**淮北师范大学电子版实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **姓名** | **张元节** | | **学号** | **20181251215** | **实验日期** | **2019.12.6** | **成绩** |
| **院别** | **计算机科学与技术学院** | | **班级** | **2018级计算机类三班** | **实验场地** | **A201** |
| **实验课程名称** | | **面向对象程序设计** | | | | |  |
| **实验项目名称** | | **多态性实验** | | | | |
| **对于每一个实验项目，实验报告（含预习）一般应包含以下内容：第一部分——预习后的书面汇报。其主要内容应包含：\*1、实验目的；\*2、实验内容。第二部分——实验结果的书面汇报。其主要内容应包含： \*3、实验源代码；\*4、实验结果及分析（含实验测试输入数据，试验运行结果截图，用简洁的语言总结实验，汇报是否达到实验目的）；\*5、实验体会、问题讨论（谈体会或感想、提出建议或意见、讨论与实验有关的且自己感兴趣的问题、回答课后思考题）。** | | | | | | | |
| **一、实验目的** | | | | | | | |
| （1）掌握运算符重载的基本方法；  （2）掌握虚函数的定义和使用方法；  （3）掌握纯虚函数和抽象类的概念和用法。 | | | | | | | |
| **二、实验内容** | | | | | | | |
| （1）声明Point类，有坐标x、y两个成员变量；对类重载++和--运算符，实现坐标值的改变。  （2）声明一个People（人员）类，具有属性如下：number（编号）、name（姓名）、sex（性别）、birthday（出生日期）、id（身份证号）。其中“出生日期”是一个子对象（请自行设计日期类）。用成员函数实现对人员信息的录入和显示。在测试程序中声明People类的对象数组，录入数据并显示。同时为People重载运算符“==”，运算符“==”为判断两个People对应的id属性是否相同。  （3）声明一个车Vehicle基类，有run、stop等成员函数，由此派生出自行车Bicycle类、汽车Motorcar类，从Bicycle、Motorcar类派生出摩托车Motorcycle类，它们都有run、stop等成员函数，观察虚函数的作用。 | | | | | | | |
| **三、实验源程序（或实验步骤）** | | | | | | | |
| **（1）声明Point类，有坐标x、y两个成员变量；对类重载++和--运算符，实现坐标值的改变。**  **#include<iostream>**  **using namespace std;**  **class Point{**  **private:**  **float x;**  **float y;**  **public:**  **Point(float a=0,float b=0);**  **void print();**  **//友元运算符重载 ,前缀方式++a**  **friend Point operator++(Point &a);**  **friend Point operator--(Point &a);**    **};**  **Point::Point(float a,float b) {**  **x=a;**  **y=b;**  **}**  **void Point::print(){**  **cout<<"x="<<x<<",y="<<y<<endl;**  **}**  **Point operator++(Point &a){**  **++a.x;++a.y;**  **return a;**  **}**  **Point operator--(Point &a){**  **--a.x;--a.y;**  **return a;**  **}**  **int main(){**  **Point A1(3.1,2.5);**  **A1.print();**  **++A1;**  **A1.print();**  **return 0;**  **}**  **（2）声明一个People（人员）类，具有属性如下：number（编号）、name（姓名）、sex（性别）、birthday（出生日期）、id（身份证号）。**  **其中"出生日期"是一个子对象（请自行设计日期类）。用成员函数实现对人员信息的录入和显示。**  **在测试程序中声明People类的对象数组，录入数据并显示。**  **同时为People重载运算符"=="，运算符"=="为判断两个People对应的id属性是否相同。**  **#include<iostream>**  **#include<string>**  **#define N 2**  **using namespace std;**  **class Date{**  **public:**  **Date(int y,int m,int d);**  **void set(int y,int m,int d);**  **void show();**  **private:**  **int year;**  **int month;**  **int day;**  **};**  **Date::Date(int y,int m,int d)**  **{**  **year=y;month=m;day=d;**  **}**  **void Date::set(int y,int m,int d)**  **{**  **year=y;month=m;day=d;**  **}**  **void Date::show()**  **{**  **cout<<"生日："<<year<<"年"<<month<<"月"<<day<<"日"<<endl;**  **}**  **class People{**  **public:**  **People(int y=2000,int m=00,int d=00,string i="340321200000000000",string na="张三",string nu="20000000000",char se='m');**  **void set(int y=2000,int m=00,int d=00,string i="340321200000000000",string na="张三",string nu="20000000000",char se='m');**  **void show();**  **//成员函数重载运算符**  **int operator==(People a);**  **private:**  **Date birthday;**  **string id;//身份证号**  **string name;**  **string number;**  **char sex;**  **};**  **People::People(int y,int m,int d,string i,string na,string nu,char se):birthday(y,m,d)**  **{**  **id=i;**  **name=na;**  **number=nu;**  **sex=se;**  **}**  **void People::set(int y,int m,int d,string i,string na,string nu,char se)**  **{**  **id=i;**  **name=na;**  **number=nu;**  **sex=se;**  **birthday.set(y,m,d);**  **}**  **void People::show()**  **{**  **cout<<"身份证号："<<id<<endl<<"姓名："<<name<<endl<<"编号："<<number<<endl<<"性别："<<sex<<endl;**  **birthday.show();**  **}**  **int People::operator==(People a){**  **if((\*this).id==a.id)return 1;**  **return 0;**  **}**  **int main()**  **{**  **People p[N];**  **int y;int m;int d;string i1;string na;string nu;char se;//定义年龄名称等参数，用于传入set函数**  **for(int i=0;i<N;i++)**  **{**  **cout<<"请输入第"<<i+1<<"位人员信息："<<endl;**  **cout<<"出生年月日：";**  **cin>>y>>m>>d;**  **cout<<"身份证号：";**  **cin>>i1;**  **cout<<"姓名：";**  **cin>>na;**  **cout<<"编号：";**  **cin>>nu;**  **cout<<"性别：";**  **cin>>se;**  **// cin>>y>>m>>d>>i1>>na>>nu>>se;**  **p[i].set(y,m,d,i1,na,nu,se);**  **cout<<"第"<<i+1<<"位人员信息:"<<endl;**  **p[i].show();**  **}**  **if(p[0]==p[1])cout<<"二者身份证号相同"<<endl;**  **else cout<<"二者身份证号不同"<<endl ;**  **return 0;**  **}**  **（3）声明一个车Vehicle基类，有run、stop等成员函数，**  **由此派生出自行车Bicycle类、汽车Motorcar类，从Bicycle、Motorcar类派生出摩托车Motorcycle类，**  **它们都有run、stop等成员函数，观察虚函数的作用。**  **#include<iostream>**  **using namespace std;**  **class Vehicle{**  **public:**  **virtual void run()=0;**  **virtual void stop()=0;**  **};**  **class Bicycle:virtual public Vehicle{**  **public:**  **void run();**  **void stop();**    **};**  **void Bicycle::run(){**  **cout<<"自行车运行中"<<endl;**    **}**  **void Bicycle::stop(){**  **cout<<"自行车停住了"<<endl;**    **}**  **class Motorcar:virtual public Vehicle{**  **public:**  **void run();**  **void stop();**    **};**  **void Motorcar::run(){**  **cout<<"摩托车运行中"<<endl;**    **}**  **void Motorcar::stop(){**  **cout<<"摩托车停住了"<<endl;**    **}**  **class Motorcycle:public Bicycle,public Motorcar{**  **public:**  **void run();**  **void stop();**  **} ;**  **void Motorcycle::run(){**  **cout<<"Motorcycle运行中"<<endl;**  **}**  **void Motorcycle::stop(){**  **cout<<"Motorcycle停住了"<<endl;**  **}**  **int main(){**  **Vehicle \*p;**  **Bicycle ob1;**  **Motorcar ob2;**  **Motorcycle ob3;**  **p=&ob1;p->run();p->stop();**  **p=&ob2;p->run();p->stop();**  **p=&ob3;p->run();p->stop();**  **return 0;**  **}** | | | | | | | |
| **四、实验结果及分析** | | | | | | | |
| 1. **程序运行界面如下图：**     **结果符合实验预期，说明实验代码正确。**   1. **程序运行时，我们共需要输入两名学生信息，输入他们的信息后，运行结果如图:**     **程序运行过程中输入实验要求数据。后得到结果符合实验预期，说明实验代码正确。**   1. **程序运行结果如图：**     **上述运行界面符合实验预期，说明实验代码正确无误。** | | | | | | | |
| **五、实验体会、问题讨论** | | | | | | | |
| 1. **通过这次实验，我掌握了运算符重载的基本方法，积累了使用友元运算符重载以及使用对象运算符重载的大量实践经验，学习了两种重载方式的优缺点对比。** 2. **在实验过程中，我理解并且实践了虚函数的定义和使用方法，学会了如何定义虚函数，并将其应用在实际的编程项目中去。** 3. **在使用虚函数以及纯虚函数的过程中，我了解了虚函数的常见应用范围，加深了对虚函数使用方法的记忆，遇到并解决了使用虚函数过程中常见的错误。** 4. **在掌握纯虚函数的过程中，我了解了纯虚函数的概念，并且学习了使用父类指针调用子类方法的实际操作流程。** 5. **此外，我加深了关于抽象类的概念和用法，学习了将抽象类运用实际的能力，期间遭遇了许多问题，但通过努力都加以克服了。** | | | | | | | |
| **教师签字： 郑颖 批改日期：** | | | | | | | |