面向对象程序设计

实验报告

项目名称：继承

专业班级：联软件 2001

学　　号：202041030102

学生姓名：张圃源

实验成绩：

批阅老师：

2021 年 11 月 20 日

目录

项目 3 继承 (4 学时)

1 目的要求

（1）能够使用继承构造继承层次结构，掌握数据成员的声明和成员函数的声明及定义方法；（2）能够使用组合重写继承层次结构，理解组合与继承的区别。

2 实验内容

（1）程序编写：根据问题描述和程序的输出结果，对给出的程序代码进行修改，最终给出自己的解决方案，本次实验内容包括：a. 创建一个继承层次结构：基类 Account，派生类 SavingsAccount 和 CheckAccount 都继承自 Account；b. 使用组合重写继承层次结构 CommissionEmployee–BasePlusCommissionEmployee 中的 BasePlusCommissionEmployee。（2）程序调试：根据给出的存在问题的动物（Animal）类层次结构的程序代码，a. 修改程序中的编译错误使之能够正确地编译执行；b. 对照程序的正确输出结果，修改程序中的逻辑错误使其输出结果和给定的正确输出结果一致。

3 实验说明

面向对象程序设计中最重要的一个概念是继承。继承允许我们依据另一个类来定义一个类，这使得创建和维护一个应用程序变得更容易。这样做，也达到了重用代码功能和提高执行效率的效果。当一个类派生自基类，该基类可以被继承为 public、protected 或 private 几种类型。继承类型是通过上面讲解的访问修饰符 access-specifier 来指定的。

我们几乎不使用 protected 或 private 继承，通常使用 public 继承。当使用不同类型的继承时，遵循以下几个规则：

公有继承（public）：当一个类派生自公有基类时，基类的公有成员也是派生类的公有成员，基类的保护成员也是派生类的保护成员，基类的私有成员不能直接被派生类访问，但是可以通过调用基类的公有和保护成员来访问。保护继承（protected）：当一个类派生自保护基类时，基类的公有和保护成员将成为派生类

的保护成员。私有继承（private）：当一个类派生自私有基类时，基类的公有和保护成员将成为派生类的私有成员。

当创建一个类时，您不需要重新编写新的数据成员和成员函数，只需指定新建的类继承了一个已有的类的成员即可。这个已有的类称为基类，新建的类称为派生类

4 实验过程及结果

承代表了 is a 关系。例如，哺乳动物是动物，狗是哺乳动物，因此，狗是动物，等等。

代码如下：

```
// 基类
class Animal {
    // eat() 函数
    // sleep() 函数
};


//派生类
class Dog : public Animal {
    // bark() 函数
};
```

4.1 基类 派生类

一个类可以派生自多个类，这意味着，它可以从多个基类继承数据和函数。定义一个派生类，我们使用一个类派生列表来指定基类。类派生列表以一个或多个基类命名，形式如下：

class derived-class: access-specifier base-class 其中，访问修饰符 access-specifier 是 public、protected 或 private 其中的一个，base-class 是之前定义过的某个类的名称。如果未使用访问修饰符 access-specifier，则默认为 private。

假设有一个基类 Shape，Rectangle 是它的派生类，如下所示：

```
#include <iostream>

using namespace std;

// 基类
```

```
class Shape
{
   public:
      void setWidth(int w)
      {
         width = w;
      }
      void setHeight(int h)
      {
         height = h;
      }
   protected:
      int width;
      int height;
};

// 派生类
class Rectangle: public Shape
{
   public:
      int getArea()
      {
         return (width * height);
      }
};

int main(void)
{
   Rectangle Rect;

   Rect.setWidth(5);
   Rect.setHeight(7);

   // 输出对象的面积
   cout << "Total area: " << Rect.getArea() << endl;

   return 0;
```

```
Total area: 35

--------------------------------
Process exited after 0.03526 seconds with return value 0
请按任意键继续. . .
```

图 1: Caption

}

4.2 多继承

多继承即一个子类可以有多个父类，它继承了多个父类的特性。

C++ 类可以从多个类继承成员，语法如下：

class < 派生类名 >:< 继承方式 1>< 基类名 1>,< 继承方式 2>< 基类名 2>,… < 派生类类体 > ;

```
#include <iostream>

using namespace std;

// 基类 Shape
class Shape
{
   public:
      void setWidth(int w)
      {
         width = w;
      }
      void setHeight(int h)
      {
         height = h;
      }
   protected:
      int width;
      int height;
};
```

```
// 基类 PaintCost
class PaintCost
{
   public:
      int getCost(int area)
      {
         return area * 70;
      }
};

// 派生类
class Rectangle: public Shape, public PaintCost
{
   public:
      int getArea()
      {
         return (width * height);
      }
};

int main(void)
{
   Rectangle Rect;
   int area;

   Rect.setWidth(5);
   Rect.setHeight(7);

   area = Rect.getArea();

   // 输出对象的面积
   cout << "Total area: " << Rect.getArea() << endl;

   // 输出总花费
   cout << "Total paint cost: $" << Rect.getCost(area) <<
       endl;
```

```
Total area: 35
Total paint cost: $2450

--------------------------------
Process exited after 0.03479 seconds with return value 0
请按任意键继续. . .
```

图 2: Caption

```
    return 0;
}
```

5 实验小结

1. 基类 private 成员在派生类中无论以什么方式继承都是不可见的。这里的不可见是指基类的私有成员还是被继承到了派生类对象中，但是语法上限制派生类对象不管在类里面还是类外面都不能去访问它。2. 基类 private 成员在派生类中是不能被访问，如果基类成员不想在类外直接访问，但需要在派生类中能访问，就定义为 protected。可见，保护成员限定符 protected 是因为继承才出现的。3. 表格里的访问方式都是取最小的“权限”。4. 使用关键字 class 时默认的继承方式是 private，使用 struct 的默认继承方式是 public，不过最好显示地写出继承方式。5.** 在实际运用中一般都使用的是 public 继承，几乎很少去使用 protected/private 继承，** 也不提倡去使用。因为 protected/private 继承下来的成员都只能在派生类的类里面使用，实际中的扩展维护性不强