



HARBIN INSTITUTE OF TECHNOLOGY

哈尔滨工业大学

## Java 程序设计实验报告

学号： \_\_\_\_\_ XXXXXXXXXXX

姓名： \_\_\_\_\_ XXXXX

专业： \_\_\_\_\_ XXXXXXXXXXX

班级： \_\_\_\_\_ XXXXXXXX



选择排序、冒泡排序算法或其他算法（选择其中一种算法实现即可）。排序前后必须保证同一个学生在所有数组中对应相同的下标！为了方便实现上述功能,可定义一个排序数组 `int sortedIndex[]`，该数组中保存了进行排序的数组排序后的下标，排序结束后，返回该数组，以便根据该数据进行打印显示。

- 5) 改编 BMI 程序，增加 `printStudents` 函数，该函数的参数为 `int sortedIndex[]`,该函数可以打印排序前和排序后的结果。打印时，每个学生的信息打印为一行，为了清晰，学号、姓名、身高、体重和计算后的 `bmi` 值之间用制表符(`\n`)隔开。
- 6) 改编 BMI 程序，在 `main` 函数中，调用上述函数，输入至少 3 名学生，并按不同属性排序，并打印排序前、排序后的结果。

### 三、实验步骤

- 1) 在 `ControlStrcutreTest` 中增加 2 个函数，根据不同的参数，可实现打印如下六个用星号组成的三角形。

Step1. 分析题意可知，应当增加的两个函数分别是 用于**选择打印的三角形类型**控制结构函数，和一个**用于实现打印动作**的功能函数。

根据对题意的理解，完成如下函数的代码：

- 1、`ChoiceTriangular(int n);`

```
public static void ChoiceTriangular(int n){
    switch(n){
        case 1:PrintTriangular(" ", "*", 1);break;
        case 2:PrintTriangular(" ", "* ", 1);break;
        case 3:PrintTriangular("*", " ", 2);break;
        case 4:PrintTriangular(" ", "*", 2);break;
        case 5:PrintTriangular(" ", "* ", 2);break;
        case 6:PrintTriangular("*", " ", 1);break;
        default:System.out.println("错误参数");
    }
}
```

- 2、`PrintTriangular(String left,String right,int mode);`

```
public static void PrintTriangular(String left,String right,int mode){
    final int N=10;
    if(mode==1){
        for(int i=0;i<N;i++){
            for(int j=0;j<N-i;j++){
                System.out.printf("%s",left);
            }
            for(int j=0;j<i;j++){
                System.out.printf("%s",right);
            }
            System.out.println();
        }
    }
}
```

```

    }else{
        for(int i=0;i<N;i++){
            for(int j=0;j<i;j++){
                System.out.printf("%s",left);
            }
            for(int j=0;j<N-i;j++){
                System.out.printf("%s",right);
            }
            System.out.println();
        }
    }
}

```

Step2.在主函数中调用并引导用户选择想要的三角形：

```

public static void main(String []argv){
    Scanner input=new Scanner(System.in);
    System.out.println("请输入参数");
    ChoiceTriangular(input.nextInt());
}

```

Step3.观察结果如下：

<p>请输入参数： 1</p> <pre> * ** *** **** ***** ***** ***** ***** ***** ***** </pre>	<p>请输入参数： 2</p> <pre>       *      **     ***    ****   *****  ***** ***** ***** ***** ***** </pre>	<p>请输入参数： 3</p> <pre>       *      **     ***    ****   *****  ***** ***** ***** ***** ***** </pre>
<p>请输入参数： 4</p> <pre> ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** </pre>	<p>请输入参数： 5</p> <pre> ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** </pre>	<p>请输入参数： 6</p> <pre> ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** </pre>

- 2) 在 `ControlStrcutureTest` 中增加函数 `ChineseTrangle(int N)`, 利用二维数组实现  $N$  (小于等于 10) 阶杨辉三角的计算和显示。提示: 代码 `System.out.printf("%4s", "");` 可以打印 4 个空格; `System.out.printf("%4d\n", 1)`; 打印显示 4 个字符宽度的数值。

Step1. 增加 `ChineseTrangle(int N)` 函数如下:

```
public static void ChineseTrangle(int N){
    int[][] number=new int[N][N];

    for(int i=0;i<N;i++)number[i][0]=1;

    for(int i=1;i<N;i++){
        for(int j=1;j<=i;j++){
            number[i][j]=number[i-1][j-1]+number[i-1][j];
        }
    }
    for(int i=0;i<N;i++){
        for(int j=0;j<N-i;j++)System.out.printf("%2s", "");
        for(int j=0;j<=i;j++)System.out.printf("%4d", number[i][j]);
        System.out.println();
    }
}
```

Step2. 在主函数中调用并观察结果:

```
public static void main(String []agrv){
    //Scanner input=new Scanner(System.in);
    //System.out.println("请输入参数");
    //ChoiceTriangular(input.nextInt());
    ChineseTrangle(11);
}
```

```
<terminated> ControlStrcutureTest [Java Application] D:\Program Files\Java
      1
    1  1
  1  2  1
1  3  3  1
1  4  6  4  1
1  5 10 10  5  1
1  6 15 20 15  6  1
1  7 21 35 35 21  7  1
1  8 28 56 70 56 28  8  1
1  9 36 84 126 126 84 36  9  1
1 10 45 120 210 252 210 120 45 10  1
```

- 3) 改编 BMI 程序, 在 `main` 函数中增加数组 `String[] ids`, `String[] names`, `float[] heights`, `float[] weights`, `float[] bmis`, 分别存储学生们的学号、姓名、身高、体重和计算后的 `bmi` 值; 先询问学生的人数, 输入指定人数后, 依次输入各个学生的学号、姓名、身高、体重, 并计算 `bmi` 值, 并将相关数据存储到这些数组中;

Step1.根据题意修改 main()函数如下:

```
public static void main(String []arg){
    Scanner input=new Scanner(System.in);
    System.out.println("请输入学生人数N: ");
    int N=input.nextInt();
    String[] ids=new String[N];
    String[] names=new String[N];
    float[] heights=new float[N];
    float[] weights=new float[N];
    float[] bmis=new float[N];
    System.out.println("请输入每个学生的学号、姓名、身高、体重: ");
    for(int i=0;i<N;i++){
        ids[i]=input.next();
        names[i]=input.next();
        heights[i]=input.nextFloat();
        weights[i]=input.nextFloat();
        bmis[i]=weights[i]/(heights[i]*heights[i]);
    }
}
```

- 4) 改编 BMI 程序, 增加 5 个排序 sortByXXX 函数, XXX 表示排序属性, 可以分别按照学生学号、姓名、身高、体重、BMI 进行由小到大排序, 排序算法可以利用简单排序、选择排序、冒泡排序算法或其他算法(选择其中一种算法实现即可)。排序前后必须保证同一个学生在所有数组中对应相同的下标! 为了方便实现上述功能, 可定义一个排序数组 int sortedIndex[], 该数组中保存了进行排序的数组排序后的下标, 排序结束后, 返回该数组, 以便根据该数据进行打印显示。

Step1.根据题意增加 SortByXXX 函数如下:

1、 public static int[] SortByName(String name[],int N)

```
public static int[] SortByName(String name[],int N){
    int sortindex[]=new int[N];
    for(int i=0;i<N;i++)sortindex[i]=i;

    for(int i=0;i<N;i++){
        int max=i;
        for(int j=i+1;j<N;j++){
            if(name[sortindex[max]].compareTo(name[sortindex[j]])<0)
                max=j;
        }
        if(max!=i){
            int temp=sortindex[max];
            sortindex[max]=sortindex[i];
            sortindex[i]=temp;
        }
    }

    return sortindex;
}
```

2、 `public static int[] SortById(String id[],int N)`

```
public static int[] SortById(String id[],int N){
    int sortindex[]=new int[N];
    for(int i=0;i<N;i++)sortindex[i]=i;

    for(int i=0;i<N;i++){
        int max=i;
        for(int j=i+1;j<N;j++){
            if(id[sortindex[max]].compareTo(id[sortindex[j]])<0)
                max=j;
        }
        if(max!=i){
            int temp=sortindex[max];
            sortindex[max]=sortindex[i];
            sortindex[i]=temp;
        }
    }

    return sortindex;
}
```

3、 `public static int[] SortByHeight(float height[],int N)`

```
public static int[] SortByHeight(float height[],int N){
    int sortindex[]=new int[N];
    for(int i=0;i<N;i++)sortindex[i]=i;

    for(int i=0;i<N;i++){
        int max=i;
        for(int j=i+1;j<N;j++){
            if(height[sortindex[max]]<height[sortindex[j]])
                max=j;
        }
        if(max!=i){
            int temp=sortindex[max];
            sortindex[max]=sortindex[i];
            sortindex[i]=temp;
        }
    }

    return sortindex;
}
```

4、 `public static int[] SortByWeight(float weight[],int N)`

```
public static int[] SortByWeight(float weight[],int N){
    int sortindex[]=new int[N];
    for(int i=0;i<N;i++)sortindex[i]=i;

    for(int i=0;i<N;i++){
        int max=i;
        for(int j=i+1;j<N;j++){
            if(weight[sortindex[max]]<weight[sortindex[j]])
                max=j;
        }
        if(max!=i){
            int temp=sortindex[max];
            sortindex[max]=sortindex[i];
            sortindex[i]=temp;
        }
    }

    return sortindex;
}
```

5、 `public static int[] SortByBMI(float bmi[],int N){`

```
public static int[] SortByBMI(float bmi[],int N){
    int sortindex[]=new int[N];
    for(int i=0;i<N;i++)sortindex[i]=i;

    for(int i=0;i<N;i++){
        int max=i;
        for(int j=i;j<N;j++){
            if(bmi[sortindex[max]]<bmi[sortindex[j]])
                max=j;
        }
        if(max!=i){
            int temp=sortindex[max];
            sortindex[max]=sortindex[i];
            sortindex[i]=temp;
        }
    }

    return sortindex;
}
```



Step2、在主函数中调用并分别观察结果：

1、按姓名：

```
<terminated> ControlStrcutureTest [Java Application] D:\Pro
请输入学生人数N:
3
请输入每个学生的学号、姓名、身高、体重:
331 ase 1.50 50
282 bty 1.70 60
199 cuu 1.60 70
199    cuu    1.60    70.0    27.34
282    bty    1.70    60.0    20.76
331    ase    1.50    50.0    22.22
```

2、按学号：

```
<terminated> ControlStrcutureTest [Java Application]
请输入学生人数N:
3
请输入每个学生的学号、姓名、身高、体重:
289 tyh 1.77 70
124 ppa 1.66 60
331 ase 1.50 50
331    ase    1.50    50.0    22.22
289    tyh    1.77    70.0    22.34
124    ppa    1.66    60.0    21.77
```

3、按身高：

```
<terminated> ControlStrcutureTest [Java Application] D:\Pro
请输入学生人数N:
3
请输入每个学生的学号、姓名、身高、体重:
331 ase 1.50 50
282 tty 1.70 60
199 uuu 1.60 70
282    tty    1.70    60.0    20.76
199    uuu    1.60    70.0    27.34
331    ase    1.50    50.0    22.22
```

4、按体重：

```
<terminated> ControlStrcutureTest [Java Application]
请输入学生人数N:
3
请输入每个学生的学号、姓名、身高、体重:
331 ase 1.50 50
282 tty 1.70 60
199 uuu 1.60 70
199    uuu    1.60    70.0    27.34
282    tty    1.70    60.0    20.76
331    ase    1.50    50.0    22.22
```

5、按 BMI：

```
<terminated> ControlStrcutreTest [Java Application] D:\Pro
请输入学生人数N:
3
请输入每个学生的学号、姓名、身高、体重:
331 ase 1.50 50
282 tty 1.70 60
199 uuu 1.60 70
199    uuu    1.60    70.0    27.34
331    ase    1.50    50.0    22.22
282    tty    1.70    60.0    20.76
```

- 5) 改编 BMI 程序, 增加 `printStudents` 函数, 该函数的参数为 `int sortedIndex[]`, 该函数可以打印排序前和排序后的结果。打印时, 每个学生的信息打印为一行, 为了清晰, 学号、姓名、身高、体重和计算后的 `bmi` 值之间用制表符(`\n`)隔开。

Step1. 增加 `printStudents` 函数函数如下:

```
public static void printStudents(String[] ids,String[] names,
    float[] heights,float[] weights,float[] bmis,int N,int sortindex[]){
    for(int i=0;i<N;i++){
        System.out.printf("%s\t%s\t%.2f\t%.1f\t%.2f\t\n",
            ids[sortindex[i]],names[sortindex[i]],heights[sortindex[i]],
            weights[sortindex[i]],bmis[sortindex[i]]);
    }
}
```

Step2. 在主函数中调用并观察结果: 该实验结果在 4) 中已完成,

故以按姓名排序为例, 其他结果均类似)

```
public static void main(String []arg){
    Scanner input=new Scanner(System.in);

    System.out.println("请输入学生人数N: ");
    int N=input.nextInt();

    String[] ids=new String[N];
    String[] names=new String[N];
    float[] heights=new float[N];
    float[] weights=new float[N];
    float[] bmis=new float[N];

    System.out.println("请输入每个学生的学号、姓名、身高、体重: ");
    for(int i=0;i<N;i++){
        ids[i]=input.next();
        names[i]=input.next();
        heights[i]=input.nextFloat();
        weights[i]=input.nextFloat();
        bmis[i]=weights[i]/(heights[i]*heights[i]);
    }
}
```

```
int sortindex[]=new int[N];
for(int i=0;i<N;i++)sortindex[i]=i;

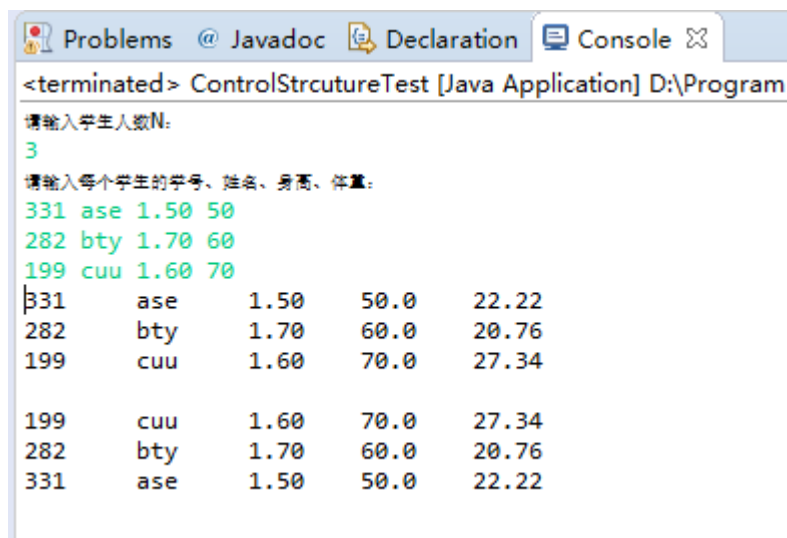
printStudents( ids, names, heights, weights, bmis, N, sortindex);

sortindex= SortByName(names,N);

printStudents( ids, names, heights, weights, bmis, N, sortindex);|

}
```

结果如下：该实验结果在 4) 中已完成，故以按姓名排序为例，其他结果均类似)



```
<terminated> ControlStrcutreTest [Java Application] D:\Program
请输入学生人数N:
3
请输入每个学生的学号、姓名、身高、体重:
331 ase 1.50 50
282 bty 1.70 60
199 cuu 1.60 70
331    ase    1.50    50.0    22.22
282    bty    1.70    60.0    20.76
199    cuu    1.60    70.0    27.34

199    cuu    1.60    70.0    27.34
282    bty    1.70    60.0    20.76
331    ase    1.50    50.0    22.22
```

- 6) 改编 BMI 程序，在 main 函数中，调用上述函数，输入至少 3 名学生，并按不同属性排序，并打印排序前、排序后的结果。

Step1. 该实验结果在 4) 5) 中已完成