



山西工学院
SHANXI COLLEGE OF TECHNOLOGY

实验报告

学院名称： 信息产业学院

专业名称： 计算机科学与技术

课程名称： 面向对象程序设计

班 级： 211060104 学号： 21106010413

学生姓名： 刘钟泽

指导老师： 任少斌

2022 年 9 月 11 日

学生姓名	刘钟泽	学号	21106010413	实验成绩	
实验项目名称	系统安装配置				
实验地点	机房	实验日期	2022 年 9 月 11 日		
<div>一、实验目的和要求</div> <p>熟悉系统的安装与配置方式。了解 Java 语言的基本规则</p>					
<div>二、实验内容和原理</div> <p>根据自己的计算机配置情况，整理Java系统的安装技术文档</p> <div><div>(1)、JDK安装过程；</div><div>(2)、系统配置方式；</div><div>(3)、其它IDE的安装(Notepad++, Eclipse等内容, 选择自己使用的版本)；</div><div>(4)、通过Hello World代码验证系统是否安装成功；</div><div>(5)、说明一下Java编程的基本规则（文件名，文件结构）</div><div>(6)、要求有插图内容。</div></div>					

三、程序代码（要求有注释说明）

<https://www.oracle.com/technetwork/java/javase/downloads> JDK 下载路径

变量名: JAVA_HOME

变量值: C:\Program Files\Java\jdk1.8.0_152

变量名 CLASSPATH.

变量值:

;%JAVA_HOME%\lib;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;

变量名: Path

变量值: 将 %JAVA_HOME%\bin;

```
public class Test1 {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

四、实验结果截屏图

```
PS C:\Users\Administrator\Desktop\java实验> javac Test1.java  
PS C:\Users\Administrator\Desktop\java实验> java Test1.java  
Hello World
```

» 此电脑 » 本地磁盘 (C:) » Program Files (x86) » Notepad++

名称	修改日期	类型	大小
localization	2022/9/2 星期五 14:35	文件夹	
plugins	2022/9/2 星期五 14:35	文件夹	
themes	2022/9/2 星期五 14:35	文件夹	
updater	2022/9/2 星期五 14:35	文件夹	
user.manual	2022/9/2 星期五 14:35	文件夹	
change.log	2014/9/8 星期一 0:34	文本文档	1 KB
config.model.xml	2014/2/18 星期二 8:18	XML 文档	5 KB
contextMenu.xml	2013/10/15 星期二 20:01	XML 文档	4 KB
functionList.xml	2014/8/15 星期五 8:01	XML 文档	11 KB
langs.model.xml	2014/6/8 星期日 9:42	XML 文档	112 KB
license.txt	2012/4/18 星期三 7:18	文本文档	17 KB
notepad++.exe	2014/9/8 星期一 5:50	应用程序	2,348 KB
NppShell_06.dll	2014/5/12 星期一 17:49	应用程序扩展	218 KB
readme.txt	2013/9/29 星期日 1:13	文本文档	2 KB
SciLexer.dll	2013/9/15 星期日 17:38	应用程序扩展	1,028 KB
shortcuts.xml	2013/12/7 星期六 8:52	XML 文档	2 KB
stylers.model.xml	2014/7/21 星期一 17:14	XML 文档	97 KB
uninstall.exe	2022/9/2 星期五 14:35	应用程序	273 KB

» 此电脑 » 下载 » Compressed » eclipse

名称	修改日期	类型	大小
configuration	2022/10/3 星期一 0:08	文件夹	
dropins	2020/12/10 星期四 11:33	文件夹	
features	2020/12/10 星期四 11:33	文件夹	
p2	2022/11/3 星期四 12:10	文件夹	
plugins	2020/12/10 星期四 11:33	文件夹	
readme	2020/12/10 星期四 11:33	文件夹	
.eclipseproduct	2020/12/2 星期三 23:06	ECLIPSEPRODUCT ...	1 KB
artifacts.xml	2020/12/10 星期四 11:33	XML 文档	268 KB
eclipse.exe	2020/12/10 星期四 11:36	应用程序	417 KB
eclipse.ini	2020/12/10 星期四 11:33	配置设置	1 KB
eclipsesec.exe	2020/12/10 星期四 11:36	应用程序	129 KB

五、实验结果与分析

1、Java 文件命名：

(1) java 程序是由类组成的；

(2) java 应用程序必须有一个包含 main 方法的 public 类，main 方法是程序的入口；

2、java 源文件的命名规则：

如果有 public 类，那么源文件名就是 public 类名(一个文件中只能有一个 public 类)

类名命名规则：

类的名字必须由大写字母开头而单词中的其他字母均为小写；如果类名称由多个单词组成，则每个单词的首字母均应为大写，如果类名称中包含单词缩写，则这个所写词的每个字母均应大写，命名类时应尽量选择名词, 如果没有 public 类，就和遵循文件名命名规则



山西工学院
SHANXI COLLEGE OF TECHNOLOGY

实验报告

学院名称： 信息产业学院

专业名称： 计算机科学与技术

课程名称： 面向对象程序设计

班 级： 211060104 学号： 21106010413

学生姓名： 刘钟泽

指导老师： 任少斌

2022 年 9 月 18 日

学生姓名	刘钟泽	学号	21106010413	实验成绩	
实验项目名称	数据类型分析				
实验地点	机房	实验日期	2022 年 9 月 18 日		
<div>一、实验目的和要求</div> <div>Java 语言的数据类型比较特殊，分成基础数据类型与引用数据类型，其中基础数据类型四类八种。Java 语言是强类型的编程语言，在编程中出现数据类型不符的时候，系统无法编译程序。本实验的重点是通过练习掌握基础数据类型的使用方式。</div>					

二、实验内容和原理

Java语言分成基础数据类型与引用数据类型，本实验的重点是分析基础类型中出现的情况与处理方式。参考下面的代码分析错误原因与解决方法。要求说明原因。

通过测试下面的代码，分析数据类型的使用方法

```
public class Test
{
    public static void main(String[] args)
    {
        int i,j;
        float f1=0.1;
        float f2=123;
        double d1=2e20;
        double d2=124;
        byte b1=1,b2=2;
        byte b3=129;
        j=j+10;
        i=i/10;
        i=i*0.1;
        char c1='a', c2=125;
        byte b=b1-b2;
        char c=c1+c2-1;
        float f3=f1+f2;
        float f4=f1+f2*0.1;
        double d=d1*i+j;
        float f=d1*5+d2;
    }
}
```

要求修改并说明原因，增加代码验证修改的结果。

三、程序代码（要求有注释说明）

```
public class Test
{
    public static void main(String[] args)
    {
        double i=1;
        int j=1;
        float f1=0.1F; // 不能从 double 双精度类型转换到 float 单精度类型。
        //浮点常量的默认类型是 double，改成 float 类型后面要加 F。
        //或者可以在前面加（float）
        float f2=123;
        double d1=2e20;
        double d2=124;
        byte b1=1,b2=2;
        int b3= 129; //由于 byte 表示范围实在是太小了
        //所以 byte 在变量的情况下会自动提升到 int，而 int 不会自动提升。
        //所以可以更改为 int 或者强制转化为 byte,(byte)
        j=j+10; //j 未初始化
        i=i/10; //i 未初始化
        i=i*0.1; //i 为 int 类型（整型）不能直接将其转化为浮点型
        //可在定义时更改为 double 类型，也可直接强制类型转换
        char c1='a', c2=125;
        int b=b1-b2; //byte 由于变量未知导致无法知道是否超过其表示范围，而会自动提升到 int
        //所以可以改为 int 类型，或者强制性转为 byte
        char c=(char) (c1+c2-1);
        float f3=f1+f2;
        float f4=(float) (f1+f2*0.1);
        //java 中默认浮点型为 double 类型
        //可强制转化为 float 或者修改定义为 double
        double d=d1*i+j;
        float f=(float) (d1*5+d2);
        //java 中默认浮点型为 double 类型
        //可强制转化为 float 或者修改定义为 double
        System.out.printf("b3=%d\n",b3);
        System.out.printf("b=%d\n",b);
        System.out.printf("c=%s\n",c);
        System.out.printf("f3=%.3f\n",f3);
        System.out.printf("f4=%.3f\n",f4);
        System.out.printf("d=%.3f\n",d);
        System.out.printf("f=%.3f\n",f);
    }
}
```

四、实验结果截屏图

问题 输出 调试控制台 终端 JUPYTER: VARIABLES

```
PS C:\Users\Administrator\Desktop\java实验> c:;; cd 'C:\Users\Administrator\AppData\Roaming\Code\User\workspaceStorage\9c306b3=129
b=-1
c=?
f3=123.100
f4=12.400
d=2000000000000000510.000
f=1000000020040877300000.000
PS C:\Users\Administrator\Desktop\java实验> █
```

```
PS C:\Users\Administrator\Desktop\java实验> & cd 'C:\Users\Administrator\AppData\Roaming\Code\User\workspaceStorage\9c306b3=129
b=-1
c=221
f3=123.100
f4=12.400
d=2000000000000000510.000
f=1000000020040877300000.000
PS C:\Users\Administrator\Desktop\java实验> █
```

五、实验结果与分析

数据类型的错误多出现在类型的转化中，在转化中多出现范围不一致导致的数据丢失。

在 java 中默认的浮点型为 double 类型，在使用过程中 float* 一个浮点型就会导致类型的转换。

char 类型的范围为 -128~127, 所以在计算之中出现 char 类型的计算时需要考虑是否超过范围。

byte 表示范围为-128~127，一般的数学计算都会超过它，所以

byte 在变量的情况下会自动提升到 int，而 int 不会自动提升。

bool: 存储值 true 或 false。

char: 通常是一个字符（八位）。这是一个整数类型。

int: 对机器而言，整数的最自然的大小。

float: 单精度浮点值。单精度是这样的格式，1 位符号，8 位指数，23 位小数。

double: 双精度浮点值。双精度是 1 位符号，11 位指数，52 位小数。



山西工學院
SHANXI COLLEGE OF TECHNOLOGY

实验报告

学院名称： 信息产业学院

专业名称： 计算机科学与技术

课程名称： 面向对象程序设计

班 级： 211060104 学号： 21106010413

学生姓名： 刘钟泽

指导老师： 任少斌

2022 年 9 月 25 日

学生姓名	刘钟泽	学号	21106010413	实验成绩	
实验项目名称	综合控制结构程序设计				
实验地点	机房	实验日期	2022 年 9 月 25 日		
<div>一、实验目的和要求</div> <p>任何编程语言，在具体实现功能时均使用三种典型的控制结构完成。但绝对不是单一的使用，绝大多数情况是综合运用这些规则进行设计。</p> <div>二、实验内容和原理</div> <div>1、基础的 IO 练习（使用 Scanner 类）</div> <div>2、选择结构的使用（使用不同的控制方式）</div> <div>3、循环结构的使用（使用不同的控制方式）</div> <p>编程实现“整数加法练习”的程序要求如下：</p> <div>1、能够实现连续若干个题（考虑如何实现）；</div> <div>2、计算过程中允许重新录入若干次答案（考虑如何实现）</div> <div>3、能够根据正确与错误统计分数；</div> <div>4、能够实现动态的评价方式；</div> <div>5、要求两个加数与答案均为两位整数（实验难点）；</div> <p>扩展要求：能够实现两位小数的加法运算</p>					

三、程序代码（要求有注释说明）

```
import java.util.Scanner;

public class Test3_1 {

    public static void main(String[] args) {
        System.out.printf("加法练习现在开始，共十题，每题有三次机会");
        try (Scanner sc = new Scanner(System.in)) {
            int x;
            int y;
            int answer;
            int score = 0;
            for (int i=0; i < 10; i++) { //进行 10 次循环表示共 10 题
                x = (int) (Math.random() * 80+10); //上限应为 90 下限为 10
                y = (int) (Math.random() * (90-x)+10); //+号前面为随机范围
                //0~（90-x）；+号后面为随机结束后增加数额

                System.out.printf("\033[H \033[2J"); //这里是实现清屏指令，
                //为了实现进行下一题的操作
                //\033[H: 将光标移动到屏幕或控制台的左上角。
                //\033[2J: 从光标处清屏到屏幕末尾。
                //此处的清屏时为了之后如需做图形页面更方便
                System.out.printf("请输入第%d 题%d+%d 的答案: ", i+1, x, y);
                for (int j=0; j < 3; j++) { //对应有三次机会
                    answer = sc.nextInt();
                    if (answer == x+y) { //检查答案正误
                        System.out.printf("回答正确，分数+10");

                        try { //循环暂停，便于读题
                            Thread.sleep(1000); //此处为毫秒，1000 毫秒=1 秒
                        } catch (InterruptedException ex) {
                            Thread.currentThread().interrupt();
                        }

                        score += 10;
                        break;
                    } else if (2-j!=0){
                        score -= 3 ;
                        System.out.printf("回答错误，你还有%d 次机会，请重新
                        输入: ", 2-j);
                    } else if (2-j ==0){
                        score -= 3 ;
                    }
                }
            }
        }
    }
}
```

```

        System.out.printf("你的三次回答都错误!!! \n 正确结果应为%d+%d=%d",x,y,x+y);
        try { //循环暂停, 便于读题
            Thread.sleep(5000); //此处为毫秒, 1000 毫秒=1 秒
        } catch (InterruptedException ex) {
            Thread.currentThread().interrupt();
        }

        System.out.printf("\033[H \033[2J");
    }

}

System.out.printf("\033[H \033[2J");
System.out.println("考试结束, 你的成绩为: " + score);
switch (score/10) {
    case 10 -> System.out.println("满分");
    case 9 -> System.out.println("优秀");
    case 8 -> System.out.println("良好");
    case 7, 6 -> System.out.println("及格");
    default -> System.out.println("不及格");
}
}
}
}

```

两位小数相加

```

import java.util.Scanner;

public class Test3_2 {

    public static void main(String[] args) {
        System.out.printf("加法练习现在开始, 共十题, 每题有三次机会");
        try (Scanner sc = new Scanner(System.in)) {
            int x;
            int y;
            double answer;
            int score = 0;
            for (int i=0; i < 10; i++) { //进行 10 次循环表示共 10 题

```

```

        x = (int) (Math.random() * 80+10); //上限应为 90 下限为 10
        y = (int) (Math.random() * (90-x)+10); //+号前面为随机范围
        0~(90-x); +号后面为随机结束后增加数额

        System.out.printf("\033[H \033[2J"); //这里是实现清屏指令，
        为了实现进行下一题的操作
        /\033[H: 将光标移动到屏幕或控制台的左上角。
        /\033[2J: 从光标处清屏到屏幕末尾。
        //此处的清屏时为了之后如需做图形页面更方便
        System.out.printf("请输入第%d 题 0.%d+0.%d 的答案: ", i+1, x,
y);

        for (int j=0; j < 3; j++) { //对应应有三次机会
            answer = sc.nextDouble();
            if (answer*100 == x+y) { //检查答案正误
                System.out.printf("回答正确，分数+10");

                try { //循环暂停，便于读题
                    Thread.sleep(1000); //此处为毫秒，1000 毫秒=1 秒
                } catch (InterruptedException ex) {
                    Thread.currentThread().interrupt();
                }

                score += 10;
                break;
            } else if (2-j!=0){
                score -= 3 ;
                System.out.printf("回答错误，你还有%d 次机会，请重新
输入: ", 2-j);
            } else if (2-j ==0){
                score -= 3 ;
                System.out.printf("你的三次回答都错误!!! \n 正确结
果应为 0.%d+0.%d=0.%d", x, y, x+y);
                try { //循环暂停，便于读题
                    Thread.sleep(5000); //此处为毫秒，1000 毫秒=1 秒
                } catch (InterruptedException ex) {
                    Thread.currentThread().interrupt();
                }

                System.out.printf("\033[H \033[2J");
            }

        }

    }

    System.out.printf("\033[H \033[2J");

```



```

        System.out.println("考试结束，你的成绩为：" + score);
        switch (score/10) {
            case 10 -> System.out.println("满分");
            case 9 -> System.out.println("优秀");
            case 8 -> System.out.println("良好");
            case 7, 6 -> System.out.println("及格");
            default -> System.out.println("不及格");
        }
    }
}
}
}

```

四、实验结果截屏图

```

PS C:\Users\Administrator\Desktop\java实验> javac Test3_1.java
PS C:\Users\Administrator\Desktop\java实验> java Test3_1.java
加法练习现在开始，共十题，每题有三次机会
  请输入第1题38+60的答案：98
回答正确，分数+10
  请输入第2题39+40的答案：79
回答正确，分数+10
  请输入第3题46+45的答案：
91
回答正确，分数+10
  请输入第4题75+11的答案：1
回答错误，你还有2次机会，请重新输入：2
回答错误，你还有1次机会，请重新输入：76
你的三次回答都错误!!!
正确结果应为75+11=86
  请输入第5题69+19的答案：88
回答正确，分数+10
  请输入第6题20+32的答案：52
回答正确，分数+10
  请输入第7题13+63的答案：76
回答正确，分数+10
  请输入第8题14+29的答案：43
回答正确，分数+10
  请输入第9题73+13的答案：86
回答正确，分数+10
  请输入第10题83+16的答案：99
回答正确，分数+10
  考试结束，你的成绩为：81
良好

```

```
PS C:\Users\Administrator\Desktop\java实验> javac Test3_2.java
PS C:\Users\Administrator\Desktop\java实验> java Test3_2.java
加法练习现在开始，共十题，每题有三次机会
  请输入第1题0.43+0.31的答案：0.74
回答正确，分数+10
  请输入第2题0.18+0.56的答案：
0.74
回答正确，分数+10
  请输入第3题0.77+0.20的答案：0.97
回答正确，分数+10
  请输入第4题0.63+0.21的答案：1
回答错误，你还有2次机会，请重新输入：1
回答错误，你还有1次机会，请重新输入：1
你的三次回答都错误！！！
正确结果应为0.63+0.21=0.84
  请输入第5题0.48+0.21的答案：0.69
回答正确，分数+10
  请输入第6题0.52+0.20的答案：0.1
回答错误，你还有2次机会，请重新输入：0.72
回答正确，分数+10
  请输入第7题0.72+0.25的答案：0.95
回答错误，你还有2次机会，请重新输入：0.97
回答正确，分数+10
  请输入第8题0.79+0.17的答案：0.96
回答正确，分数+10
  请输入第9题0.15+0.11的答案：0.26
回答正确，分数+10
  请输入第10题0.36+0.26的答案：0.62
回答正确，分数+10
  考试结束，你的成绩为：75
及格
```

五、实验结果与分析

本题难点两个加数与答案均为两位整数，需要限制每一个加数都小于 90 且大于 10，对应代码即为：

```
x = (int) (Math.random() * 80+10)
```

为了相加不大于 100，则需要第一个随机数的最大值减第二个随机数，对应代码即为：

```
y = (int) (Math.random() * (90-x)+10);
```

对于两位小数的计算，考虑浮点数精度导致计算结果误差的问

题，我选择使用输入结果*100，将这个结果与两个整数之和相比较，可避免因浮点数精度而输出错误的问题

```
answer = sc.nextDouble();  
if (answer*100 == x+y)
```

为了输出的整洁度，以及下一阶段制作图形界面的需要，故有以下代码：

```
System.out.printf("\033[H \033[2J");  
  
try {//循环暂停，便于读题  
Thread.sleep(1000);//此处为毫秒，1000 毫秒=1 秒  
} catch (InterruptedException ex) {  
Thread.currentThread().interrupt();  
}
```



山西工学院
SHANXI COLLEGE OF TECHNOLOGY

实验报告

学院名称： 信息产业学院
专业名称： 计算机科学与技术
课程名称： 面向对象程序设计
班 级： 211060104 学号： 21106010413
学生姓名： 刘钟泽
指导老师： 任少斌

2022 年 10 月 2 日

学生姓名	刘钟泽	学号	21106010413	实验成绩	
实验项目名称	类的定义与抽象实现				
实验地点	机房	实验日期	2022 年 10 月 2 日		
<div>一、实验目的和要求</div> <p>掌握类的类的方式、对象生成方式，整体结构的处理，如何封装信息，如何验证效果</p> <div>二、实验内容和原理</div> <p>参考上面的案例代码效果设计如下的类，注意通过测试类验证类的设计效果。</p> <p>1、设计一个空间“点”类，计算任意两点之间的距离，计算任意一点距离原点的距离。</p> <p>2、创建一个复数类 <code>complex</code>，以附属进行数学运算、复数具有如下格式：<code>RealPart+ImaginaryRart*I</code>，其中，<code>I</code> 为-1 的平方根。</p> <p>要求如下：</p> <p>① 利用浮点变量表示此类的私有数据。提供两个构造方法，一个用于此类声明的对象的初始化；一个为默认的空参构造方法。</p> <p>② 提供两复数加、减、乘的运算方法。</p> <p>③ 按格式(a,b)打印复数。其中 a 为实部，b 为虚部。</p>					

三、程序代码（要求有注释说明）

1、空间点问题

```
import java.util.Scanner;

public class Test4_1 {
    public static void main(String[] args) {
        System.out.printf("\033[H \033[2J");//这里是实现清屏指令
        //\033[H: 将光标移动到屏幕或控制台的左上角。
        //\033[2J: 从光标处清屏到屏幕末尾。
        //此处的清屏时为了之后如需做图形页面更方便
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入第一个点的空间坐标: ");
        Ponit p1= new Ponit(sc.nextDouble(), sc.nextDouble(),
sc.nextDouble());
        System.out.println("请输入第二个点的空间坐标: ");
        Ponit p2= new Ponit(sc.nextDouble(), sc.nextDouble(),
sc.nextDouble());
        System.out.printf("\033[H \033[2J");
        while (true) {
            System.out.println
            ("-----");
            System.out.println
            ("-----1、输出点 1 到原点的距离-----");
            System.out.println
            ("-----2、输出点 2 到原点的距离-----");
            System.out.println
            ("-----3、两点到原点的距离都输出-----");
            System.out.println
            ("-----0、退出系统(quit)-----");
            System.out.println
            ("-----");
            Scanner in = new Scanner(System.in);
            int choice = in.nextInt();
            switch (choice) { //用选择结构, 进行图形界面的功能选择
                case 0:
                    System.exit(0);
                    break;
                case 1:
                    System.out.println
                    ("-----1、输出点 1 到原点的距离-----");
                    p1.count1(p2);
```

```

        break;
    case 2:
        System.out.println
("-----2、输出点 2 到原点的距离-----");
        p2.count1(p1);
        break;
    case 3:
        System.out.println
("-----3、两点到原点的距离都输出-----");
        p1.count1(p2);
        p2.count1(p1);
        break;
    default:
        System.out.println("您的选择有误！请重新选择");
        break;
    }
}
}
}

}

class Ponit {
    private double x;//创建私有变量
    private double y;
    private double z;

    public Ponit(double a, double b, double c) {
        x = a;
        y = b;
        z = c;
    }
    public double getX() {
        return x;
    }
    public void setX(double a) {
        x = a;
    }
    public double getY() {
        return y;
    }
    public void setY(double b) {
        y = b;
    }
    public double getZ() {
        return z;
    }
}

```

```

    }
    public void setZ(double c) {
        z = c;
    }

    public void count1(Ponit p) {
        double m = Math.sqrt((x - p.x) * (x - p.x)
+ (y - p.y) * (y - p.y)+ (z - p.z) * (z - p.z));
        //数学公式两点距离
        System.out.println("两点的距离: " + m);
        double distance = Math.sqrt(x * x + y * y + z * z);
        System.out.println("点 1 到原点的距离: " + distance);
    }
    public void count2(Ponit p) {
        double m = Math.sqrt((x - p.x) * (x - p.x)
+ (y - p.y) * (y - p.y)+ (z - p.z) * (z - p.z));
        System.out.println("两点的距离: " + m);
        double distance = Math.sqrt(x * x + y * y + z * z);
        System.out.println("点 2 到原点的距离: " + distance);
    }
}
}

```

2、复数问题

```

import java.util.Scanner;

public class Test4_2 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in); //输入第一个复数
        System.out.println("请输入复数 1 的实部");
        double r1 = in.nextDouble();
        System.out.println("请输入复数 1 的虚部");
        double im1 = in.nextDouble();

        System.out.println("请输入复数 2 的实部"); //输入第二个复数
        double r2 = in.nextDouble();
        System.out.println("请输入复数 2 的虚部");
        double im2 = in.nextDouble();

        Complex c1 = new Complex(r1, im1);
    }
}

```



```

        Complex c2 = new Complex(r2, im2);

        System.out.println("复数 1 为: "+c1);
        System.out.println("复数 2 为: "+c2);
        System.out.println("(c1 + c2) = " + c1.add(c2));
        System.out.println("(c1 - c2) = " + c1.sub(c2));
        System.out.println("(c1 * c2) = " + c1.mul(c2));
    }
}

class Complex {
    private double real, imaginary;
    //有参构造用于声明对象
    Complex(double r, double i) {
        this.imaginary = i;
        this.real = r;
    }

    public Complex add(Complex c2) { //实现两个复数相加
        Complex ans = new Complex(0, 0);
        ans.real = this.real + c2.real;
        ans.imaginary = this.imaginary + c2.imaginary;
        return ans;
    }

    public Complex sub(Complex c2) { //实现两个复数相减
        Complex ans = new Complex(0, 0);
        ans.real = this.real - c2.real;
        ans.imaginary = this.imaginary - c2.imaginary;
        return ans;
    }

    public Complex mul(Complex c2) { //实现两个复数相乘
        Complex ans = new Complex(0, 0);
        ans.real = this.real * c2.real - this.imaginary * c2.imaginary;
        ans.imaginary = this.real * c2.imaginary + this.imaginary * c2.real;
        return ans;
    }

    public String toString() { //实现复数的打印, 为无参构造
        return " ( " + real + " , " + imaginary + " ) ";
        //尊崇题目原则 (a, b) 的格式
        //需要注意复数的虚部不含有 i
    }
}

```

四、实验结果截屏图

1、空间点问题

```
PS C:\Users\Administrator\Desktop\java实验> javac Test4_1.java
PS C:\Users\Administrator\Desktop\java实验> java Test4_1.java
    请输入第一个点的空间坐标：
1 1 1
    请输入第二个点的空间坐标：
2 2 2
-----
-----1、输出点1到原点的距离-----
-----2、输出点2到原点的距离-----
-----3、两点到原点的距离都输出-----
-----0、退出系统(quit)-----
-----
1
-----1、输出点1到原点的距离-----
两点的距离： 1.7320508075688772
点1到原点的距离： 1.7320508075688772
-----
-----1、输出点1到原点的距离-----
-----2、输出点2到原点的距离-----
-----3、两点到原点的距离都输出-----
-----0、退出系统(quit)-----
-----
2
-----2、输出点2到原点的距离-----
两点的距离： 1.7320508075688772
点1到原点的距离： 3.4641016151377544
-----
-----1、输出点1到原点的距离-----
-----2、输出点2到原点的距离-----
-----3、两点到原点的距离都输出-----
-----0、退出系统(quit)-----
-----
3
-----3、两点到原点的距离都输出-----
两点的距离： 1.7320508075688772
点1到原点的距离： 1.7320508075688772
两点的距离： 1.7320508075688772
点1到原点的距离： 3.4641016151377544
-----
-----1、输出点1到原点的距离-----
-----2、输出点2到原点的距离-----
-----3、两点到原点的距离都输出-----
-----0、退出系统(quit)-----
-----
```

2、复数问题

```
PS C:\Users\Administrator\Desktop\java实验> javac Test4_2.java
PS C:\Users\Administrator\Desktop\java实验> java Test4_2.java
请输入复数1的实部
2.1
请输入复数1的虚部
3.5
请输入复数2的实部
1.6
请输入复数2的虚部
5
复数1为: ( 2.1 , 3.5 )
复数2为: ( 1.6 , 5.0 )
(c1 + c2) = ( 3.7 , 8.5 )
(c1 - c2) = ( 0.5 , -1.5 )
(c1 * c2) = ( 1.6 , 5.0 )
```

五、实验结果与分析

1、类的特点:

类是对象的数据类型;类是具有相同属性和行为的一组对象的集合;

2、对象:

客观存在的事物都是对象;

3、对象的属性:

对象具有的各种特征,每个对象的每个属性都有特定的值;

4、对象的使用创建对象:

格式: 类名 对象名 = new 类名();

范例: Phone p = new Phone();

5、this 关键字:

表示当前类的对象;在普通的方法中, this 总是指向调用该方法的对象;

(1) 在构造方法中, this 总是指向正要初始化的对象;

(2) 可以使用 `this` 调用重载的构造方法,但是必须位于构造方法的第一句代码;

(3) `this` 不能用在 `static` 方法;

(4) `this` 可以在方法内部获取对象的属性信息

(5) `this` 可以区分局部变量和成员变量;

6、构造方法

构造方法是一种特殊的方法,``

执行时机: `new` 创建对象的时候,自动调用;

功能: 主要完成对象数据的初始化;

格式: `public 类名() {}`

(1) 构造方法的创建

如果没有定义构造方法,系统将给出一个默认的非参数的构造方法;

如果定义了构造方法,系统将不再给出默认的非参数构造方法;

(2) 构造方法的重载

如果自定义了构造方法,还要使用非参数的构造方法,就必须写一个非参数构造方法;

(3) 推荐的代码风格

无论是否使用非参构造方法,都要手工书写非参构造方法;

(4) 构造方法有返回值,但不能定义返回值类型,不能在构造方法里面 `return` 一个返回值;



山西工学院
SHANXI COLLEGE OF TECHNOLOGY

实验报告

学院名称： 信息产业学院

专业名称： 计算机科学与技术

课程名称： 面向对象程序设计

班 级： 211060104 学号： 21106010413

学生姓名： 刘钟泽

指导老师： 任少斌

2022 年 10 月 9 日

学生姓名	刘钟泽	学号	21106010413	实验成绩	
实验项目名称	程序封装实现与测试				
实验地点	机房	实验日期	2022 年 10 月 9 日		
<div>一、实验目的和要求</div> <p>在面向对象程序设计过程中, 为了保证数据安全, 通常不允许从外面直接访问类的成员变量, 系统通过类中设计的特定的方法处理与类相关的数据。因此在设计中对于全体成员变量或需要保护的成员变量通过 <code>private</code> 修饰。需要设计一组 <code>get/set</code> 方法来处理相关业务逻辑。此方式使用 Eclipse 中的生成方式可以快速解决。</p> <div>二、实验内容和原理</div> <p>参考上面的演示代码, 创建银行账号 <code>Account</code> 类, 成员变量为 <code>ID (帐号) number(金额)</code>, 实现存款类(<code>balance</code>)的不同业务: 存(<code>deposit</code>)、取(<code>withdraw</code>)和查询(<code>getbalance</code>)等功能。要求对成员变量进行封装处理。考虑如何设计菜单选项。</p> <p>提示: 在 Eclipse 中, 在代码区按 <code>Alt_Shift_S</code> 快捷键, 可以直接使用生成工具完成, 也可以在代码点击鼠标右键, 选择 <code>Source</code>, 然后在级联菜单中选择 <code>Generate Getters/Setters...</code> 功能。</p>					

三、程序代码（要求有注释说明）

```
import java.util.Scanner;

public class Test5 {
    public static void main(String[] args) {
        System.out.printf("\033[H \033[2J");//这里是实现清屏指令
        //\033[H: 将光标移动到屏幕或控制台的左上角。
        //\033[2J: 从光标处清屏到屏幕末尾。
        //此处的清屏时为了之后如需做图形页面更方便

        System.out.printf("请先录入第一个信息\n输入账户 id, 姓名, 密码, 金额\n");

        Scanner sc1 = new Scanner(System.in);//输入账户信息
        Account acc = new Account
        (sc1.nextInt(), sc1.next(), sc1.nextInt(), sc1.nextInt());
        Balance bal = new Balance
        (acc.getId(), acc.name, acc.getPassword(), acc.getNumber());
        while (true) {
            System.out.println
            ("-----");
            System.out.println
            ("-----欢迎进入银行账户操作系统-----");
            System.out.println
            ("--welcome to the Bank Account Operating System--");
            System.out.println
            ("-----1 查询余额(getbalance)-----");
            System.out.println
            ("-----2 取款操作(withdraw)-----");
            System.out.println
            ("-----3 存款操作(withdraw)-----");
            System.out.println
            ("-----4 录入信息(enterInformation)-----");
            System.out.println
            ("-----0 退出系统(quit)-----");
            System.out.println
            ("-----");

            Scanner in = new Scanner(System.in);
            int choice = in.nextInt();
            switch (choice) { //用选择结构, 进行图形界面的功能选择
                case 0:
```

```

        System.exit(0);
        break;
    case 1:
        System.out.println
            ("-----1 查询余额
(getbalance)-----");
        bal.getbalance();
        break;
    case 2:
        System.out.println
            ("-----2 取款操作
(withdraw)-----");
        bal.withdraw();
        break;
    case 3:
        System.out.println
            ("-----3 存款操作
(withdraw)-----");
        int inNum = sc1.nextInt();
        bal.deposit(inNum);
        break;
    case 4:
        System.out.println
            ("-----4 录入信息
(enterInformation)-----");
        acc.enterInformation();
        break;
    default:
        System.out.println
            ("您的选择有误! 请重新选择");
        break;
    }
}
}
}

class Account {
    private int id;//使用 private 保护成员变量
    public String name;
    private int number;
    private int password;

    public Account(int id, String name, int password, int number) {
        this.setId(id);//使用 get/set 来实现调用
    }
}

```



```

        this.name = name;
        this.setPassword(password);
        this.setNumber(number);
    }

    public void enterInformation() { //实现录入信息
        System.out.printf("输入账户 id, 姓名, 密码, 金额\n");
        Scanner sc = new Scanner(System.in); //输入账户信息
        setId(sc.nextInt());
        name = sc.next();
        setPassword(sc.nextInt());
        setNumber(sc.nextInt());
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }

    public int getPassword() {
        return password;
    }

    public void setPassword(int password) {
        this.password = password;
    }
}

class Balance extends Account{
    public Balance(int id, String name, int password, int number) {
        super(id, name, password, number);
    }
    public void getbalance() { //实现查询功能
        System.out.println("账号: " + getId());
        System.out.println("姓名: " + name);
        System.out.println("金额: " + getNumber());
    }
    public void withdraw() { //实现存钱功能

```

```
while (true) {
    Scanner sc = new Scanner(System.in);
    System.out.println("请输入密码进行验证!");
    int pass = sc.nextInt();
    if (pass == getPassword()) {
        System.out.println("请输入需要取款的金额: ");
        int withdrawals = sc.nextInt();
        if (withdrawals <= getNumber()) {
            setNumber(getNumber() - withdrawals);
            System.out.println("账户余额: " + getNumber());
        } else {
            System.out.println("当前余额不足!");
        }
        break;
    } else {

        System.out.println("你输入的密码有误, 请重新输入!");
    }

}

}

public void deposit(int inmoney) { //实现取钱功能
    setNumber(getNumber() + inmoney);
    System.out.println("此次存款为: " + inmoney);
    System.out.println("账户余额: " + getNumber());
}

}
```

四、实验结果截屏图

```
C:\Users\Administrator\Desktop\java实验> javac Test5.java
PS C:\Users\Administrator\Desktop\java实验> java Test5.java
```

请先录入第一个信息

输入账户id, 姓名, 密码, 金额

10413 刘钟泽 123456 10000000

```
-----欢迎进入银行账户操作系统-----
--welcome to the Bank Account Operating System--
-----1查询余额(getbalance)-----
-----2取款操作(withdraw)-----
-----3存款操作(withdraw)-----
-----4录入信息(enterInformation)-----
-----0退出系统(quit)-----
-----
```

1

```
-----1查询余额(getbalance)-----
账号: 10413
姓名: 刘钟泽
金额: 10000000
-----
```

```
-----欢迎进入银行账户操作系统-----
--welcome to the Bank Account Operating System--
-----1查询余额(getbalance)-----
-----2取款操作(withdraw)-----
-----3存款操作(withdraw)-----
-----4录入信息(enterInformation)-----
-----0退出系统(quit)-----
-----
```

2

```
-----2取款操作(withdraw)-----
请输入密码进行验证!
123456
请输入需要取款的金额:
5000
账户余额: 9995000
```

```
-----
-----欢迎进入银行账户操作系统-----
--welcome to the Bank Account Operating System--
-----1查询余额(getbalance)-----
-----2取款操作(withdraw)-----
-----3存款操作(withdraw)-----
-----4录入信息(enterInformation)-----
-----0退出系统(quit)-----
-----
```

2

```
-----2取款操作(withdraw)-----
```

请输入密码进行验证！

123

你输入的密码有误，请重新输入！

请输入密码进行验证！

123456

请输入需要取款的金额：

6666

账户余额：9988334

```
-----
-----欢迎进入银行账户操作系统-----
--welcome to the Bank Account Operating System--
-----1查询余额(getbalance)-----
-----2取款操作(withdraw)-----
-----3存款操作(withdraw)-----
-----4录入信息(enterInformation)-----
-----0退出系统(quit)-----
-----
```

3

```
-----3存款操作(withdraw)-----
```

50000

此次存款为：50000

账户余额：10038334

```
-----
-----欢迎进入银行账户操作系统-----
--welcome to the Bank Account Operating System--
-----1查询余额(getbalance)-----
-----2取款操作(withdraw)-----
-----3存款操作(withdraw)-----
-----4录入信息(enterInformation)-----
-----0退出系统(quit)-----
-----
```

0

五、实验结果与分析

封装可是一个保护屏障，防止该类的代码和数据被外部类定义的代码随机访问。要访问该类的代码和数据，必须通过严格的接口控制。

封装最主要的功能在于我们能修改自己的实现代码，而不用修改那些调用我们代码的程序片段。

适当的封装可以让程式码更容易理解与维护，也加强了程式码的安全性。

封装的优点

1. 良好的封装能够减少耦合。
2. 类内部的结构可以自由修改。
3. 可以对成员变量进行更精确的控制。
4. 隐藏信息，实现细节。

在这个实验中任然留有遗憾，没有实现多个数据的录入，与输出，目前思路为创建结构体数组用于存储录入数据，输出时可以用循环输出



山西工学院
SHANXI COLLEGE OF TECHNOLOGY

实验报告

学院名称： 信息产业学院

专业名称： 计算机科学与技术

课程名称： 面向对象程序设计

班 级： 211060104 学号： 21106010413

学生姓名： 刘钟泽

指导老师： 任少斌

2022 年 10 月 16 日

学生姓名	刘钟泽	学号	21106010413	实验成绩	
实验项目名称	类的继承实现与测试				
实验地点	机房	实验日期	2022 年 10 月 16 日		
<p>一、实验目的和要求</p> <p>参考课堂教学内容，设计一个能够体现继承概念的故事。</p> <p>二、实验内容和原理</p> <p>设计一个动物(Animal 类)设计子类(Dog 与 Cat 实现继续 Animal 模式)。编写一个测试类验证继承方式与结果，特别注意继承父类构造方法的方式，使用 Eclipse 编写此功能，注意代码中的区别是什么？</p> <p>三、程序代码（要求有注释说明）</p> <pre>public class Test6 { public static void main(String[] args) { Cat cat = new Cat(); Dog dog = new Dog(); cat.setAge(6); cat.setName("小猫咪"); //继承会继承属性和方法 dog.setAge(8); dog.setName("大黑狗"); cat.voice(); cat.eat(); } }</pre>					

```

        cat.leg();//调用父类的方法
        cat.act();
        cat.age();
        System.out.printf("\n");
        dog.voice();
        dog.eat();
        dog.leg();//调用父类的方法
        dog.act();
        dog.like();//子类特有的方法
        dog.age();
    }
}

class Animal { //动物类
    //创建一个 Animal 类，封装属性，保留接口
    private String name;
    private int age;

    // 构造方法
    // 无参构造
    public Animal() {
        super();
    }
    public void leg() {
        System.out.println(this.getName() + "四条腿");
    }

    // 有参构造
    public Animal(String name, int age) {
        super();
        this.name = name;
        this.age = age;
    }
    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

```



```
    public int getAge() {
        return age;
    }

    public void age() {
        System.out.println(this.getName() + age + "岁");
    }
}

class Cat extends Animal {    // 猫类
    //继承 Animal 类, extends 关键字
    public void voice() {
        System.out.println(this.getName() + "喵喵叫");
    }

    public void eat() {
        System.out.println(this.getName() + "爱吃鱼");
    }
    public void act() {
        System.out.println(this.getName() + "抓老鼠");
    }
}

class Dog extends Animal {    // 狗类
    //继承 Animal 类, extends 关键字
    public void voice() {
        System.out.println(this.getName() + "汪汪叫");
    }

    public void eat() {
        System.out.println(this.getName() + "爱吃肉");
    }

    public void act() {
        System.out.println(this.getName() + "看大门");
    }
    public void like() {
        System.out.println("大黑狗爱捡飞盘");
    }
}
```

四、实验结果截屏图

```
PS C:\Users\Administrator\Desktop\java实验> javac Test6.java
PS C:\Users\Administrator\Desktop\java实验> java Test6.java
小猫咪喵喵叫
小猫咪爱吃鱼
小猫咪四条腿
小猫咪抓老鼠
小猫咪6岁

大黑狗汪汪叫
大黑狗爱吃肉
大黑狗四条腿
大黑狗看大门
大黑狗爱捡飞盘
大黑狗8岁
```

五、实验结果与分析

1、所谓继承：是指可以让某个类型的对象获得另一个类型的对象的属性的方法

2、继承通过 extends 实现

格式：class 子类 extends 父类 { }

举例：class Cat extends Animal { }

3、好处

(1) 继承可以让类与类之间产生关系，即子父类关系。产生子父类关系后，子类则可以使用父类中非私有的成员变量和成员方法。

(2) 子类想要有自己的特有的方法可以自己添加或重写

4、特点

(1) 子类可以继承父类的可访问的成员变量和方法

(2) 子类可以有自己的成员变量和方法

(3) 静态方法、静态成员变量也可以被子类继承（private 的不会）