**实验一 面向对象程序设计**

学生姓名： 黄丹 学 号： 6130116093 专业班级：航天班

实验类型：□ 验证 □ 综合 □ 设计 □ 创新 实验日期： 2019/3/20 实验成绩：

1. 实验项目名称

面向对象程序设计

1. 实验目的

1、基本掌握Java语言；

2、掌握面向对象编程；

1. 掌握开发工具Eclipse的使用；
2. 掌握Java类的定义及类的继承；
3. 掌握类的使用的方法；
4. 掌握函数重载的方法；
5. 掌握函数重写的方法。
6. 实验基本原理
7. 主要仪器设备及耗材

PC微机

Windows 操作系统

Eclipse/JDK1.8编程环境

github

五、实验内容

1、熟悉在Eclipse环境下开发java应用，编译执行“java实验（第三周）.rar”中的代码。

2、编写程序，设计一个判别素数的函数，每次在主函数中输入一个整数，调用这个判别函数，并输出是否为素数的信息，当某次输入的整数是小于等于0的数时退出该程序。

3、定义一个二维坐标内的点类DPoint，模拟实现对点的坐标的设置、读取，在坐标上移动点及计算两个点之间的距离的功能。

4、在3的基础上，定义一个三维坐标内的点类TPoint，模拟实现对点的坐标的设置、读取，在坐标上移动点及计算两个点之间的距离的功能。

5、声明一个交通工具类，描述它们的一些特性，由此派生出陆地交通工具类、空中交通工具类、及水上交通工具类，并编写成员函数描述它的一些基本功能。

6、声明一个Student类，在该类中包括一个数据成员score（分数）、两个静态数据成员total\_score（总分）和count（学生人数）；还包括一个成员函数account（）用于设置分数、累计学生的成绩之和、累计学生人数，一个静态成员函数sum（）用于返回学生的成绩之和，另一个静态成员函数average（）用于求全部成绩的平均值。在main函数中，输入某班同学的成绩，并调用上述函数求出全班同学的成绩之和和平均分。

7、编写一个学生和教师的数据输入和显示程序。学生数据有编号、姓名、性别、年龄、系别和成绩，教师数据有编号、姓名、性别、年龄、职称和部门。要求将编号、姓名、性别、年龄的输入和显示设计成一个类Person，并作为学生类Student和教师类Teacher的基类。

8、按要求阅读、编辑、编译、调试和运行以下程序。

（1）阅读、编辑、编译、调试和运行以下程序，并写出程序运行结果。

public class MyArray{

public MyArray(int leng)

{

if(leng<=0)

{

System.out.println("error length");

System.exit(1);

}

alist=new int [leng];

length=leng;

System.out.println("MyArray 类对象已创建。");

}

public void init()

{

int i;

for(i=0;i<length;i++)

alist[i] = length - i;

}

public void display()

{

int i;

System.print("MyArray中的元素为：");

for(i=0;i<length;i++)

System.out.print(“\t d”, alist[i]);

System.out.println();

}

protected int \*alist;

protected int length;

}

public class ArrayTest

{

public static void main()

{

MyArray a = new MyArray(5);

a.input();

a.display();

}

}

（2）声明一个类SortArray继承类MyArray，在该类中定义一个函数，具有将数组中元素从小到大进行排序的功能。

【提示】：请注意编写继承类的构造函数、析构函数，以及修改主函数。

（3）声明一个类ReArray继承类MyArray，在该类中定义一个函数，即有将数组中的元素进行倒置的功能。

【提示】：请注意编写继承类的构造函数、析构函数，以及修改主函数。

（4）声明一个类Average继承类MyArray，在该类中定义一个函数，即有将输入的整数进行求平均值的功能。

【提示】：请注意编写基继承类的构造函数、析构函数，以及修改主函数。

（5）声明一个NewArray类，复用已有类SortArray，ReArray和Average的代码，使得NewArray对象同时具有排序、倒置和求平均值的功能。

9、给出下面的抽象基类container。

abstract class Container{

protected double radius;

public Container(double radius);

abstact double surface\_area();

abstract double volume();

};

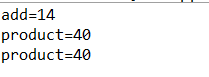
建立3个继承Container的派生类Cube、Sphere与Cylinder，让每一个派生类都包含抽象函数surface\_area（）和volume（），分别用来计算正方体、球体和圆柱体的表面积及体积。

1. 实验数据及处理结果

1、编译“java实验.rar”

(1)AB.java

输出：

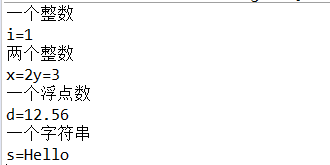


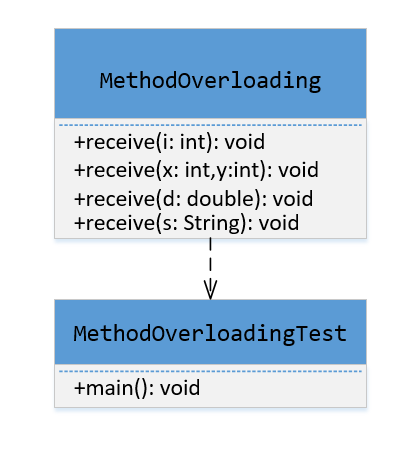
UML图：

(2)HelloWorld.java

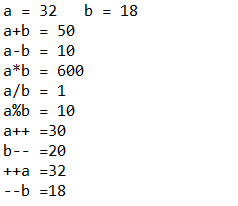


(3)MethodOverloadingTest.java

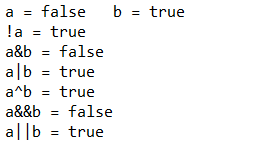




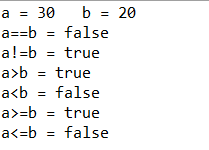
(4)NumberOper.java



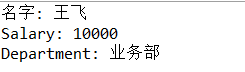
(5)TestLogic.java



(6)TestRelation.java



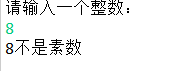
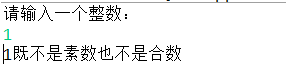
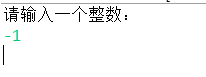
(7)UseConstruct.java





2.判断素数

一开始写程序的时候下意识的排除了1，因为在大家的认知里1是我们不会主动去判别它是否是素数的一个数字，因为我们自己知道。但是写程序的时候这就是一个bug，加了一个判断之后就可以让用户知道。

**package** lab1;

**import** java.util.Scanner;

**public** **class** IsPrime {

**public** **static** **void** main(String[] args) {

//输入一个整数

Scanner input=**new** Scanner(System.***in***);

System.***out***.println("请输入一个整数：");

**int** p = input.nextInt();

**if**(p<=0) {

System.*exit*(0);

}

**else** **if**(p==1) {

System.***out***.println(p+"既不是素数也不是合数");

}

**else** {

**if**(*isPrimeNumber*(p)) {

System.***out***.println(p+"是素数");

}

**else** {

System.***out***.println(p+"不是素数");

}

}

}

**public** **static** **boolean** isPrimeNumber(**int** number) {

//Judge the number

**for**(**int** div=2;(div\*div)<=number;div++) {

**if** (number % div == 0) {

**return** **false**;

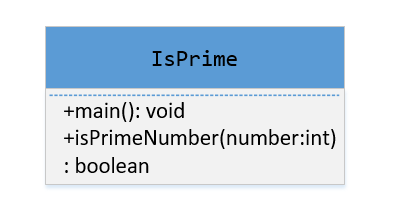
}

}

**return** **true**;

}

}



3.定义二维坐标里的点类DPoint

定义坐标里的点类，有一个困惑是移动点，写移动点的函数时一直无法定义点的上下左右方向。因为在我的认知里电脑上的坐标是反的，如果需要移动的话坐标值怎么变化。Jiej 方法是直接加减就好，不用管坐标反，如果是反的，也是y变成-y，即y轴坐标对调。

**package** lab1;

**public** **class** DPoint {

**private** **double** x;

**private** **double** y;

**public** DPoint(){

}

**public** DPoint(**double** x,**double** y){

**this**.x=x;

**this**.y=y;

}

**public** **double** getX() {

**return** **this**.x;

}

**public** **double** getY() {

**return** **this**.y;

}

**public** **void** setX(**double** a) {

**this**.x=a;

}

**public** **void** setY(**double** b) {

**this**.y=b;

}

//计算两点之间的距离

**public** **double** distance(**double** x1,**double** y1,**double** x2,**double** y2) {

**return** Math.*pow*((Math.*sqrt*(Math.*abs*(x1-x2))+Math.*sqrt*(Math.*abs*(y1-y2))),2);

}

//移动点的坐标

**public** DPoint direction(String d,DPoint p) {

**if**(d.charAt(0)=='L') {//左移

p.x=p.x-Integer.*parseInt*(d.substring(1));

}

**if**(d.charAt(0)=='R') {//右移

p.x=p.x+Integer.*parseInt*(d.substring(1));

}

**if**(d.charAt(0)=='U') {//上移

p.y=p.y+Integer.*parseInt*(d.substring(1));

}

**if**(d.charAt(0)=='D') {//下移

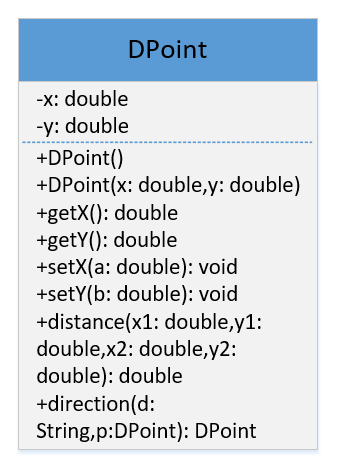
p.y=p.y-Integer.*parseInt*(d.substring(1));

}

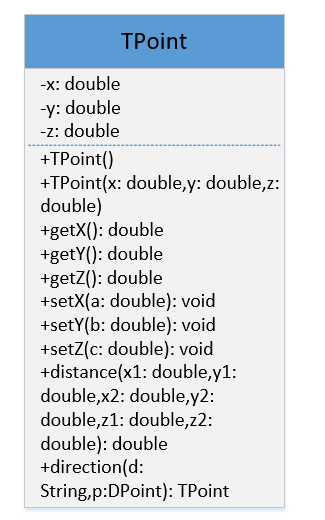
**return** p;

}

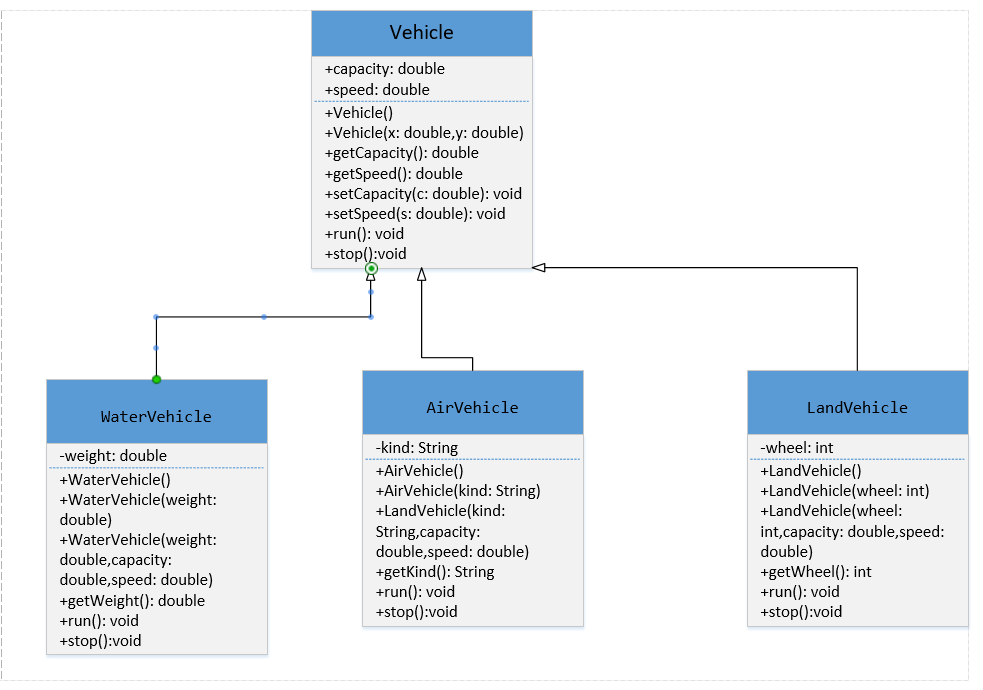
}



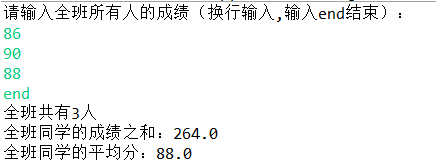
4.定义三维坐标里的点TPoint

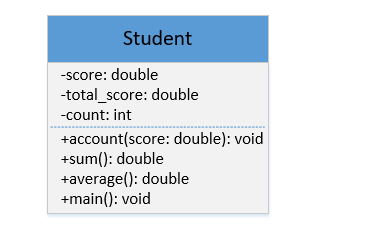


5.交通工具类



6.Student类





**package** lab1;

**import** java.util.Scanner;

**public** **class** Student {

**private** **double** score;//分数

**private** **static** **double** *total\_score*;//总分

**private** **static** **int** *count*;//人数

//Student(){}

**public** **void** account(**double** score) {

**this**.score=score;

**this**.*total\_score*+=score;

**this**.*count*+=1;

}

**public** **static** **double** sum() {

**return** *total\_score*;

}

**public** **static** **double** average() {

**return** *total\_score*/*count*;

}

**public** **static** **void** main(String[] args) {

Student student=**new** Student();

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("请输入全班所有人的成绩（换行输入,输入end结束）：");

**while**(!input.hasNext("end")) {//输入end完成成绩输入

**double** score=input.nextDouble();

student.account(score);

}

System.***out***.println("全班共有"+student.*count*+"人");

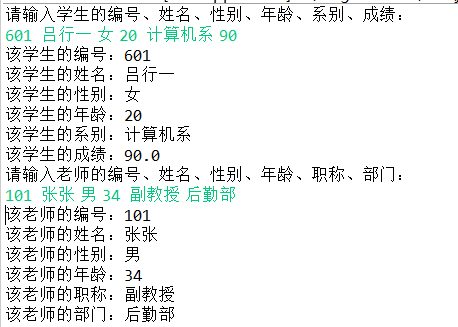
System.***out***.println("全班同学的成绩之和："+student.*sum*());

System.***out***.println("全班同学的平均分："+student.*average*());

}

}

7. 学生和教师的数据输入和显示程序





8.数组运用

修改代码为：

**package** lab1;

**public** **class** MyArray{

**protected** **int**[] alist;

**protected** **int** length;

MyArray(){

}

**public** MyArray(**int** leng)

{

**if**(leng<=0)

{

System.***out***.println("error length");

System.*exit*(1);

}

alist=**new** **int** [leng];

length=leng;

//System.out.println("MyArray 类对象已创建。");

}

**public** **void** init()

{

**int** i;

**for**(i=0;i<length;i++)

alist[i] = length - i;

}

**public** **void** display()

{

**int** i;

System.***out***.print("MyArray中的元素为：");

**for**(i=0;i<length;i++)

System.***out***.print(alist[i]+"\t");

System.***out***.println();

}

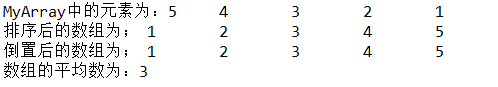
}





“\t d”在一开始出现错误，数组初始化都是d或者如上图所示

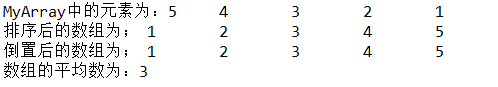
编译结果如下：是直接调用MyArray初始化数组





JAVA不支持多继承，因此不能同时实现一个类复用其它几个类的情况，但是可以复用类的代码，但需要改类名，然后将类都放在一个类中。

运行结果如下：





参考代码：

**package** lab1;

**public** **class** NewArray {

**public** **static** **void** main(String[] args)

{

MyArray a = **new** MyArray(5);

a.init();

a.display();

SArray aa=**new** SArray();

aa.sortArray();

System.***out***.println();

RArray aaa=**new** RArray();

aaa.reArray();

System.***out***.println();

Ave aaaa=**new** Ave();

aaaa.average();

}

}

**class** SArray **extends** MyArray{

MyArray a = **new** MyArray(5);

**public** **void** sortArray() {

a.init();

**for** (**int** i = 0; i < a.length -1 ; i++) {

//这个是和第一个数的比较的数

**for** (**int** j = i+1; j < a.length; j++) {

//定义一个临时的变量，用来交换变量

**int** temp ;

**if**(a.alist[i]>a.alist[j]){

temp = a.alist[i];

a.alist[i] = a.alist[j];

a.alist[j] = temp;

}

}

}

System.***out***.print("排序后的数组为；");

**for**(**int** i=0;i<a.length;i++)

System.***out***.print("\t"+a.alist[i]);

}

}

**class** RArray **extends** MyArray{

MyArray a = **new** MyArray(5);

**public** **void** reArray() {

a.init();

**for**(**int** i=0;i < a.length / 2;i++){

**int** temp = a.alist[i];

a.alist[i] = a.alist[a.length - 1 -i];

a.alist[a.length - 1 -i] = temp;

}

System.***out***.print("倒置后的数组为；");

**for**(**int** i = 0;i < a.length;i++){

System.***out***.print("\t"+a.alist[i]);

}

}

}

**class** Ave **extends** MyArray{

MyArray a = **new** MyArray(5);

**public** **void** average() {

a.init();

**int** sum=0;

**int** ave=0;

**for**(**int** i=0;i<a.length;i++) {

sum+=a.alist[i];

}

ave=sum/a.length;

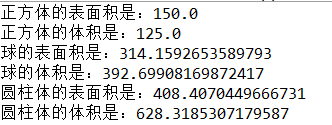
System.***out***.println("数组的平均数为："+ave);

}

}

9.Container类

半径为5，圆柱体的高为8





1. 思考讨论题
2. 在报告中回答以下问题：

定义类时，那些成员数据和成员函数是必须的，哪些不是？public、protected、private分别表示什么意思？能否对类的成员数据在定义时初始化？

答：成员函数和成员数据都不是必须的，可以为空类。Public表示该成员数据或成员函数是公有的，其它所有类都可以访问；protected表示该成员数据或成员方法只可以被其子类访问；private表示私有的，即该成员数据或成员函数只可以被自身调用。

2、在主调函数中，能否用：“对象名.私有成员名”这样的方式来访问对象的私有成员？为什么？在类的成员函数中可以吗？

答：不可以，因为对象的私有成员只能被该对象自己访问。在类的成员函数中可以。

3、在类的定义中是否须有构造函数？构造函数可以重载吗？

答：构造函数不是必须的额，如果类本身不具备构造函数，在创建该对象时会默认调用该类的无参构造函数。构造函数不能重载。

4、构造函数与析构函数是否需要一一对应？

答：不需要，只要你创建了对象就会调用构造函数 包括你传参数时，参数为对象也会调用，当程序运行遇到大括号时就会隐式调用析构函数，并不一定两者次数会一样

5、拷贝函数与应用场合是什么？

答：当用类的一个对象去初始化该类的另一个对象时；如果函数的形式参数是类的对象，调用函数时，进行形参和实参结合时；如果函数的返回值是类的对象，函数执行完成返回调用者时。

6、 在静态成员函数中是否可以直接引用静态变量？可以直接引用非静态变量吗？

答：静态成员函数中可以可以直接引用静态变量，不可以直接引用非静态变量。

7、在静态成员函数中，引用非静态成员时需要通过对象吗？通过成员限定可以吗？

答：需要通过对象。不可以通过成员限定。

8、在构造函数中是否可以对类中静态的数据成员进行初始化？初始化的静态数据成员是否可以将该值保持到下一次被改变为止？

答：可以；可以。

1. 参考资料

九、附加信息

以上所有完整代码已上传至Github，可以查看。网址如下：

<https://github.com/inocentnine/JAVA-Object>

<https://github.com/inocentnine/JAVA-Object/tree/master/JAVA-Object>