|  |
| --- |
| **一、实验目的：**  1、掌握数组的定义和使用方法  2、熟悉数组的排序、查找等算法 |
| **二、实验环境：**  有Java环境的计算机若干台 |
| **三、实验内容和要求：**  1、定义一个10个元素的数组，要求从键盘输入（Scanner类，循环输入），并在里面查找某个数据，找到输出“有”，没有这个数提示“无”。  *import* java.util.Scanner;  *public class* Q1 {  *public static void* main(String[] args) {  *int* max = 10;  *float*[] fArr = *new float*[max];  Scanner scanner = *new* Scanner(System.*in*);   *for* (*int* i = 0; i < max; i++) {  fArr[i] = scanner.nextFloat();   }   System.*out*.println("find a element: ");  *float* x = scanner.nextFloat();  *for* (*float* i : fArr) {  *if* (Math.*abs*(i - x) < 0.001d) {  System.*out*.println("有");  *return*;  }   }  System.*out*.println("无");  } }    思路：使用Scanner类输入数组元素，然后再输入要查找的元素，循环比较查找。  2、编写一个数组处理类，要求里面有这样的两个方法，第一个方法，可以对输入的数组进行排序，第二个方法可以删除指定位置的数组元素；然后再写一个主类，测试上面这个类的正确性。  思路：第一个类框架  class first{  void sort（int [] a）{//该方法为对a数组排序的方法}  void delelement（int[] a,int index）{//该方法实现删除a数组内第index 位置的元素}  //可以增加折半等等  }  第二个类框架：  class second{  public static void main（String args[]）  { 使用Scanner输入一个数组，采用前一个类的第一方法对其排序，输出看看是否正确，在采用前一个类的第二个方法，输出第3个元素，验证正确性。  }  *import* java.util.Arrays; *import* java.lang.System; *import* java.util.Scanner;  *public class* Q2 {  *public static void* main(String[] args) {  *int*[] iArr = *new int*[5];  Scanner scanner = *new* Scanner(System.*in*);  *for* (*int* i = 0; i< 5;i++) {  iArr[i] = scanner.nextInt();  }  Until u = *new* Until();  u.sort(iArr);   *for* (*int* i :  iArr) {  System.*out*.println(i);  }  *int*[] delArr = u.deleteElem(iArr, 1);  *for* (*int* i :  delArr) {   System.*out*.println(i);  }  } }  *class* Until {  *void* sort(*int*[] arr) {  *int* temp;   *for* (*int* i = 0, length = arr.length; i < length - 1; i++) {  *for* (*int* j = 0; j < length - 1 - i; j++) {  *if* (arr[j] > arr[j + 1]) {  temp = arr[j];  arr[j] = arr[j + 1];  arr[j + 1] = temp;  }  }  }  }   *int*[] deleteElem(*int*[] arr, *int* index) {  *for* (*int* i = index, length = arr.length; i < length - 1; i++) {  arr[i] = arr[i + 1];  }   *return* Arrays.*copyOf*(arr, arr.length - 1); *// 新数组缩容* } }  3、给定一个十进制正整数（不超过 3000 位） ， 请将其转换成对应的二进制整数  比如3465555369355754989772  输出  101110111101111000111101111101110010000100111001011011011110100011001100    *import* java.math.BigInteger;  *public class* Q3 {  *public static void* main(String[] args) {  String n = "3465555369355754989772";  BigInteger num = *new* BigInteger(n);  *toBin*(num);  }   *private static void* toBin(BigInteger n) {  BigInteger base = *new* BigInteger("2");  *if* ( n.divide(base).compareTo(*new* BigInteger("0")) > 0) {  *toBin*(n.divide(base));  }  System.*out*.print(n.mod(base));  } } |
| **四、实验步骤：** |
| **五、实验结果与分析（**含程序、数据记录及分析和实验总结等**）：** |