|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称： 面向对象程序设计 | | 指导教师： 高文超 | | |
| 班级： 信工17-1 | 姓名： 徐响 | | 学号： 1710480117 | |
| 实验项目名称：  类的继承与派生 | | | | |
| 实验目的及要求：  1.理解继承的含义，掌握派生类的定义方法和实现。  2.理解和掌握公有继承、私有继承和保护继承对基类成员的访问机制。  3.深刻理解在各种继承方式下构造函数和析构函数的执行顺序。  4.理解虚基类的目的和作用。 | | | | |
| 实验内容（方法和步骤）：  必做题目：  1.验证性题目：  （1）调用次序如下    （2）  执行次序如下：    （3）  程序中的错误见下图：    （4）  程序运行结果如下：    2.设计题：  （1）  代码：  #include <iostream>  #include <string>  using namespace std;  class Course {  public:  void setCno(int cNumber) { cno = cNumber; }  void setCredit(double crd) { credit = cno; }  void setCname(string cname) { courseName = cname; }  int getCno() { return cno; }  double getCredit() { return credit; }  string getCourseName(){ return courseName; }  Course(int Cno=0,double cre=0, string cName="") { setCourse(Cno, cre,cName); }  void display() {  cout<<"课程号: "<<cno<<"\t 课程名称: "<<courseName<<"\t 学分: " <<credit<<endl;}  void setCourse(int Cno = 0,double cre = 0,string cName = "") {  cno = Cno;  credit = cre;  courseName = cName ;}  private:  int cno;  double credit ;  string courseName;  };  class Student {  public:  void setSno( int Snumber) {  sno =Snumber; }  void setStudentName(string Sname) { stuName = Sname; }  int getSno() { return sno; }  string getStudentName() { return stuName;}  Student(int Sno = 0,string SName =" ") { setStudent(Sno, SName); }  void display() { cout<<"学 号: "<<sno <<"\t 姓名: "<<stuName<<endl; }  void setStudent(int Sno = 0,string Sname = "") { sno = Sno; stuName = Sname;}  private:  int sno;  string stuName;  };  class SelectCourse {  private:  int maxNum,curNum;  Course course;  Student \*stu;  public:  SelectCourse() { stu = new Student[10]; }  SelectCourse(Course C,int mNum=10,int cNum=0,Student s[]=nullptr):course(C), maxNum(mNum),  curNum( cNum), stu(new Student [maxNum]) {  for (int i = 0; i <cNum; i++)  stu[i] = s[i];  }  ~SelectCourse() { delete []stu; }  SelectCourse(const SelectCourse &o):course(o.course), maxNum(o.maxNum), curNum(o.curNum) {  stu = new Student[o.maxNum];  for (int i = 0; i < o.curNum; i++)  stu[i] = o.stu[i];}  SelectCourse& operator=(const SelectCourse o) {  course = o.course;  maxNum = o.maxNum;  curNum = o.curNum;  for(int i=0;i<o.curNum;i++)  stu[i] = o.stu[i];  return \*this;  }  void setCourse(Course C) {course = C; }  void setMaxNum(int n) { maxNum = n; }  void setCurNum(int n) { curNum = n; }  int getMaxNum() { return maxNum; }  int getCurNum() { return curNum; }  Course getCourse(){ return course;}  Student\* getStudent() { return stu; }  void setStudent(Student s[]) { stu = s; }  Student getAt(int n) { return stu[n-1]; }  void appendStudent(Student s){  if( curNum<maxNum)  stu[ curNum++]=s;  }  void display() {  course.display();  cout<< "最多选课人数:"<< maxNum << "\t实选人数:" << curNum << endl;  cout<< "选课学生名单:" << endl;  for(int i=0; i<curNum; i++)  stu[i].display();  }  };  void main(){  Course course;  course.setCourse(101, 4, "C++面向对象程序设计");  Student s[2], s1;  s[0].setStudent(1710480101, "李一");  s[1].setStudent(1710480102, "李二");  SelectCourse sc(course,10,2,s);  cout<<"-----------------sc------------------------"<<endl;  sc.display();  //下面的代码段测试SelectCourse类的拷贝构造函数和添加选课学生函数的运行情况  SelectCourse sc2, sc1 = sc;  s1.setStudent(1710480103, "李三");  sc1.appendStudent(s1);  cout<<"----------------------sc1(sc)----------------"<<endl;  sc1.display();  //下面的代码段测试SelectCourse类的赋值运算符函数的运行情况  sc2 = sc1;  cout <<"---------------------sc2=sc1------------------"<<endl ;  sc2.display();  //下面的代码段测试SelectCourse类中获取学生名单和人数的成员函数的运行情况  Student \*sname = sc2. getStudent();  cout <<"-------------------sc2.getStudent()----------"<<endl;  for(int i=0; i<sc2.getCurNum(); i++)  (sname++)->display();  }  运行结果如下：    （2）  代码：  #include< iostream>  using namespace std;  #include <string>  class Date  { public:  Date(){}  Date(int y,int m,int d);  Date(Date &d);  void SetDate(int y,int m,int d);  void ShowDate();  private:  int year,month,day;  };  Date::Date(int y,int m,int d){year=y; month=m;day=d; }  Date::Date(Date &d){ year=d.year; month=d.month; day=d.day; }  void Date::SetDate(int y,int m,int d)  { year=y; month=m;  day=d; }  void Date::ShowDate()  { cout<<year<<"年"<<month<<"月"<<day<<"日"<<endl; }  class Time  {public:  Time(){}  Time(int h,int m,int s);  Time(Time &t);  void SetTime(int h,int m,int s);  void ShowTime();  private:  int hour,minute,second;  };  Time::Time(int h,int m,int s){hour=h; minute=m;second=s; }  Time::Time(Time &t){ hour=t.hour;  minute=t.minute;  second=t.second; }  void Time::SetTime(int h,int m,int s){ hour=h; minute=m;  second=s; }  void Time::ShowTime(){ cout<<hour<<":"<<minute<<":"<<second<<endl; }  class Publication  {public:  Publication(){}  Publication(string title,string name,float price,int y,int m,int d);  void inputData();  void display();  private:  string title;  string name;  float price;  Date date;  };  Publication::Publication(string title,string name,float price,int y,int m,int d):title(title),name(name),price(price ),date(y,m,d){}  void Publication::inputData( )  {cout<<"Please input title, name, price, publication date:"<<endl;  cin>>title>>name>>price;  int year,month,day;  cin>>year>> month>>day;  date.SetDate(year,month,day);  }  void Publication::display()  { cout<<"title="<<title<<endl;  cout<<"name= "<<name<<endl;  cout<<"price="<<price<<endl;  cout<<"date=";  date.ShowDate();  }  class Book:public Publication  {public:  Book(){}  Book(string title,string name,float price,int y,int m,int d,int page);  void inputData();  void display();  private:  int page;  };  Book::Book(string title,string name,float price,int y,int m,int d,int page)  :Publication(title,name,price,y,m,d){this->page=page;}  void Book::inputData()  {Publication::inputData();  cout<<"Please the pages of the book:"<<endl;  cin>>page;  }  void Book::display()  {  Publication::display();  cout<<" page="<<page<<endl;  }  class CD:public Publication  {public:  CD(){}  CD(string title,string name, float price,int y,int m,int d,Time playtime);  void inputData();  void display();  private:  Time playtime;  };  CD::CD(string title,string name,float price,int y,int m,int d,Time playtime)  :Publication(title,name,price,y,m,d),playtime(playtime){}  void CD::inputData()  {  Publication::inputData();  cout<<"Please the playtime of the tape:"<<endl;  int hour,minute,second;  cin>>hour>> minute>>second;  playtime.SetTime(hour,minute,second);  }  void CD::display()  { Publication::display();  cout<<"playtime=";  playtime.ShowTime();  }  int main()  {Book book1("教材","C+ +程序设计",30.00,2019,6,1 ,300);  book1.display();  cout<<endl;  Time time1(30,10,20);  CD cd1("光盘","C++程序设计视频", 10.00,2020,5,1,time1);  cd1.display();  Book book2;  book2.inputData();  book2.display();  cout<<endl;  CD cd2;  cd2.inputData();  cd2.display();  return 0;  }  运行结果如下：    选做题：  1.  代码：  #include<iostream>  #include<iomanip>  #include<string>  using namespace std;  class person{  private:  string name;  int age;  string sex;  public:  void inputdata(){  cin >> name >> age >> sex;  }  void display(){  cout << setw(8) << name << setw(8) << age << setw(8) << sex ;  }  };  class Teacher:public person{  private:  int number;  char post[8];  char department[8];  public:  void inputdata()  {  cout << "请输入教师姓名、年龄、性别、编号、职称、系别" << endl;  person::inputdata();  cin >> number >> post >> department;  }  void display()  {  cout << setiosflags(ios::left);  cout << setw(8) <<"姓名" << setw(8) << "年龄" << setw(8) << "性别" << setw(8) << "编号" << setw(8) << "职称" << setw(8) << "系别" << endl;  person::display();  cout << setw(8) << number << setw(8) << post << setw(8) << department << endl;  }  };  class student:public person{  private:  int number;  char grade[10];  int china;  int math;  int english;  public:  void inputdata()  {    cout << "请输入学生姓名、年龄、性别、学号、班级、语文、数学、英语三门课程成绩" << endl;  person::inputdata();  cin >> number >> grade >> china >> math >> english ;  }  void display()  {  cout << setiosflags(ios::left);  cout << setw(8) << "姓名 " << setw(8) << "年龄 " << setw(8) << "性别 " << setw(12) << "学号 " << setw(12) << "班级 " << setw(10) << "语文成绩 " << setw(10) << "数学成绩 " << setw(10) << "英语成绩 " << endl;  person::display();  cout << setw(12) << number << setw(12) << grade << setw(10) << china << setw(10) << math << setw(10) << english << endl;  }  };  void main()  {  Teacher t;  t.inputdata();  t.display();  student s;  s.inputdata();  s.display();  }  运行结果：    2.  代码：  #include<iostream>  using namespace std;  class Shape  {  public:  virtual float printArea() const {return 0.0;};  };  class Circle:public Shape  {  public:  Circle(float =0);  virtual float printArea() const {return 3.14159\*radius\*radius;}  protected:  float radius;  };  Circle::Circle(float r):radius(r)  {  }  class Rectangle:public Shape  {  public:  Rectangle(float =0,float =0);  virtual float printArea() const;  protected:  float height;  float width;  };  Rectangle::Rectangle(float w,float h):width(w),height(h){  }  float Rectangle::printArea()const  {  return width\*height;  }  class Triangle:public Shape  {  public:  Triangle(float =0,float =0);  virtual float printArea() const;  protected:  float height;  float width;  };  Triangle::Triangle(float w,float h):width(w),height(h){  }  float Triangle::printArea()const  {  return 0.5\*width\*height;  }  void printArea(const Shape&s)  {  cout<<s.printArea()<<endl;  }  int main()  {  Circle circle(12.6);  cout<<"area of circle=";  printArea(circle);  Rectangle rectangle(4.5,8.4);  cout<<"area of rectangle=";  printArea(rectangle);  Triangle triangle(4.5,8.4);  cout<<"area of triangle=";  printArea(triangle);  }  运行结果： | | | | |
| 实验结果与分析：  这次实验让我更好地理解集成，巩固了派生类的定义方法和实现方法，在不同的集成下狗在函数和析构函数有不同的执行顺序，我也了解了虚基类的作用，在这次实验中得到了收获。 | | | | |
| 成绩： | 批阅教师签名： | | | 2020 年 5 月 14 日 |