《面向对象程序设计》实验报告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | | | 《面向对象程序设计》 | | | | |
| 实验名称 | | | 实验三 类和对象 | | | | |
| 姓名 | | 卓如意 | | 学号 | 22145138 | 班级 | 计算机科学与技术221 |
| 实  验  目  的 | 1. 掌握数组和循环语句的结合应用 2. 掌握二维数组的使用，理解Java多维数组的内存结构 3. 常见类string、Math、Calendar等类的综合使用 | | | | | | |
| 实  验  内  容 | 1. 结合PTA在线练习3数组掌握Java数组的定义和使用，初步理解引用类型和值类型的差别； 2. 结合PTA在线练习4~5 掌握Java类库中常用类BigDecimal、Math、String、数据包裹类、日期相关类等的基本属性和方法； | | | | | | |
| 实验过程记录 | 练习3数组  7-7找出不是两个数组的共有元素   1. 实验思路：   在两个数组中找出不是两者共有的元素添加到另一个数组中，并去除重复元素。   1. 实验难点/优化方向：   难点1：如何高效去重  目前我的解决方法是在取得不共有重复数组c的情况下依次比对先前的元素，保证在此位置的情况下唯一。  难点2：如何高效比对共有元素。  个人认为本人给出的解决方案效率还是太低，因为采取的是暴力的两次比对。   1. 实验代码：   import java.util.Scanner;  public class Main{  public static void main(String [] ss){  Scanner sc=new Scanner(System.in);  int n,k=0,s,p=0,i,j,m;  n=sc.nextInt();  int [] a=new int[n+1];  int [] b=new int[n+1];  int [] c=new int[n+1];  for( i=0;i<n;i++){  a[i]=sc.nextInt();  }  s=sc.nextInt();  for( i=0;i<s;i++){  b[i]=sc.nextInt();  }  for( i=0; i<n; i++){  for( j=0; j<s; j++){  if(a[i]==b[j])  break;  }  if(j>=s){  c[k]=a[i];  k++;  }  }  for( i=0; i<s; i++){  for( j=0; j<n; j++){  if(b[i]==a[j])  break;  }  if(j>=n){  c[k]=b[i];  k++;  }  }  int cnt=0;  for(i=0;i<k;i++){  boolean fl=true;  for(j=0;j<i;j++){  if(c[i]==c[j]){  fl=false;  break;  }  }  if(fl==true){  if(cnt!=0)System.out.print(" ");  System.out.print(c[i]);  cnt++;  }  }  System.out.print("\n");  }  }   1. 实验结果：     在线练习4~5  练习5 7-6月历   1. 实验思路：   首先导入Scanner和Calendar类，用于接收用户的输入和处理日期。  定义一个二维数组和一个字符串数组Mon，用于存储月份和英文月份的名称。接收用户输入的年份和月份，并根据月份从Mon数组中获取英文月份名称，打印出来。初始化一个字符串数组week，用于存储一周的英文名称，按顺序打印出来。使用for循环计算从1900年至用户指定年份之间闰年的总天数sumDay，用这个总天数加上用户指定年的1月1日至当前日期之前的天数。创建一个Calendar实例，并设置年份和月份为用户输入的值。调用calendar.getActualMaximum(Calendar.DAY\_OF\_MONTH)获取当月的最大天数。获得1月1日所在的星期，并打印之前没有日期的部分。逐日打印日历，如果当前日期与第一个日期不在同一天，则在其前面插入空白。如果打印完一周后还有剩余的日期，继续打印下一周。   1. 实验难点/优化方向：   难点主要在于如何获得当月最大天数和如何定位这个月从星期几开始。   1. 实验代码：   import java.util.Scanner;  import java.time.\*;  import java.util.Calendar;  public class Main{  public static void main(String[]ss){  Scanner sc=new Scanner(System.in);  int year=sc.nextInt();  int mon=sc.nextInt();  String [] Mon={" ","JAN.","FEB.","MAR.","APR.","MAY.","JUN.","JUL.","AUG.","SEP.","OCT.","NOV.","DEC."};  System.out.print(Mon[mon]+" "+year+"\n");  String [] week={" ","Mon","Tue","Wed","Thu","Fri","Sat","Sun"};  for(int i=1;i<8;i++){  if(i>1)System.out.print(" ");  System.out.print(week[i]);  }  System.out.println();  System.out.println("---------------------------");  int sumDay=0;  for(int i=1900;i<year;i++){  sumDay+=(i%4==0 && i%100!=0 || i%400==0)?366:365;  }  Calendar calendar=Calendar.getInstance();  calendar.set(year, mon-1,01);  int wk=calendar.get(Calendar.DAY\_OF\_WEEK)-1;  int max = calendar.getActualMaximum(Calendar.DAY\_OF\_MONTH);  if(wk==0)wk=7;  for(int i=1;i<wk;i++){  if(i>1)System.out.print(" ");  System.out.print(" ");  }  for(int i=1;i<=max;i++){  if((i+wk-1)%7!=1)System.out.print(" ");  System.out.printf("%3d",i);  if((i+wk-1)%7==0){  System.out.print("\n");  }  }  }   1. 实验结果： | | | | | | |
| 实验心得  体会 | 一、关于数组  数组是一种特殊的数据类型，它可以存储多个相同类型的数据。Java中的数组分为两种：一种是简单类型的数组，如int、char、boolean等；另一种是对象类型的数组，即数组中的元素都是某种对象。  数组具有以下四个基本特点：  1. 数组的长度是确定的，数组一旦被创建，它的大小就是不可以改变的。如果越界，则报异常；  2. 其元素必须是相同类型，不允许出现混合类型；  3. 数组中的元素可以是任何数据类型，包括基本类型和引用类型；  4. 数组变量属引用类型，数组也可以看成是对象，数组中的每个元素相当于该对象的成员变量，数组本身就是对象，Java中对象是在堆中的，因此数组无论保存原始类型还是其他对象类型，数组对象本身是在堆中的。  Java中有一个内置的`Arrays`类，它提供了很多实用的方法，可以直接对数组进行排序、搜索、复制、填充等操作。例如，可以使用`Arrays.sort()`方法来对数组进行排序。   1. 关于常用类 2. BigDecimal：BigDecimal类在Java中用于执行精确的十进制运算，特别是针对货币计算或其他需要精确度较高的场合。它提供了add()、subtract()、multiply()、divide()等基本运算方法，以及setScale()、toString()等方法。 3. Math：Math类提供了许多静态方法用于执行数学运算，例如绝对值、最大值、最小值、指数、对数、平方根、正弦、余弦、正切等。 4. String：String类表示字符串对象，是Java中最常用的类之一。它提供了一套丰富的API用于字符串的操作，例如concat()、length()、substring()、indexOf()、charAt()等。 5. 数据包裹类：Java中的数据包装类如Byte、Short、Integer、Long、Float、Double、Character和Boolean，将基本类型封装成了对象类型，从而可以利用面向对象的特性，比如equals()、compareTo()、valueOf()等。 6. 日期相关类：在Java中，我们有多种日期相关的类，例如Date、Calendar、LocalDate、LocalTime、LocalDateTime等。这些类提供了操作日期和时间的功能，如getYear()、getMonth()、getDay()等，可以用来获取、设置和操作日期和时间。 | | | | | | |
| 教师批阅 |  | | | | | | |