# 实验一：Java 基本程序设计

## 一、实验目的

1）掌握标准输入输出函数的使用。
2）静态函数的定义和使用（本实验要求所有函数均为静态函数）；
2）掌握简单排序算法；
3）Java 基础语法综合运行（非面向对象版本 BMI 程序）；

## 二、实验内容

1)编写 BMI 类，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、计算后的 bmi 值和胖瘦健康状况。注意，上述数值均需保留两位小数存储。

2）定义 inputStudents 函数，该函数的参数为上述数组，该函数的功能是输入多个学生的相关信息，并将相关数据存储到上述数组中；

3）在 BMI 类中，增加一个函数 checkHealth，函数参数为 bmi 值，该函数按下表中 BMI 取值范围判断胖瘦健康状况，该函数的返回值为字符串，返回结果即下表中的第一列中的值，并在 inputStudents 函数中调用该函数，获得学生的胖瘦健康状况。

| Category | BMI (kg/m$^2$) | |
|---|---|---|
| | from | to |
| Underweight | | 18.5 |
| Normal Range | 18.5 | 23 |
| Overweight—At Risk | 23 | 25 |
| Overweight—Moderately Obese | 25 | 30 |
| Overweight—Severely Obese | 30 | |

4）在 BMI 类中，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。**

5）在 BMI 类中，增加 printStudents 函数，该函数的参数含有 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(\t)隔开。

6）定义 menu 函数，提供输入学生、打印学生，5 种排序、程序退出等 8 种选项，用户输入指定选项后，运行相应函数功能。**注意，在调用 inputStudents 函数前，需先提示用户数输入指定人数。**

7）在 BMI 类的 main 函数中，调用 menu 函数，测试运行各项功能。

**注意，身高、体重、及 bmi 等数值均需保留两位小数的格式进行存储和显示。**

## 三、实验代码

**程序代码：**

```
/**
 * 2018.7.14
 */
/**
 * @author gyl-天堂牧场
 *
 */
package edu.hit.java.exp1.hit1170300621;

import java.util.Scanner;

public class BMI
{
    static Scanner in = new Scanner(System.in);

    public static int StudentsSum; //记录总人数

    public static void main(String[] args)
    {
        //定义变量学号，姓名，身高，体重和BMI
        String[] ids = new String[3000];
        String[] names = new String[3000];
        float[] heights = new float[3000];
        float[] weights = new float[3000];
        float[] bmis = new float[3000];
        int[] sortedIndex = new int[3000];

        //初始化
        StudentsSum = 0;

        //欢迎界面
        System.out.println("Welcome To The Students' Healthy Information
System!");
        System.out.println();
        menu(ids, names, heights, weights, bmis, sortedIndex); //执行操作
        in.close();
    }

    //主菜单程序
    public static void menu(String[] ids, String[] names, float[] heights,
float[] weights, float[] bmis, int[] sortedIndex)
```

```
    {
        //提示输入信息
        System.out.println("1. Input students' information"); //输入学生信息
        System.out.println("2. Print students' information"); //打印学生信息
        System.out.println("3. Sort the students by IDs"); //按照学生的学号进
行排序
        System.out.println("4. Sort the students by Names"); //按照学生的姓名
进行排序
        System.out.println("5. Sort the students by Heights"); //按照学生的身
高进行排序
        System.out.println("6. Sort the students by Weights"); //按照学生的体
重进行排序
        System.out.println("7. Sort the students by BMIs"); //按照学生的BMI值
进行排序
        System.out.println("8. Exit the students' healthy information system");
//退出系统
        System.out.println();

        int input;
        do
        {
            //输入想进行的操作
            System.out.print("Please input the number you want to do: ");
            input = in.nextInt();
            System.out.println();

            switch(input)
            {
            case 1:
                inputStudents(ids, names, heights, weights, bmis,
sortedIndex);
                break;
            case 2:
                printStudents(ids, names, heights, weights, bmis,
sortedIndex);
                break;
            case 3:
                sortByIDs(ids, sortedIndex, 1, StudentsSum);
                break;
            case 4:
                sortByNames(names, sortedIndex, 1, StudentsSum);
                break;
            case 5:
                sortByHeights(heights, sortedIndex, 1, StudentsSum);
```

```
                break;
            case 6:
                sortByWeights(weights, sortedIndex, 1, StudentsSum);
                break;
            case 7:
                sortByBMI(bmis, sortedIndex, 1, StudentsSum);
                break;
            case 8:
                System.out.println("Goodbye! Thank you for using.");
                break;
            default:
                System.out.print("You input the wrong number. Please input
again.");
                break;
            }
        } while(input != 8);
    }

    //Case 1: 输入学生信息
    public static void inputStudents(String[] ids, String[] names, float[]
heights, float[] weights, float[] bmis, int[] sortedIndex)
    {
        //输入要输入的学生的个数
        System.out.print("Please input the numbers of the students: ");
        int num = in.nextInt();
        System.out.println();

        //对每一个学生的信息进行输入
        for(int i = StudentsSum + 1; i <= StudentsSum + num; i++)
        {
            //输入学号
            System.out.print("Please input the ID of the No." + i + " student:
");
            ids[i] = in.next();
            in.nextLine();

            //输入姓名
            System.out.print("Please input the name of the No." + i + " student:
");
            names[i] = in.nextLine();

            //输入身高
            System.out.print("Please input the height of the No." + i + " student:
");
```

```
            heights[i] = in.nextFloat();

            //输入体重
            System.out.print("Please input the weight of the No." + i + " student:
");
            weights[i] = in.nextFloat();

            //计算BMI
            bmis[i] = weights[i] / (heights[i] * heights[i]);
            checkHealth(bmis[i]);
            System.out.println();

            sortedIndex[i] = i; //初始化学生的位置
        }

        StudentsSum += num;
    }


    //BMI值的对应输出判断
    public static void checkHealth(float bmis)
    {
        if(bmis <= 18.5)
            System.out.println("Underweight");
        else if(bmis <= 23)
            System.out.println("Normal Range");
        else if(bmis <= 25)
            System.out.println("Overweight--At Risk");
        else if(bmis <= 30)
            System.out.println("Overweight--Moderately Obese");
        else
            System.out.println("Overweight--Severely Obese");
    }


    //Case 2: 打印学生信息
    public static void printStudents(String[] ids, String[] names, float[]
heights, float[] weights, float[] bmis, int[] sortedIndex)
    {
        for(int i = 1; i <= StudentsSum; i++)
            System.out.printf(ids[sortedIndex[i]] + "\t" +
names[sortedIndex[i]] + "\t%.2f\t%.2f\t%.2f\n", heights[sortedIndex[i]],
weights[sortedIndex[i]], bmis[sortedIndex[i]]);
    }
```

```
    /*
     * 排序整体应用快速排序的思想，将时间复杂度由n^2降到了nlogn
     * 其实也考虑过用堆排序（即二分）的思想，但用主席树维护的区间值的修改太过繁琐，
便放弃了
     * 具体的算法思想就是比较身高、体重等函数，调整sortedIndex[]的值
     */

    //Case 3: 按照学生的学号进行排序
    public static void sortByIDs(String[] ids, int[] sortedIndex, int begin,
int end)
    {
        if(begin >= end)
            return;
        int left = begin;
        int right = end;
        int tmp = sortedIndex[left];
        while(left < right)
        {
            while(left < right &&
ids[sortedIndex[right]].compareTo(ids[tmp]) >= 0)
                right--;
            if(left < right)
                sortedIndex[left++] = sortedIndex[right];
            while(left < right && ids[sortedIndex[left]].compareTo(ids[tmp])
<= 0)
                left++;
            if(left < right)
                sortedIndex[right--] = sortedIndex[left];
        }
        sortedIndex[left] = tmp;
        sortByIDs(ids, sortedIndex, begin, left - 1);
        sortByIDs(ids, sortedIndex, left + 1, end);
    }

    //Case 4: 按照学生的姓名进行排序
    public static void sortByNames(String[] names, int[] sortedIndex, int begin,
int end)
    {
        if(begin >= end)
            return;
        int left = begin;
        int right = end;
        int tmp = sortedIndex[left];
```

```
        while(left < right)
        {
            while(left < right &&
names[sortedIndex[right]].compareTo(names[tmp]) >= 0)
                right--;
            if(left < right)
                sortedIndex[left++] = sortedIndex[right];
            while(left < right &&
names[sortedIndex[left]].compareTo(names[tmp]) <= 0)
                left++;
            if(left < right)
                sortedIndex[right--] = sortedIndex[left];
        }
        sortedIndex[left] = tmp;
        sortByNames(names, sortedIndex, begin, left - 1);
        sortByNames(names, sortedIndex, left + 1, end);
    }

    //Case 5: 按照学生的身高进行排序
    public static void sortByHeights(float[] heights, int[] sortedIndex, int
begin, int end)
    {
        if(begin >= end)
            return;
        int left = begin;
        int right = end;
        int tmp = sortedIndex[left];
        while(left < right)
        {
            while(left < right && heights[sortedIndex[right]] >= heights[tmp])
                right--;
            if(left < right)
                sortedIndex[left++] = sortedIndex[right];
            while(left < right && heights[sortedIndex[left]] <= heights[tmp])
                left++;
            if(left < right)
                sortedIndex[right--] = sortedIndex[left];
        }
        sortedIndex[left] = tmp;
        sortByHeights(heights, sortedIndex, begin, left - 1);
        sortByHeights(heights, sortedIndex, left + 1, end);
    }

    //Case 6: 按照学生的体重进行排序
```

```
    public static void sortByWeights(float[] weights, int[] sortedIndex, int
begin, int end)
    {
        if(begin >= end)
            return;
        int left = begin;
        int right = end;
        int tmp = sortedIndex[left];
        while(left < right)
        {
            while(left < right && weights[sortedIndex[right]] >= weights[tmp])
                right--;
            if(left < right)
                sortedIndex[left++] = sortedIndex[right];
            while(left < right && weights[sortedIndex[left]] <= weights[tmp])
                left++;
            if(left < right)
                sortedIndex[right--] = sortedIndex[left];
        }
        sortedIndex[left] = tmp;
        sortByWeights(weights, sortedIndex, begin, left - 1);
        sortByWeights(weights, sortedIndex, left + 1, end);
    }

    //Case 7: 按照学生的BMI值进行排序
    public static void sortByBMI(float[] bmis, int[] sortedIndex, int begin,
int end)
    {
        if(begin >= end)
            return;
        int left = begin;
        int right = end;
        int tmp = sortedIndex[left];
        while(left < right)
        {
            while(left < right && bmis[sortedIndex[right]] >= bmis[tmp])
                right--;
            if(left < right)
                sortedIndex[left++] = sortedIndex[right];
            while(left < right && bmis[sortedIndex[left]] <= bmis[tmp])
                left++;
            if(left < right)
                sortedIndex[right--] = sortedIndex[left];
        }
```

```
        sortedIndex[left] = tmp;
        sortByBMI(bmis, sortedIndex, begin, left - 1);
        sortByBMI(bmis, sortedIndex, left + 1, end);
    }
}
```

**运行结果：**

```
Welcome To The Students' Healthy Information System!

1. Input students' information
2. Print students' information
3. Sort the students by IDs
4. Sort the students by Names
5. Sort the students by Heights
6. Sort the students by Weights
7. Sort the students by BMIs
8. Exit the students' healthy information system

Please input the number you want to do: 1

Please input the numbers of the students: 4

Please input the ID of the No.1 student: 4
Please input the name of the No.1 student: Bi
Please input the height of the No.1 student: 1.73
Please input the weight of the No.1 student: 62.4
Normal Range

Please input the ID of the No.2 student: 2
Please input the name of the No.2 student: Huang
Please input the height of the No.2 student: 1.75
Please input the weight of the No.2 student: 60.1
Normal Range

Please input the ID of the No.3 student: 1
Please input the name of the No.3 student: Guo
Please input the height of the No.3 student: 1.68
Please input the weight of the No.3 student: 70.8
Overweight--Moderately Obese

Please input the ID of the No.4 student: 3
Please input the name of the No.4 student: Gao
Please input the height of the No.4 student: 1.80
```

```
Please input the weight of the No.4 student: 53.7
Underweight

Please input the number you want to do: 4

Please input the number you want to do: 2

4   Bi      1.73    62.40   20.85
3   Gao     1.80    53.70   16.57
1   Guo     1.68    70.80   25.09
2   Huang   1.75    60.10   19.62

Please input the number you want to do: 1

Please input the numbers of the students: 6

Please input the ID of the No.5 student: 5
Please input the name of the No.5 student: Zhuang
Please input the height of the No.5 student: 1.62
Please input the weight of the No.5 student: 52.9
Normal Range

Please input the ID of the No.6 student: 9
Please input the name of the No.6 student: Yang
Please input the height of the No.6 student: 1.82
Please input the weight of the No.6 student: 83.4
Overweight--Moderately Obese

Please input the ID of the No.7 student: 6
Please input the name of the No.7 student: Kang
Please input the height of the No.7 student: 1.76
Please input the weight of the No.7 student: 51.5
Underweight

Please input the ID of the No.8 student: 7
Please input the name of the No.8 student: Liang
Please input the height of the No.8 student: 1.83
Please input the weight of the No.8 student: 90.2
Overweight--Moderately Obese

Please input the ID of the No.9 student: 10
Please input the name of the No.9 student: Jiang
Please input the height of the No.9 student: 1.78
Please input the weight of the No.9 student: 88.9
```

```
Overweight--Moderately Obese

Please input the ID of the No.10 student: 8
Please input the name of the No.10 student: Zhao
Please input the height of the No.10 student: 1.72
Please input the weight of the No.10 student: 71.0
Overweight--At Risk

Please input the number you want to do: 5

Please input the number you want to do: 2

5   Zhuang  1.62    52.90   20.16
1   Guo     1.68    70.80   25.09
8   Zhao    1.72    71.00   24.00
4   Bi      1.73    62.40   20.85
2   Huang   1.75    60.10   19.62
6   Kang    1.76    51.50   16.63
10  Jiang   1.78    88.90   28.06
3   Gao     1.80    53.70   16.57
9   Yang    1.82    83.40   25.18
7   Liang   1.83    90.20   26.93

Please input the number you want to do: 6

Please input the number you want to do: 2

6   Kang    1.76    51.50   16.63
5   Zhuang  1.62    52.90   20.16
3   Gao     1.80    53.70   16.57
2   Huang   1.75    60.10   19.62
4   Bi      1.73    62.40   20.85
1   Guo     1.68    70.80   25.09
8   Zhao    1.72    71.00   24.00
9   Yang    1.82    83.40   25.18
10  Jiang   1.78    88.90   28.06
7   Liang   1.83    90.20   26.93

Please input the number you want to do: 3

Please input the number you want to do: 2

1   Guo     1.68    70.80   25.09
10  Jiang   1.78    88.90   28.06
```

```
2   Huang   1.75    60.10   19.62
3   Gao     1.80    53.70   16.57
4   Bi      1.73    62.40   20.85
5   Zhuang  1.62    52.90   20.16
6   Kang    1.76    51.50   16.63
7   Liang   1.83    90.20   26.93
8   Zhao    1.72    71.00   24.00
9   Yang    1.82    83.40   25.18

Please input the number you want to do: 8

Goodbye! Thank you for using.
```