实验一：Java 基本程序设计

一、实验目的

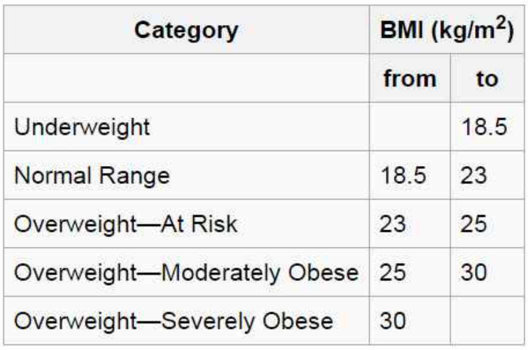
1）掌握标准输入输出函数的使用。
2）静态函数的定义和使用（本实验要求所有函数均为静态函数）；
2）掌握简单排序算法；
3）Java 基础语法综合运行（非面向对象版本 BMI 程序）；

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

注意：将程序代码和运行结果截图粘贴在此处，注意源代码中注释行数不少于全部代码的 **1/3**，程序源代码请压缩后上传，压缩文件按照 学号**.zip** 进行命名，注意源程序于报告请分别上传到不同的文件夹中！

```java
package edu.hit.java.exp1.L170300502;
//郑民采

import java.util.Scanner;

public class HitL170300502 {

private static int stunum=0;

public static void main(String[] args) {
HitL170300502 bmi = new HitL170300502();

String[] ids = new String[100];
String[] names = new String[100];
float[] heights = new float[100];
float[] weights = new float[100];
float[] bmis = new float[100];
int[] sort = new int[100];









selectMenu(ids,names,heights,bmis,weights,sort);


}

public static void selectMenu(String[] ids, String[] names, float[]
heights,float[] bmis ,float[] weights,int[] sort) {


    Scanner scr = new Scanner(System.in);
    printMenu();


 int menuId = scr.nextInt();

if(menuId == 1) {
     inputInf(ids, names, heights,bmis,weights, sort);

} else if(menuId == 2) {
     outputInf(ids, names, heights,bmis,weights, sort);

}
else if(menuId == 3) {
    idSort(ids,names, heights, bmis, weights, sort);
}
else if(menuId == 4) {
    nameSort(ids,names, heights, bmis, weights, sort);
}
else if(menuId == 5) {
    heightSort(ids,names, heights, bmis, weights, sort);
}
else if(menuId == 6) {
    weightSort(ids,names, heights, bmis, weights, sort);
}
else if(menuId == 7) {
    bmiSort(ids,names, heights, bmis, weights, sort);
}
else if(menuId == 8) {

    System.out.println("byebye");
}

scr.close();
}



public static void printMenu() {
System.out.println("select menu : \n");
System.out.println("1.input students' information\n");
System.out.println("2.output students'information\n");
System.out.println("3. Sort the information of IDs\n");
System.out.println("4. Sort the information of Names\n");
System.out.println("5. Sort the information of Heights\n");
System.out.println("6. Sort the information of Weights\n");
System.out.println("7. Sort the information of BMIs\n");
System.out.println("8. Exit the System\n");
}

public static void inputInf(String[] ids, String[] names, float[]
heights,float[] bmis ,float[] weights,int[] sort) {


    System.out.println("Input the number of students : ");
    Scanner scr = new Scanner(System.in);
    stunum=scr.nextInt();




for(int i = 0; i < stunum ; i++) {
System.out.println("Input information of "+ (i + 1) + "/" + stunum + " student");

System.out.println("Input ID:\n");
ids[i]=scr.next();

System.out.println("Input NAME:\n");
names[i]=scr.next();

System.out.println("Input HEIGHT:\n");
heights[i]=scr.nextFloat();

System.out.println("Input WEIGHT:\n");
weights[i]=scr.nextFloat();

bmis[i]=weights[i]/((heights[i]/10)*(heights[i]/10));

checkbmi(bmis[i]);
sort[i]=i;

}

selectMenu(ids,names,heights,bmis,weights,sort);




scr.close();
```

```
}

public static void outputInf(String[] ids, String[] names, float[]
heights,float[] bmis ,float[] weights,int[] sort) {



    for(int i = 0; i < stunum ; i++) {

        System.out.println((i+1)+":");
        System.out.printf(ids[i]+"\n");
        System.out.printf(names[i]+"\n");
        System.out.printf(heights[i]+"\n");
        System.out.printf(weights[i]+"\n");
        System.out.printf(bmis[i]+"\n");
        checkbmi(bmis[i]);


    }
    selectMenu(ids,names,heights,bmis,weights,sort);

}
public static void checkbmi(float bmis){
    if(bmis<=18.5) {
        System.out.println("Underweight\n");
    }
    else if(bmis<=23) {
        System.out.println("Normal\n");
    }
    else if(bmis<=25) {
        System.out.println("Overweight\n");
    }
    else if(bmis<=30) {
        System.out.println("obesity\n");
    }
    else {
        System.out.println("extremely obese\n");
    }


}
public static void idSort(String[] ids, String[] names, float[] heights,float[]
bmis ,float[] weights,int[] sort) {



    for(int i =0 ; i<stunum ; i++)
    {
        for(int j=1 ;j<stunum;j++)
        {
            if(ids[i].compareTo(ids[j])>0)  {
                String tmp = ids[i];
                ids[i]= ids[j];
                ids[j] = tmp;
            }
        }
    }
    System.out.println("To check for aligned informaiton :\n");
    outputInf(ids, names, heights,bmis,weights, sort);
    selectMenu(ids,names,heights,bmis,weights,sort);



}
public static void nameSort(String[] ids, String[] names, float[]
heights,float[] bmis ,float[] weights,int[] sort) {

    for(int i =0 ; i<stunum ; i++)
    {
        for(int j=1 ;j<stunum;j++)
        {
            if(ids[i].compareTo(names[j])>0)  {
                String tmp = names[i];
                names[i]= names[j];
                names[j] = tmp;
            }
        }
    }
    System.out.println("To check for aligned informaiton :\n");
    outputInf(ids, names, heights,bmis,weights, sort);
    selectMenu(ids,names,heights,bmis,weights,sort);

}
public static void heightSort(String[] ids, String[] names, float[]
heights,float[] bmis ,float[] weights,int[] sort) {
    for(int i =0 ; i<stunum ; i++)
    {
        for(int j=1 ;j<stunum;j++)
        {
            if(j>i &&  heights[i]> heights[j])  {
                float tmp = heights[i];
                heights[i]= heights[j];
                heights[j] = tmp;
            }
        }
    }
    System.out.println("To check for aligned informaiton :\n");
    outputInf(ids, names, heights,bmis,weights, sort);
    selectMenu(ids,names,heights,bmis,weights,sort);

}
public static void weightSort(String[] ids, String[] names, float[]
heights,float[] bmis ,float[] weights,int[] sort) {

    for(int i =0 ; i<stunum ; i++)
    {
        for(int j=1 ;j<stunum;j++)
        {
            if(j>i && weights[i]>weights[j])  {
                float tmp = weights[i];
                weights[i]= weights[j];
                weights[j] = tmp;
            }
        }
    }
    System.out.println("To check for aligned informaiton :\n");
    outputInf(ids, names, heights,bmis,weights, sort);
    selectMenu(ids,names,heights,bmis,weights,sort);

}
public static void bmiSort(String[] ids, String[] names, float[]
heights,float[] bmis ,float[] weights,int[] sort) {

    for(int i =0 ; i<stunum ; i++)
    {
        for(int j=1 ;j<stunum;j++)
        {
            if(j>i && bmis[i]>bmis[j])  {
                float tmp = bmis[i];
                bmis[i]= bmis[j];
                bmis[j] = tmp;
            }
        }
    }
    System.out.println("To check for aligned informaiton :\n");
    outputInf(ids, names, heights,bmis,weights, sort);
    selectMenu(ids,names,heights,bmis,weights,sort);

}




}
```

```
select menu :

1.input students' information
2.output students'information
3. Sort the information of IDs
4. Sort the information of Names
5. Sort the information of Heights
6. Sort the information of Weights
7. Sort the information of BMIs
8. Exit the System

1
Input the number of students :
3
Input information of 1/3 student
Input ID:

1111122222
Input NAME:

zmc
Input HEIGHT:

170
Input WEIGHT:

60
Normal
```

```
Input information of 2/3 student
Input ID:

1111155555
Input NAME:

nty
Input HEIGHT:

180
Input WEIGHT:

90
obesity

Input information of 3/3 student
Input ID:

ywx
Input NAME:

172
Input HEIGHT:

70
Input WEIGHT:

80
extremely obese

select menu :

1.input students' information
2.output students'information
3. Sort the information of IDs
4. Sort the information of Names

5. Sort the information of Heights
6. Sort the information of Weights
7. Sort the information of BMIs
8. Exit the System

3
```

```
To check for aligned informaiton :

1:
1111122222
zmc
170.0
60.0
20.761246
Normal

2:
ywx
nty
180.0
90.0
27.77778
obesity

3:
1111155555
172
70.0
80.0
163.26532
extremely obese

select menu :

1.input students' information
2.output students'information
3. Sort the information of IDs
4. Sort the information of Names
5. Sort the information of Heights
6. Sort the information of Weights
7. Sort the information of BMIs
8. Exit the System

4
To check for aligned informaiton :

1:
1111122222
zmc
170.0
```

```
60.0
20.761246
Normal

2:
ywx
172
180.0
90.0
27.77778
obesity

3:
1111155555
nty
70.0
80.0
163.26532
extremely obese

select menu :

1.input students' information
2.output students'information
3. Sort the information of IDs
4. Sort the information of Names
5. Sort the information of Heights
6. Sort the information of Weights
7. Sort the information of BMIs
8. Exit the System

5
To check for aligned informaiton :

1:
1111122222
zmc
70.0
60.0
20.761246
Normal

2:
ywx
```

```
172
170.0
90.0
27.77778
obesity

3:
1111155555
nty
180.0
80.0
163.26532
extremely obese

select menu :

1.input students' information
2.output students'information
3. Sort the information of IDs
4. Sort the information of Names
5. Sort the information of Heights
6. Sort the information of Weights
7. Sort the information of BMIs
8. Exit the System

6
To check for aligned informaiton :

1:
1111122222
zmc
70.0
60.0
20.761246
Normal

2:
ywx
172
170.0
80.0
27.77778
obesity
```

```
3:
1111155555
nty
180.0
90.0
163.26532
extremely obese

select menu :
1.input students' information
2.output students'information
3. Sort the information of IDs
4. Sort the information of Names
5. Sort the information of Heights
6. Sort the information of Weights
7. Sort the information of BMIs
8. Exit the System

7
To check for aligned informaiton :

1:
1111122222
zmc
70.0
60.0
20.761246
Normal

2:
ywx
172
170.0
80.0
27.77778
obesity

3:
1111155555
nty
180.0
90.0
163.26532
extremely obese
```

```
select menu :
1.input students' information
2.output students'information
3. Sort the information of IDs
4. Sort the information of Names
5. Sort the information of Heights
6. Sort the information of Weights
7. Sort the information of BMIs
8. Exit the System
```