

# 中北大学软件学院

## 实 验 报 告

专 业:	软件工程
课程名称:	面向对象程序设计
班 级:	22130403
学 号:	2213040340
姓 名:	张恒瑜
辅导教师:	李华玲

2021 年 9 月 制

成绩：\_\_\_\_\_

实验时间	2023 年 9 月 28 日 17 时至 21 时	学时数	2 学时
1. 实验名称 数组和循环嵌套			
2. 实验目的  (1) 掌握数组的定义、初始化、访问和使用； (2) 学会循环嵌套语句在数组中的使用； (3) 学会使用数组解决实际问题； (4) 学会数组在面向对象程序设计中的运用；			
3. 训练知识点集群  定义数组，初始化数组与数组赋值，使用数组，数据类型转换，while 循环块，for 循环块			
4. 实验内容  (1) 根据用户输入的 10 个数值，自己写排序算法进行升序排列并输出排序后的结果。 (2) 向控制台输出 10 * 10 的数值表格。输出效果如下： <pre>(0,0)(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7)(0,8)(0,9) (1,0)(1,1)(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)(1,8)(1,9) (2,0)(2,1)(2,2)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)(2,9) (3,0)(3,1)(3,2)(3,3)(3,4)(3,5)(3,6)(3,7)(3,8)(3,9) (4,0)(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)(4,7)(4,8)(4,9) (5,0)(5,1)(5,2)(5,3)(5,4)(5,5)(5,6)(5,7)(5,8)(5,9) (6,0)(6,1)(6,2)(6,3)(6,4)(6,5)(6,6)(6,7)(6,8)(6,9) (7,0)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(7,7)(7,8)(7,9) (8,0)(8,1)(8,2)(8,3)(8,4)(8,5)(8,6)(8,7)(8,8)(8,9) (9,0)(9,1)(9,2)(9,3)(9,4)(9,5)(9,6)(9,7)(9,8)(9,9)</pre> (3) 课程融合之“N 元组”： 设计实现抽象数据类型“N 元组 (Tuple)”。每个 N 元组由任意 N 个实数的序列构成，基本操作包括：创建一个 N 元组，取 N 元组的任意一个分量，置 N 元组的任意一个分量，求 N 元组的最大分量、最小分量，显示 N 元组等。 要求：用类封装需要定义的数据类型 Tuple，完成所有基本操作的 Java 语言实现与调用，并写测试程序。 (4) 小组自主创作实验：围绕实验目的，题目自拟。			
5. 写出实验内容（1）所采用的排序算法的思路  实验一所使用的是冒泡排序  思路：冒泡排序是有两层循环，第一层是交换的次数，第二层是比较的次数，次数是比较难想象的，但是可以通过列举数字			

13245 等等少量数字来达到目的。

冒泡排序是从第一个数开始，一个个比较，12 相比，若  $1 > 2$  则交换，第一次会在末尾得到数组中最大的数，第二次会将数组中第二大的数排在倒数第二个的位置，以此类推，得到升序或者降序的排列是十分容易的

6. 画出实验内容（3）对应的类图

NTuple
Elem[];float
Initntuple();void
SearElem();float
ChangeElem();void

```
Maxntuple();float
```

```
Minntuple();float
```

```
Showntuple();void
```

## 7.实验源代码



( 1 ) TestBubbleSort.java

//根据用户输入的 10 个数值，自己写排序算法进行升序排列  
并输出排序后的结果

//升序排列，这里使用冒泡排序进行

```
import java.util.Scanner;
```

```
class methodSort {
```

```
    int[] arr;
```

```
    public void bubbleSort(int[] arr) {
```

```
        boolean flag = true;// 做一个标志 flag 并把它定义
```

为 true

```
        for(int i = 0; i < arr.length - 1; i++){
```

```
            for(int j = 0 ; j < arr.length - i - 1; j++){
```

```
                if(arr[j] > arr[j + 1]) {
```

```
                    flag = false; //若经过上述判断进入
```

循环则将标志物定义为 false 继续进行 if 条件

```
        int temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
    }
}
```

if(flag) { //若没有进行 if 条件的交换参与，则

直接按照原来的 true 进行 break

```
        break;
    }
}
}
}
}
public class TestBubbleSort {
    public static void main(String[] args) {
        methodSort method = new methodSort();
        Scanner sc = new Scanner(System.in);
        int[] ints = new int[10];

        System.out.println("请用户输入 10 个数：");
        for(int i = 0; i < ints.length; i++) {
            ints[i] = sc.nextInt();
        }
        method.bubbleSort(ints);
        for(int anInt:ints) {
            System.out.println(anInt + " ");
        }
    }
}
```



( 2 ) FormOptimize.java

```
class Form0 {
    public void crOptimize(int row,int colume) {
        for(int i = 0; i < row; i++ ) {
            for(int j = 0; j < colume; j++) {
                System.out.print("(" + i + "," + j + ")");
            }
            System.out.println();
        }
    }
}
```

```

}
class FormOptimize {
    public static void main(String[] arge) {
        Form0 form = new Form0();
        form.crOptimize(10,10);
    }
}

```



( 3 ) TestNtuple.java

//设计实现抽象数据类型“N 元组 (Tuple)” 。每个 N 元组由任意 N 个实数的序列构成，基本操作包括：创建一个 N 元组，取 N 元组的任意一个分量，置 N 元组的任意一个分量，求 N 元组的最大分量、最小分量，显示 N 元组等。

```

import java.util.Scanner;

class NTuple {
    float[] elem;
    public void Initntuple() {
        Scanner sc = new Scanner(System.in);

        System.out.println("请输入你想创建 n 元组的元素
个数：");

        int num = sc.nextInt();
        elem = new float[num];
        for(int i = 0; i < elem.length; i++) {
            int j = i + 1;

            System.out.println("请输入第 " + j + "个数：");

            elem[i] = sc.nextFloat();
            System.out.println(" ");
        }
    }
    public void SearchElem() {
        Scanner sc1 = new Scanner(System.in);

        System.out.println("请输入你要查数组内第几个

```

```

数:  ");

        int num = sc1.nextInt();
        System.out.println(elem[num - 1]);
    }
    public void ChangeElem() {
        Scanner sc2 = new Scanner(System.in);

        System.out.println("请输入你要改变数组内第几个
数: ");

        int num = sc2.nextInt();

        System.out.println("这个数你要改变为: ");
        elem[num - 1] = sc2.nextFloat();

        System.out.println("数组内第" + num + "个数已
经改为" + elem[num - 1]);
    }
    public float Maxntuple() {
        float max = elem[0];

        int maxindex = 0;//用排序后再找也不是不行，但是
聪明绝顶的张某立马想到了选择排序，而这里只用到了选择排
序的一部分，也是最重要的一部分，就是利用数组下标进行查
找

        for(int i = 0; i < elem.length - 1; i++) {
            if(elem[i] < elem[i + 1]) {
                maxindex = i + 1;
            }
        }
        max = elem[maxindex];
        return max;
    }
    public float Minntuple() {
        float min = elem[0];
        int minindex = 0;
        for(int i = 0; i < elem.length - 1; i++) {
            if(elem[i] > elem[i + 1]) {

```

minindex = i + 1;//可以想象成传送锚点，  
在最小值设置一个，然后直接传送！

```
        }
    }
    min = elem[minindex];
    return min;
}

public void Showntuple() {
    for(float anFloat:elem) {
        System.out.println "[" + anFloat + "];"
    }
}
}

public class TestNtuple {
    public static void main(String[] args) {
        Scanner sc3 = new Scanner(System.in);
        NTuple tuple = new NTuple();
        while(true) {
            System.out.println("*****请输入你想进行的
操作*****");

            System.out.println("*****1.创建一个 N
元组*****");

            System.out.println("*****2.查找 N 元组
的数*****");

            System.out.println("*****3.修改 N 元组
的数*****");

            System.out.println("*****4.找 N 元组最
大值*****");

            System.out.println("*****5.找 N 元组最
```



小值\*\*\*\*\*");

System.out.println("\*\*\*\*\*6.列目前的 N

元组\*\*\*\*\*");

System.out.println("\*\*\*\*\*7.退出程

序\*\*\*\*\*");

int choose = sc3.nextInt();

switch(choose) {

case 1 -> {

tuple.Initntuple();

break;

}

case 2 -> {

tuple.SearchElem();

break;

}

case 3 -> {

tuple.ChangeElem();

break;

}

case 4 -> {

System.out.println("N 元组中最

大的数是：" + tuple.Maxntuple());

break;

}

case 5 -> {

System.out.println("N 元组中最

小的数是：" + tuple.Minntuple());

break;

}

case 6 -> {

tuple.Showntuple();

break;

}

case 7 -> {

return;

```
        }  
        default -> {  
            System.out.println("输入错误请重新  
输入！");  
  
            break;  
        }  
    }  
}  
}
```

## 8.实验运行结果截图及心得体会

(1)

```
D:\zhy0340_java\chap02>javac TestBubbleSort.java

D:\zhy0340_java\chap02>java TestBubbleSort
请用户输入10个数：
10 9 8 7 6 5 4 3 2 1
1
2
3
4
5
6
7
8
9
10

D:\zhy0340_java\chap02>
```

( 2 )

```
D:\zhy0340_java\chap02>javac FormOptimize.java

D:\zhy0340_java\chap02>java FormOptimize
(0,0)(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7)(0,8)(0,9)
(1,0)(1,1)(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)(1,8)(1,9)
(2,0)(2,1)(2,2)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)(2,9)
(3,0)(3,1)(3,2)(3,3)(3,4)(3,5)(3,6)(3,7)(3,8)(3,9)
(4,0)(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)(4,7)(4,8)(4,9)
(5,0)(5,1)(5,2)(5,3)(5,4)(5,5)(5,6)(5,7)(5,8)(5,9)
(6,0)(6,1)(6,2)(6,3)(6,4)(6,5)(6,6)(6,7)(6,8)(6,9)
(7,0)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(7,7)(7,8)(7,9)
(8,0)(8,1)(8,2)(8,3)(8,4)(8,5)(8,6)(8,7)(8,8)(8,9)
(9,0)(9,1)(9,2)(9,3)(9,4)(9,5)(9,6)(9,7)(9,8)(9,9)

D:\zhy0340_java\chap02>
```

( 3 )

```
D:\zhy0340_java\chap02>java TestNtuple
```

```
*****请输入你想进行的操作*****
```

```
*****1. 创建一个N元组*****
```

```
*****2. 查找N元组的数*****
```

```
*****3. 修改N元组的数*****
```

```
*****4. 找N元组最大值*****
```

```
*****5. 找N元组最小值*****
```

```
*****6. 列目前的N元组*****
```

```
*****7. 退出程序*****
```

```
1
```

```
请输入你想创建n元组的元素个数：
```

```
4
```

```
请输入第1个数：
```

```
33
```

```
请输入第2个数：
```

```
22
```

```
请输入第3个数：
```

```
55
```

```
请输入第4个数：
```

```
11
```

```
*****请输入你想进行的操作*****
```

```
*****1. 创建一个N元组*****
```

```
*****2. 查找N元组的数*****
```

```
*****3. 修改N元组的数*****
```

```
*****4. 找N元组最大值*****
```

```
*****5. 找N元组最小值*****
```

```
*****6. 列目前的N元组*****
```

```
*****7. 退出程序*****
```

```
2
```

```
请输入你要查数组内第几个数：
```

```
3
```

```
55.0
```

```
*****请输入你想进行的操作*****
```

```
*****1. 创建一个N元组*****
```

```
*****2. 查找N元组的数*****
```

```
*****3. 修改N元组的数*****
```

```
*****4. 找N元组最大值*****
```

```
*****5. 找N元组最小值*****
```

```
*****6. 列目前的N元组*****
```

```
*****7. 退出程序*****
```

```
4
N元组中最大的数是：55.0
*****请输入你想进行的操作*****
*****1. 创建一个N元组*****
*****2. 查找N元组的数*****
*****3. 修改N元组的数*****
*****4. 找N元组最大值*****
*****5. 找N元组最小值*****
*****6. 列目前的N元组*****
*****7. 退出程序*****
5
N元组中最小的数是：11.0
*****请输入你想进行的操作*****
*****1. 创建一个N元组*****
*****2. 查找N元组的数*****
*****3. 修改N元组的数*****
*****4. 找N元组最大值*****
*****5. 找N元组最小值*****
*****6. 列目前的N元组*****
*****7. 退出程序*****
6
[33.0]
[22.0]
[55.0]
[11.0]
*****请输入你想进行的操作*****
*****1. 创建一个N元组*****
*****2. 查找N元组的数*****
*****3. 修改N元组的数*****
*****4. 找N元组最大值*****
*****5. 找N元组最小值*****
*****6. 列目前的N元组*****
*****7. 退出程序*****
7
D:\zhy0340_java\chap02>
```

心得体会：和三元组多的只是一个数组和排序方法，大同小异，虽然思路很到位，但还是码了很长时间，希望能够更快！再接再厉！

