山东大学 软件 学院

高级程序设计语言 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300125 | 姓名：贾星宇 | | 班级： 2020级5班 |
| 实验题目：数组 | | | |
| 实验学时：4 | | 实验日期：2020年11月16日 | |
| 实验目的：  熟练掌握一维及二维数组的使用 | | | |
| 硬件环境：  Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11GHz | | | |
| 软件环境：  IntelliJ IDEA Community;  Java jdk1.8.0\_261 | | | |
| 实验步骤与内容：  由键盘输入年份，编写一个程序显示当年的日历，用Application程序实现. | | | |
| 结论分析与体会：  1.求某一年的1月1日是星期几的最常见的公式为（设当前年份是Y）：  W = [Y-1] + [(Y-1)/4] - [(Y-1)/100] + [(Y-1)/400]  公式中的[...]指只取计算结果的整数部分。算出来的W除以7，余数是几就  是星期几。如果余数是0，则为星期日。  2.具有N个数组的索引为0~（N-1）。  3.在Java中，数组是必须实例化的对象。  4.数组边界检查保证引用一个元素的索引值在有效范围内。  5.初始值表可以用于实例化一个数组对象，这是不再需要用new运算符创建数组对象。  6.整个数组可以作为一个参数传递给方法，使得方法的形参成为原始数组的别名。  7.实例化对象数组只是为保存对象引用而预留空间，对每个数组元素所代表的对象必须分别实例化。  8.命令行参数存储在string对象数组中，并将传递给main方法。  9. 假设我们已经定义了一个日期类MyDate，MyDate [ ] date=new MyDate[3];仅仅是创建了一个数组，数组中有三个变量date[0] 、date[1]、 date[2]，用于存放MyDate对象的地址，在使用date[i]之前，必须还要有如下的语句 for (int i=0; i<3; i++ ) date[i]=new MyDate(); 这样才为数组中的每个元素申请了内存空间。这在数组元素为类类型变量的时候要特别注意。  10.length为数组的公共实例变量，指数组的长度（可容纳的元素个数）。date数组的长度可通过date.length得到。而数组的下标总是从0开始，所以最后一个下标应比数组的长度小1即date.length-1。所以遍历数组时应为 for (int i=0; i< date.length-1; i++ ) 若数组下标越界，编译时无任何出错信息，但运行时报错：ArrayIndexOutOfBoundsException  11.一个java方法可以定义为参数个数可变的方法。  12.在面向对象的系统中，很少使用高于二维的数组。  主要代码：  import java.util.Scanner; public class Experiment6 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.*in*);  System.*out*.print("Please enter the year:");  int year = scanner.nextInt();  Experiment6 exp = new Experiment6();  exp.last(year);  }  *//获得某年月日是星期几* public static int getWeekDay(int year,int month,int day){  int w,allDay=0;  w = ((year-1)+(int)((year-1)/4)-(int)((year-1)/100)+(int)((year-1)/400))%7;  for (int i=1;i<month;i++){  allDay += *getDay*(year, i);  }  allDay += day;  allDay %= 7;  w += allDay;  return w;  }  *//得到每月有多少天* public static int getDay(int year,int month){  switch (month) {  case 1:  case 3:  case 5:  case 7:  case 8:  case 10:  case 12:  return 31;  case 4:  case 6:  case 9:  case 11:  return 30;  case 2:  if (year % 4 == 0 && year % 100 != 0)  return 29;  else  return 28;  }  System.*out*.println("error month!");  return -1;  }   public void display(int year,int month,int day){  if(day==1)  {  for(int i=0;i<*getWeekDay*(year,month,day);i++){  System.*out*.print(" ");  }  }  if(day<10)  System.*out*.print(" ");  System.*out*.print(" "+day);  if(*getWeekDay*(year,month,day)==6)  System.*out*.println();   } *//最终输出,格式化输出* public void last(int year){  System.*out*.println();  System.*out*.println(" "+year+"年日历");  System.*out*.println();   for(int i=0;i<12;i++){  System.*out*.println(year+"-"+(i+1));  System.*out*.println(" 日 一 二 三 四 五 六");  for(int j=0;j<*getDay*(year,i+1);j++){  display(year,i+1,j+1);  }  System.*out*.println();  System.*out*.println();  }  } } | | | |