

**计算机与信息 学院实验报告**

|  |  |  |
| --- | --- | --- |
| 实验课程： | 面向对象程序设计实验 | |
| 实验编号： | 实验一 | |
| 实验名称： | 面向对象编程入门 | |
| 实验人员： | 学号 | 22111302041 |
| 姓名 | 高蝶 |
| 班级 | 计算机科学与技术1班 |
| 指导教师： | 陶涛 | |
| 实验室： | 2号实验楼202 | |
| 实验日期： | 2023.2.25 | |

# 实验目的

1. 了解面向对象程序设计的一般方法；

2. 掌握使用eclipse集成环境进行Java程序的编写、编译运行、调试；

3. 掌握Java语言中基本数据类型的输入、输出方法

4. 初步理解类、对象、方法的定义和调用方式

# 实验要求

1. 能够正确的编写、编译运行Java程序；

2. 正确实现int型、double型数据的输入、输出

3. 能够根据具体问题设计类的属性和方法，并进行测试

# 实验内容

1.验证型实验

（1）仔细读下面的Java语言源程序，分析程序的运行结果，再上机运行程序进行验证，如果有错，弄清楚出错原因。

public class LX1\_1

{

public static void main(String args[])

{

char x,y,z,w;

x='K';

y='\\';

z='\127';

w='\u0041';

System.out.println(x);

System.out.println(y);

System.out.println(z+","+w);

for(int i=20302;i<=20322;i++) {

System.out.print((char)i);

}

}

}

（2）仔细读下面的Java语言源程序，分析程序的运行结果，再上机运行程序进行验证，如果有错，弄清楚出错原因。

public class LX1\_2

{

public static void main(String args[])

{

boolean x,y,z;

int a=12,b=24;

x=(a>b);

y=(a!=b);

z=(a+b==36);

System.out.println("x="+x);

System.out.println("y="+y);

System.out.println("z="+z);

System.out.println("a+b= "+a+b);

System.out.println("a+b= "+(a+b));

}

}

2.应用型实验

（1）以下Java程序的功能是：输入一个大于2的自然数，判断其是否为素数，是素数输出YES，不是素数输出NO。请不改动程序原有结构和代码，将程序补充完整，并得到正确运行结果。

输入：5

输出：YES

**import** java.util.Scanner;

**public** **class** PrimeTest {

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

Scanner reader=**new** Scanner(System.***in***);//定义读取数据对象reader

**int** n;

n=reader.nextInt(); //读取一个整数

**if**(*isPrimeNumber*(n)==1)

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

**else**

System.***out***.println("NO");

}

**static** **int** isPrimeNumber (**int** n){

//此处补充代码

}

}

（2）将一个正整数分解为其质因数的积。例如：输入90，输出90=2\*3\*3\*5。

请不改动程序原有结构和代码，将程序补充完整，并得到正确运行结果。

输入：90

输出：90=2\*3\*3\*5

**import** java.util.Scanner;

**public** **class** PrimeFactorTest {

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

Scanner reader=**new** Scanner(System.***in***);//定义读取数据对象reader

**int** n;

n=reader.nextInt(); //读取一个整数

PrimeFactorTest pf=**new** PrimeFactorTest();

pf.printPrimeFactor(n);

}

**void** printPrimeFactor(**int** n){ //求整数n的质因子，按按要求输出

//此处补充代码

}

**int** isPrimeNumber(**int** n){

//此处补充代码

}

}

3. 设计型实验

（1）以下程序实现分数的加法运算，请在此程序的基础上，实现分数的减法运算、乘法运算、除法运算，并进行测试。输入要求：整数，不考虑分母为0情况；输出要求：最简分数，分母不为1，分子不为0，分母不为0

输入：1 4 1 2

输出：

3/4

-1/4

1/8

1/2

**import java.util.Scanner;**

**public** **class** FractionTest {

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

Fraction f1=**new** Fraction(1,4);

Fraction f2=**new** Fraction(1,4);

Fraction f3=f1.add(f2);

f3.print();

}

}

**class** Fraction{

**int** numerator;

**int** denominator;

Fraction(){}

Fraction(**int** n,**int** d){

numerator=n;

denominator=d;

}

Fraction add(Fraction f2){

Fraction f3=**new** Fraction();

f3.numerator=numerator\*f2.denominator+denominator\*f2.numerator;

f3.denominator=denominator\*f2.denominator;

**return** f3;

}

Fraction sub(Fraction f2){ //分数的减法运算

}

Fraction multi(Fraction f2){ //分数的乘法运算

}

Fraction divide(Fraction f2){ //分数的除法运算

}

**void** print(){

reduction();

//以下输出需要增加判断条件，修改成符合题目要求的形式

System.***out***.println(numerator+"/"+denominator);

}

**void** reduction(){ //对分数进行约分，化成最简分数

//请补充完整。此处直接对numerator和denominator进行约分，本函数不需要输入、输出

//求分子分母的最大公约数a

**int** a;

numerator=numerator/a;

denominator=denominator/a;

}

}

（2）参考上题分数程序，并结合以下程序程序代码，实现复数的加法、减法、乘法运算，并进行测试。

输入：

1 2

3 4

输出：

4+6i

-2-2i

-5+10i

**import** java.util.Scanner;

**public** **class** ComplexTest {

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

Scanner reader=**new** Scanner(System.***in***);//定义读取数据对象reader

**int** r1,v1,r2,v2;

r1=reader.nextInt(); //读取第一个复数的实部

v1=reader.nextInt(); //读取第一个复数的虚部

}

}

**class** Complex{

**int** real;

**int** virtual;

Complex(){}

Complex(**int** r,**int** v){

rear=r;

virtual=v;

}

Complex add(Complex f2){

}

Complex sub(Complex f2){

}

Complex multi(Complex f2){

}

**void** print(){

//以下输出需要修改成符合题目要求的形式

System.***out***.println(real+"+"+virtual+"i");

}

}

（3）参考上述两个程序，请你设计一个点类（Point），其属性包括横坐标和纵坐标，方法包括计算点所在象限getQuadrant、两点之间距离calDistance（小数部分四舍五入）。在PointTest类中对Point类的两个方法进行测试。

输入：

1 2

3 4

输出：

1 1 3

# 程序清单

2.应用型实验

（1）

**测试数据：**

第1组：5

第2组：6

第3组：11

**程序源码：**

**import** java.util.Scanner;

**public** **class** Main {

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

Scanner reader=**new** Scanner(System.***in***);//定义读取数据对象reader

**int** n;

n=reader.nextInt(); //读取一个整数

**if**(*isPrimeNumber*(n)==1)

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

**else**

System.***out***.println("NO");

}

**static** **int** isPrimeNumber (**int** n){

**int** i;

**if**(n<=1) **return** 0;

**for**(i=2;i<n;i++) {

**if**(n%i==0) **return** 0;

}

**return** 1;

}

}

（2）

**测试数据：**

第1组：90

第2组：2

第3组：10

**程序源码：**

**import** java.util.Scanner;

**public** **class** Main {

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

Scanner reader=**new** Scanner(System.***in***);//定义读取数据对象reader

**int** n;

n=reader.nextInt(); //读取一个整数

Main pf=**new** Main();

pf.printPrimeFactor(n);

}

**void** printPrimeFactor(**int** n){ //求整数n的质因子，按按要求输出

**int** i,j,ntemp=n;

System.***out***.printf("%d=", n);

**for**(i=2;i<=n;i++) {

**if**(ntemp%i==0&&isPrimeNumber(i)==1) {

**while**(ntemp%i==0) {

**if**(ntemp/i==1) System.***out***.printf("%d", i);

**else** System.***out***.printf("%d\*", i);

ntemp/=i;

}

}**else** **continue**;

}

}

**int** isPrimeNumber(**int** n){

**if**(n<=1) **return** 0;

**int** i;

**for**(i=2;i<n;i++) {

**if**(n%i==0) **return** 0;

}

**return** 1;

}

}

3.设计型实验

（1）

**测试数据：**

第1组：1 4 1 2

第2组：1 1 1 1

第3组：1 3 2 3

**程序源码：**

**import** java.util.Scanner;

**public** **class** Main {

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

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

**int** n1,n2,n3,n4;

n1=reader.nextInt();

n2=reader.nextInt();

n3=reader.nextInt();

n4=reader.nextInt();

Fraction f1=**new** Fraction(n1,n2);

Fraction f2=**new** Fraction(n3,n4);

Fraction f3=f1.add(f2);

f3.print();

Fraction f4=f1.sub(f2);

f4.print();

Fraction f5=f1.multi(f2);

f5.print();

Fraction f6=f1.divide(f2);

f6.print();

}

}

**class** Fraction{

**int** numerator,flag=0,flag1=0;

**int** denominator;

Fraction(){}

Fraction(**int** n,**int** d){

numerator=n;

denominator=d;

}

Fraction add(Fraction f2){

Fraction f3=**new** Fraction();

f3.numerator=numerator\*f2.denominator+denominator\*f2.numerator;

f3.denominator=denominator\*f2.denominator;

**return** f3;

}

Fraction sub(Fraction f2){ //分数的减法运算

Fraction f4=**new** Fraction();

f4.numerator=numerator\*f2.denominator-denominator\*f2.numerator;

f4.denominator=denominator\*f2.denominator;

**return** f4;

}

Fraction multi(Fraction f2){

Fraction f5=**new** Fraction();

f5.numerator=numerator\*f2.numerator;

f5.denominator=denominator\*f2.denominator;

**return** f5;

}

Fraction divide(Fraction f2){ //分数的除法运算

Fraction f6=**new** Fraction();

f6.numerator=numerator\*f2.denominator;

f6.denominator=denominator\*f2.numerator;

**return** f6;

}

**void** print(){

reduction();

**if**(flag1==1)System.***out***.printf("0\n");

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

**if**(denominator==1)System.***out***.println("-"+numerator);

**else** System.***out***.println("-"+numerator+"/"+denominator);

}

**else** {

**if**(denominator==1)System.***out***.println(numerator);

**else** System.***out***.println(numerator+"/"+denominator);

}

}

**void** reduction(){

**if**(numerator<0) {

numerator=-numerator;flag=1;

**int** r,m=denominator,n=numerator,a;

r=m%n;

**while**(r!=0) {

m=n;

n=r;

r=m%n;

}

a=n;

numerator=numerator/a;

denominator=denominator/a;}

**else** **if**(numerator>0){

**int** r,m=denominator,n=numerator,a;

r=m%n;

**while**(r!=0) {

m=n;

n=r;

r=m%n;

}

a=n;

numerator=numerator/a;

denominator=denominator/a;

}

**else** **if**(numerator==0) flag1=1;

}

}

（2）

**测试数据：**

第1组：1 2 3 4

第2组：1 1 2 2

第3组：3 2 4 1

**程序源码：**

**import** java.util.Scanner;

**public** **class** Main {

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

Scanner reader=**new** Scanner(System.***in***);//定义读取数据对象reader

**int** r1,v1,r2,v2;

r1=reader.nextInt(); //读取第一个复数的实部

v1=reader.nextInt(); //读取第一个复数的虚部

r2=reader.nextInt(); //读取第一个复数的实部

v2=reader.nextInt();

Complex f1=**new** Complex(r1,v1);

Complex f2=**new** Complex(r2,v2);

Complex f3=f1.add(f2);

f3.print();

Complex f4=f1.sub(f2);

f4.print();

Complex f5=f1.multi(f2);

f5.print();

}

}

**class** Complex{

**int** real;

**int** virtual;

Complex(){}

Complex(**int** r,**int** v){

real=r;

virtual=v;

}

Complex add(Complex f2){

Complex f3=**new** Complex();

f3.real=real+f2.real;

f3.virtual=virtual+f2.virtual;

**return** f3;

}

Complex sub(Complex f2){

Complex f4=**new** Complex();

f4.real=real-f2.real;

f4.virtual=virtual-f2.virtual;

**return** f4;

}

Complex multi(Complex f2){

Complex f5=**new** Complex();

f5.real=real\*f2.real-virtual\*f2.virtual;

f5.virtual=virtual\*f2.real+real\*f2.virtual;

**return** f5;

}

**void** print(){

**if**(real==0&&virtual==0) System.***out***.println("0");

**else** **if**(real==0&&virtual!=0) {

System.***out***.println(virtual+"i");

}**else** **if**(real!=0&&virtual==0) {

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

}**else** {

**if**(virtual>0) {

System.***out***.println(real+"+"+virtual+"i");

}**else** System.***out***.println(real+(virtual+"i"));

}

}

}

（3）

**测试数据：**

第1组：1 2 3 4

第2组：-1 2 3 -1

第3组：2 3 -1 -1

**程序源码：**

**import** java.util.Scanner;

**public** **class** Main {

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

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

**double** x1,y1,x2,y2;

x1=reader.nextDouble();

y1=reader.nextDouble();

x2=reader.nextDouble();

y2=reader.nextDouble();

Point f1=**new** Point(x1,y1);

Point f2=**new** Point(x2,y2);

f1.getQuadrant();

f2.getQuadrant();

f1.calDistance(f2);

}

}

**class** Point{

**double** x,y;

Point (){}

Point (**double** x1,**double** y1){

x=x1;

y=y1;

}

**void** getQuadrant(){

**if**(x>0&&y>0) System.***out***.printf("1 ");

**if**(x>0&&y<0) System.***out***.printf("4 ");

**if**(x<0&&y>0) System.***out***.printf("2 ");

**if**(x<0&&y<0) System.***out***.printf("3 ");

}

**void** calDistance(Point f){

**double** ans;

ans=Math.*sqrt*((x-f.x)\*(x-f.x)+(y-f.y)\*(y-f.y));

**int** res;

res=(**int**)Math.*round*(ans);

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

}

}

# 实验结果及分析

1.验证型实验

（1）结果：K

\

W,A

低住佐佑佒体佔何佖佗佘余佚佛作佝佞佟你佡佢

1. 结果：x=0 y=1 z=1 a+b=1224 a+b=36

2.应用型实验

（1）

各个类、方法的作用：Main 负责完成数据的输入，类的调用，数据的输出

isPrimeNumer 负责1判断一个数是否为素数

运行结果：1:NO 2:YES 3:NO

（2）

各个类、方法的作用：main 负责数据输入，调用方法；printPrimeFactor负责求数n的质因子，按要求输出；isPrimeNumber负责判断数是否为素数。

运行结果：1:90=2\*3\*3\*5

2:2=2

3:10=2\*5

3.设计型实验

（1）

各个类、方法的作用：main负责输入调用输出；Fraction类里进行分数的加减乘除运算，结果的规范化处理（分子分母约至最简）

运行结果：1：3/4 -1/4 1/8 1/2

2: 2 0 1 1

3: 1 -1/3 2/9 1/2

（2）

各个类、方法的作用：main负责输入调用输出；Complex负责复数的加减乘运算，输出；

运行结果：1：4+6i -2-2i -5+10i

2：3+3i -1-1i 4i

3：7+3i -1+1i 10+11i

（3）

各个类、方法的作用：main负责输入调用输出；Point类里判断象限和计算两点间距离

运行结果：1： 1 1 3

2：2 4 5

3：1 3 5

# 实验小结

[本次实验的反思、收获和体会：]

反思：1：eclipse 还不太会使用，写起代码还不太习惯

2：JAVA的写代码的流程有些模糊，还不能独立地写出来，很多基础的语法还不知道

3：实验报告与以往不大相同，写起来不太顺手

收获：1：第一次写java，我克服了重重困难写出了题目，对其有了一点点了解

2：自己切身体验到遇到问题自己思考不太会之后上网对我想了解的东西进行搜索，学习，并及时运用解决问题的快乐

体会：学习是件自己的事情，自己要对自己负责。java的基础语法有必要平时多看看，学习。平时注重锻炼自己思考问题，解决问题，总结归纳的能力。逐步培养对java的学习兴趣。新学期新知识我要认真对待，就像老师说的那样，课上不懂的知识及时记在小本子上，课后及时去上网搜索弄明白。刚开始会遇到很多问题，不应该忽视，应及时解决。