

**实 验 报 告**

**（ 2023 / 2024 学年 第 二 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 面向对象程序设计及C++ | | | | | |
| 实验名称 | 实验三： 多态性实验 | | | | | |
| 实验时间 | 2024 | 年 |  | 月 |  | 日 |
| 指导单位 | 计算机学院、软件学院、网络空间安全学院 | | | | | |
| 指导教师 |  | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 |  | 班级学号 |  |
| 学院(系) | 计软学院 | 专 业 | 信息安全 |

| **实 验 报 告** | | | |
| --- | --- | --- | --- |
| **实验名称** | **多态性实验** | | |
| **实验类型** | **综合** | **实验学时** | **2** |
| 1. **实验目的和要求**   （1）熟悉类的设计、运用继承与派生机制设计派生类，合理设置数据成员和成员函数。  （2）掌握双目运算符、单目运算符的重载方法，对常用算术运算符能在自定义类中通过友元函数、成员函数进行重载，以实现静态多态性。  （3）掌握通过继承、虚函数、基类的指针或引用实现动态多态性的方法。  （4）理解并掌握有纯虚函数的抽象类的作用，在各派生类中重新定义各纯虚函数的方法，以及此时实现的动态多态性。 | | | |
| 二、**实验环境(实验设备)**  硬件: 微型计算机  软件: Windows 操作系统、Microsoft Visual Studio 2010 | | | |
| **三、实验原理及内容**  **实验题目1：**定义点类Point，有两个double 类型的数据成员x 和y，分别表示横坐标和纵  坐标，要求完成如下内容。  （1）定义坐标默认值为原点(0.0,0.0)的构造函数。  （2）以成员函数形式重载：前置“++”运算符和双目运算符“−”。  （3）用友元函数形式重载：双目运算符“+”（两种版本，详见实验指导部分）、插  入运算符。  （4）先根据main()主函数代码和运行结果，补充类的定义和相关函数的定义，写出完  整程序。  （5）程序正确后，删除main()函数体，根据运行结果，自己重新完成main()函数。  main()主函数代码如下。**（**中文五号宋体，英文五号Consolas字体，单倍行距)  int main()  {  Point pt1(10.5,20.8),pt2(-5.3,18.4),pt3;  cout<<"original pt1,pt2,pt3 are:\n";  cout<<pt1<<pt2<<pt3;  pt3=pt1+100.8;  cout<<"after pt3=pt1+100.8, pt3 is:"<<pt3;  pt3=pt1+pt2;  cout<<"after pt3=pt1+pt2, pt3 is:"<<pt3;  pt3=++pt1;  ++pt2;  cout<<"after ++ pt1,pt2,pt3 are:\n";  cout<<pt1<<pt2<<pt3;  pt3=pt1-pt2;  cout<<"after pt3=pt1-pt2, pt3 is:"<<pt3;  return 0 ;  }  程序运行结果如下。  original pt1,pt2,pt3 are:  (10.5,20.8)  (-5.3,18.4)  (0,0)  after pt3=pt1+100.8, pt3 is:(111.3,121.6)  after pt3=pt1+pt2, pt3 is:(5.2,39.2)  after ++ pt1,pt2,pt3 are:  (11.5,21.8)  (-4.3,19.4)  (11.5,21.8)  after pt3=pt1-pt2, pt3 is:(15.8,2.4)  **实验解答：**   1. 类Point的构造函数：   Point(double x = 0.0, double y = 0.0) : x(x), y(y)  {  this->x = x;  this->y = y;  }   1. 用成员函数重载：前置“++”运算符和双目运算符“−”：   Point& operator++()  {  ++x;  ++y;  return \*this;  }  Point operator-(Point& other)  {  return Point(x - other.x, y - other.y);  }   1. 用友元函数形式重载：双目运算符“+”（两种版本，详见实验指导部分）   friend Point operator+(const Point& p1, const Point& p2); // Point + Point  friend Point operator+(const Point& p1, double value); // Point + Double   1. 友元函数重载插入运算符：   Point operator+(const Point& p1, const Point& p2)  {  return Point(p1.x + p2.x, p1.y + p2.y);  }  Point operator+(const Point & p1, double value)  {  return Point(p1.x + value, p1.y + value);  }   1. 程序正确后，根据运行结果，重新完成的main()函数：   int main()  {  Point pt1(10.5, 20.8), pt2(-5.3, 18.4), pt3;  cout << "original pt1,pt2,pt3 are:\n";  pt1.show();  pt2.show();  pt3.show();  pt3 = pt1 + 100.8;  cout << "after pt3=pt1+100.8, pt3 is: ";  pt3.show();  pt3 = pt1 + pt2;  cout << "after pt3=pt1+pt2, pt3 is: ";  pt3.show();  pt3 = ++pt1;  ++pt2;  cout << "after ++ pt1,pt2,pt3 are:\n";  pt1.show();  pt2.show();  pt3.show();  pt3 = pt1 - pt2;  cout << "after pt3=pt1-pt2, pt3 is:" << endl;  pt3.show();  return 0;  }  **实验题目2**: 定义一个抽象类容器类，其中定义了若干纯虚函数，实现求表面积、体积、输出等功能。由此抽象类派生出正方体、球体和圆柱体等多个派生类，根据需要定义自己的成员变量，在各个派生类中重新定义各纯虚函数，实现各自类中相应功能，各个类成员的初始化均由本类构造函数实现。  ① 在主函数中，定义容器类的指针和各个派生类的对象，使指针指向不同对象处调用相同的函数能执行不同的函数代码，从而实现动态多态性。  ② 定义一个顶层函数void TopPrint(Container &r);使得主函数中调用该函数时，根据实在参数所有的类自动调用对应类的输出函数。  ③ 主函数中定义一个Container类对象，观察编译时的错误信息，从而得出什么结论？  **实验解答：**  （1）基类Container的定义见实验教材。  （2）各个派生类的定义，根据提示进行填写完整代码：  ① //正方体类，从Container类公有继承，定义构造函数，重新定义基类的3个纯虚函数  class Cube : public Container {  private:  double radius;  public:  Cube(double r = 0) : radius(r) {};  virtual double area() {  return 6 \* radius \* radius;  }  virtual double volumn() {  return radius \* radius \* radius;  }  virtual void print() {  using namespace std;  cout << "Cube's area is " << area() << " and cube's volume is " <<  volumn() << endl;  }  };  ② //球类，从Container类公有继承，定义构造函数，重新定义基类的3个纯虚函数  class Sphere : public Container{  private:  double radius;  public:  Sphere(double r = 0) : radius(r) {};  virtual double area() {  return 4 \* PI \* radius \* radius;  }  virtual double volumn() {  return 4 / 3 \* PI \* radius \* radius \* radius;  }  virtual void print() {  using namespace std;  cout << "Sphere's area is " << area() << " and sphere's volume is " <<  volumn() << endl;  }  };  ③ //圆柱体类，从Container类公有继承，需要增加的成员变量，定义构造函数，重新定义基类的三个纯虚函数  class Cylinder : public Container {  private:  double radius;  public:  Cylinder(double r = 0) : radius(r) {};  virtual double area() {  return 2 \* PI \* radius \* radius;  }  virtual double volumn() {  return PI \* radius \* radius \* radius;  }  virtual void print() {  using namespace std;  cout << "Cylinder's area is " << area() << " and cylinder's volume is " <<  volumn() << endl;  }  };  **（3）正确定义各派生类对象，记录程序的运行结果是：**  正方体对象：\_\_\_\_\_\_\_\_\_\_\_\_ cube(1)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  球体对象：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ sphere(2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_  圆柱体对象：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ cylinder(3)\_\_\_\_\_\_\_\_\_\_\_\_\_  运行结果：6  1  Cube's area is 6 and cube's volume is 1  50.264  25.132  Sphere's area is 50.264 and sphere's volume is 25.132  56.547  84.8205  Cylinder's area is 56.547 and cylinder's volume is 84.8205  （4）主函数中定义一个Container类对象，编译器的报错信息：   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 严重性 | 代码 | 说明 | 项目 | 文件 | 行 | 禁止显示状态 | 详细信息 | | 错误(活动) | E0322 | 不允许使用抽象类类型 "Container" 的对象: | Draft | C:\Users\Robin\OneDrive\文档\VisualStudio\Draft\Draft\Draft\Dratf.cpp | 72 |  |  |   试说明原因：  这是因为尝试实例化一个抽象类Container，而抽象类不能直接实例化。抽象类是包含至少一个纯虚函数的类，这些纯虚函数需要在派生类中实现。由于Container类中包含纯虚函数area、volumn和print，所以它是一个抽象类。  **四、实验小结**（包括问题和解决方法、心得体会、意见与建议等）  **（**中文五号宋体，英文五号Consolas字体，单倍行距)  (一)实验中遇到的主要问题及解决方法  1.在题目（1）中在主函数中定义Container类的对象，**会产生**报错信息，解释原因。  这是因为尝试实例化一个抽象类Container，而抽象类不能直接实例化。抽象类是包含至少一个纯虚函数的类，这些纯虚函数需要在派生类中实现。由于Container类中包含纯虚函数area、volumn和print，所以它是一个抽象类。  2. 在题目（2）中通过代码验证，请总结友元函数与成员函数在实现重载时的区别。  友元函数和成员函数在实现重载时主要区别在于它们的定义位置、调用方式、访问权限以及在具体实现中的参数传递方式。  3．请总结前置++与后置++在实现重载时的不同  前++：  操作顺序：先递增，再返回对象。  返回类型：返回对象的引用，这样可以支持链式调用（如++(++obj)）。  效率：通常比后置++更高效，因为不涉及对象的拷贝。  后++：  操作顺序：先返回对象当前状态，然后递增。  返回类型：返回对象的副本，而不是引用。  效率：通常比前置++效率低，因为需要创建和返回对象的副本。  4.其它问题及解决方法：  无  （二）实验心得  通过这次实验，我对面向对象编程中的多态性有了更深入的理解和掌握，尤其是静态多态性和动态多态性的实现方法。静态多态性通过操作符重载实现，而动态多态性则是通过继承和虚函数来实现的。这些概念在实际编程中非常重要，能够帮助我们设计出更加灵活和可扩展的程序。此外，通过实验，我还学会了如何正确使用友元函数和成员函数进行操作符重载，并理解了它们之间的区别。  （三）意见与建议（没有可省略）  在实验过程中，实验指导书提供的例子和代码片段非常有帮助，但我建议在指导书中增加一些更多的实际应用案例和扩展练习，以帮助学生更好地理解和应用这些概念。同时，增加对常见问题的解答和解决方案的详细说明也会对学生的学习过程大有裨益。 | | | |

**实 验 报 告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **支撑毕业要求指标点**   **信息安全：**  1.2-M掌握计算机软硬件相关工程基础知识，能将其用于分析信息安全领域的相关工程问题。  3.1-H掌握信息安全领域所涉及的软硬件系统，从数字电路、计算机系统、到各类系统软件的基本理论与设计结构。 | | | | | | |
| **六、指导教师评语** | | | | | | |
| **评 分 细 则** | **评分项** | **优秀** | **良好** | **中等** | **合格** | **不合格** |
| **遵守实验室规章制度** |  |  |  |  |  |
| **学习态度** |  |  |  |  |  |
| **算法思想准备情况** |  |  |  |  |  |
| **程序设计能力** |  |  |  |  |  |
| **解决问题能力** |  |  |  |  |  |
| **算法设计合理性** |  |  |  |  |  |
| **算法效能评价** |  |  |  |  |  |
| **报告书写认真程度** |  |  |  |  |  |
| **内容详实程度** |  |  |  |  |  |
| **文字表达熟练程度** |  |  |  |  |  |
| **其它评价意见** |  | | | | |
| **本次实验能力达成评价（总成绩）** |  | **批阅人** |  | **日期** |  |