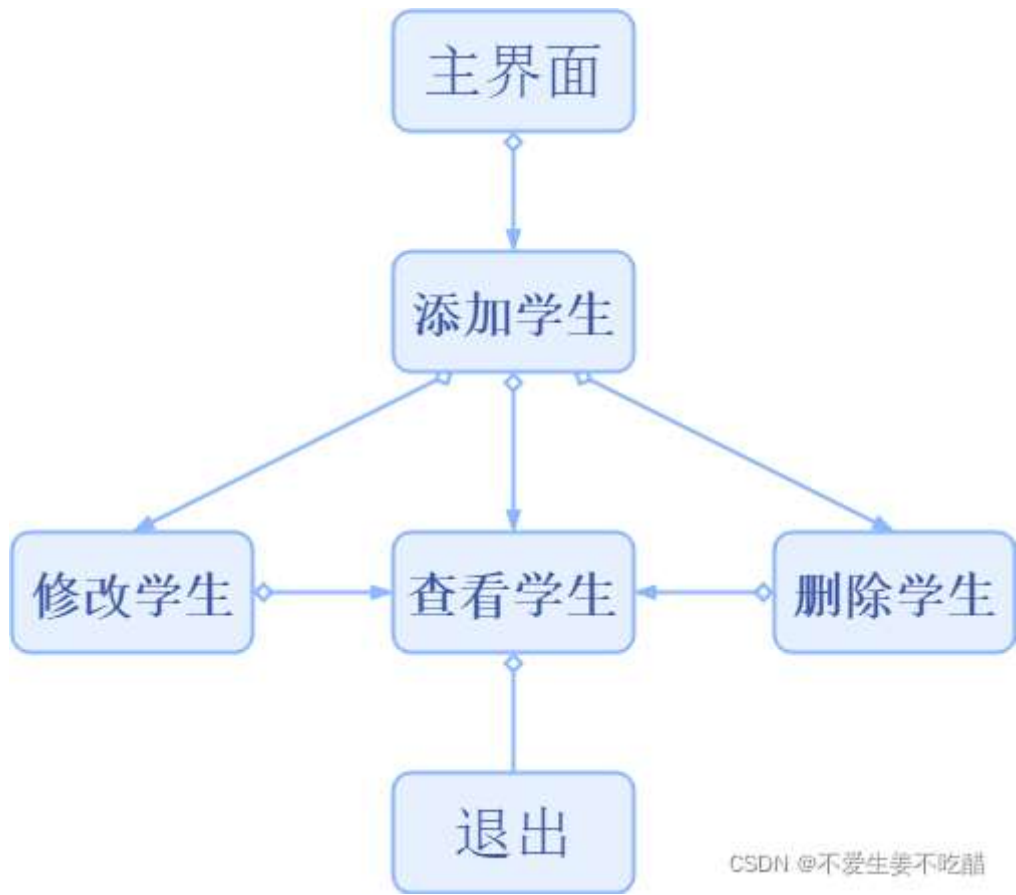


## 🌴一、学籍管理系统功能介绍

### 1.介绍

学籍管理系统首先有外部的主界面展示功能，进入主界面后可以选择想要的操作，如**添加学生信息**、**修改学生信息**、**查找学生信息**、**删除学生信息**、**查看学生信息**和**退出系统操作**。

### 2.学籍管理系统的主要功能图示



### 3.代码编写的思路

- 1.1 定义学生类
- 1.2 主界面的代码编写
- 1.3 添加学生的代码编写
- 1.4 查看学生的代码编写
- 1.5 删除学生的代码编写
- 1.6 修改学生的代码编写

## 🌴二、定义学生类

### 1. 学生类名称：Student

### 2. 包含的成员变量

学号: num  
姓名: name  
年龄: age  
地址: address

### 3.构造方法

无参构造

带四个参数的构造

### 4.成员方法

每个成员变量给出对应的get/set方法

### 5.代码示例

```
1 public class Student {  
2     //不爱生姜不吃醋  
3     private String num;//学生学号  
4     private String name;//学生姓名  
5     private String age;//学生年龄  
6     private String address;//学生地址  
7  
8     public Student() {  
9         }//无参构造  
10  
11     public Student(String num, String name, String age, String address) {  
12         this.num = num;  
13         this.name = name;  
14         this.age = age;  
15         this.address = address;  
16     }//带四个参数的构造  
17  
18     public String getNum() {  
19         return num;  
20     }  
21  
22     public void setNum(String num) {  
23         this.num = num;  
24     }  
25  
26     public String getName() {  
27         return name;  
28     }  
29  
30     public void setName(String name) {  
31
```

```
32         this.name = name;
33     }
34
35     public String getAge() {
36         return age;
37     }
38
39     public void setAge(String age) {
40         this.age = age;
41     }
42
43     public String getAddress() {
44         return address;
45     }
46
47     public void setAddress(String address) {
48         this.address = address;
49     }
49 }
```

## 🌴三.主界面的代码编写

### 1.思路

- 新建一个主类（*Studentdemo*）
- 用输出语句完成主界面的编写
- 用 `Scanner` 实现键盘录入数据
- 用 `switch` 语句完成操作的选择
- 用循环完成再次回到主界面
- 循环之中包含**添加、查找、删除、修改**的方法

### 2.代码示例

主类：主界面的编写

```
1 public class Studentdemo {
2     public static void main(String[] args){
3         //不爱生姜不吃醋
4         ArrayList<Student> array=new ArrayList<>();
5         while(true) {
6             System.out.println("-----学生管理系统-----");
7             System.out.println("1.添加学生");
8         }
9     }
10 }
```

```
9      System.out.println("2. 删除学生");
10     System.out.println("3. 修改学生");
11     System.out.println("4. 查看所有学生");
12     System.out.println("5. 退出");
13     System.out.println("请选择你需要的操作: ");
14
15     Scanner sc = new Scanner(System.in);
16     String line = sc.nextLine();
17
18     switch (line) {
19         case "1":
20             /** 此处调用“添加学生方法” **
21             break;
22         case "2":
23             /** 此处调用“删除学生方法” **
24             break;
25         case "3":
26             /** 此处调用“修改学生方法” **
27             break;
28         case "4":
29             /** 此处调用“查找学生方法” **
30             break;
31         case "5":
32             System.out.println("谢谢使用! ");
33             System.exit(0);
34     }
35 }
36 }
```

## 四、系统功能方法的实现

### 1. 添加学生信息的代码编写

#### 1.1 思路

1. 键盘录入选择**添加**学生;
2. 定义一个方法用于添加学生;
  - 显示提示信息
  - 键盘录入学生对象信息所需要的数据
  - 创建学生对象, 将录入的数据赋值给学生对象的成员变量
  - 将学生对象添加到集合中

- 给出添加成功提示

### 3. 调用方法

#### 1.2 添加学生信息的方法说明图



#### 1.3 代码示例

添加学生方法

```
1 public static void Addstudent(ArrayList<Student> array){
2     // 不爱生姜不吃醋
3     // 定义一个方法用于添加学生;
4     Scanner sc=new Scanner(System.in);
5
6     System.out.print("请输入学生学号: ");
7     String num=sc.nextLine();
8 }
```

```
9      System.out.print("请输入学生姓名: ");
10     String name=sc.nextLine();
11     System.out.print("请输入学生年龄: ");
12     String age=sc.nextLine();
13     System.out.print("请输入学生地址: ");
14     String address=sc.nextLine();
15
16     // 创建学生对象, 将键盘录入的数据赋值给学生对象的成员变量
17     Student s=new Student();
18     s.setNum(num);
19     s.setName(name);
20     s.setAge(age);
21     s.setAddress(address);
22
23     // 将学生对象添加到集合中
24     array.add(s);
25
26     // 给出添加成功提示
27     System.out.println("添加学生成功!");
    }
```

添加学生方法代码写完后, 注意要在 `main` 方法中调用: `Addstudent(array);`

## 2.查看学生信息的代码编写

### 2.1 思路

1. 键盘录入选择**查看**所有学生信息;
2. 定义一个方法, 用于查看学生信息;
  - 显示表头信息
  - 将集合中数据取出按照对应格式显示学生信息
3. 调用方法。

### 2.2 查看学生信息的方法说明图

ArrayList

➔

-----欢迎来到学籍管理系统-----

1.添加学生  
2.删除学生  
3.修改学生  
4.查看学生  
5.退出  
请输入你的选择：  
4

学号	姓名	年龄	居住地
001	张三	30	芜湖

CSDN @不爱吃醋不吃醋

## 2.3 代码示例

查看学生方法

```
1 // 定义一个方法, 用于查看学生信息;
2 public static void Findallstudent(ArrayList<Student> array){
3     if(array.size()==0){
4         System.out.println("无信息!");
5         return;
6     }
7     // 显示表头信息
8     // 将集合中数据取出按照对应格式显示学生信息
9     for(int i=0;i<array.size();i++){
10         Student s=array.get(i);
11         System.out.println("学号: "+s.getNum());
12         System.out.println("姓名: "+s.getName());
13         System.out.println("年龄: "+s.getAge());
14         System.out.println("地址: "+s.getAddress());
15         System.out.println(" ");
16     }
17 }
```

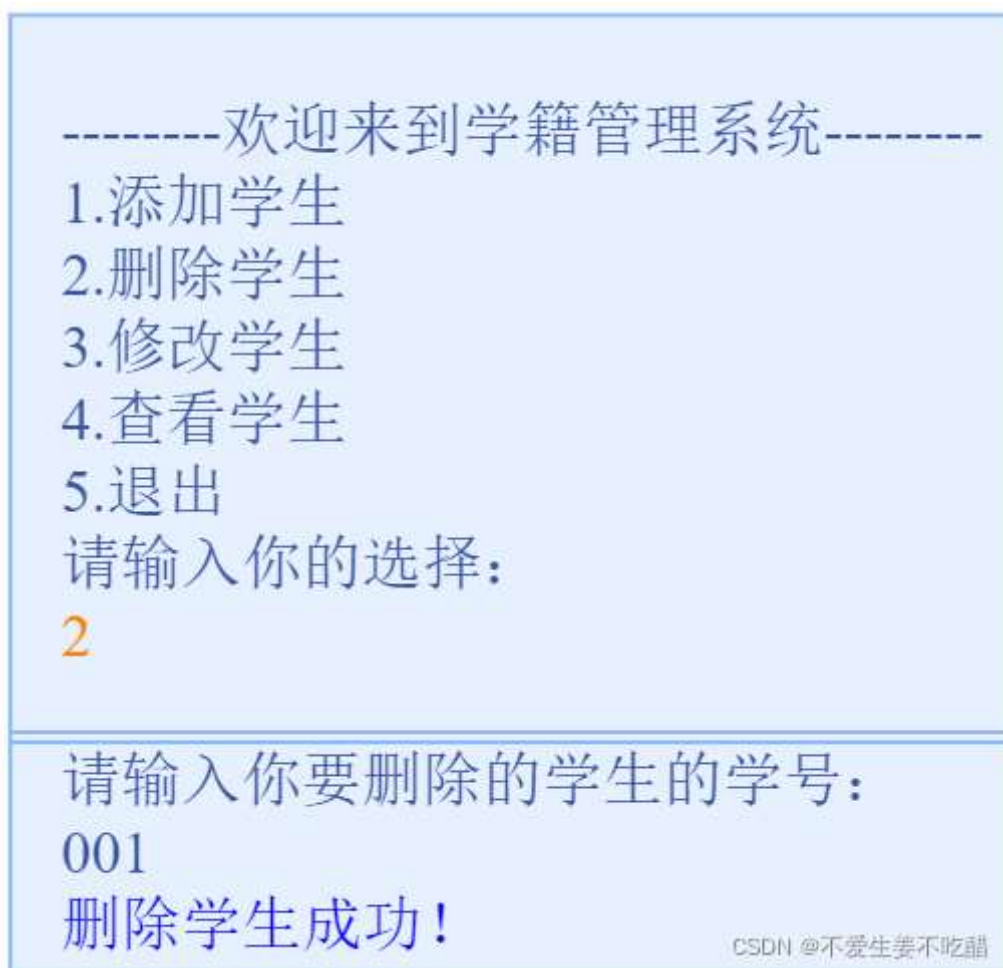
查看学生方法代码写完后, 注意要在 main 方法中调用: Findallstudent(array);

## 3.删除学生信息的代码编写

### 3.1 思路

1. 键盘录入要删除的学生信息；
2. 定义一个方法，用于删除学生信息；
  - 显示提示信息
  - 键盘录入要删除的学生学号
  - 遍历集合将对应学生对象从集合中删除
  - 给出删除成功提示
3. 调用方法

### 3.2 删除学生信息的方法说明图



### 3.3 代码示例

#### 删除学生方法

```
1 // 定义一个方法，用于删除学生信息；
2 public static void Deletestudent(ArrayList<Student> array){
3     // 键盘录入要删除的学生信息，显示提示信息
4     Scanner sc=new Scanner(System.in);
5     System.out.println("请输入要删除学生的学号：");
6     String num=sc.nextLine();
7
8     int index = -1;
9 }
```



```
10
11      // 遍历集合将对应学生对象从集合中删除
12      for(int i=0;i<array.size();i++){
13          Student s=array.get(i);
14          if(s.getNum().equals(num)){
15              index = i;
16              break;
17          }
18      }
19
20      if(index == -1){
21          System.out.println("学号有误！");
22      }else{
23          array.remove(index);
24          System.out.println("删除学生信息成功！");
25      }
    }
```

删除学生方法代码写完后，注意要在 `main` 方法中调用：`Deletestudent (array);`

## 4.修改学生的代码编写

### 4.1 思路

1. 键盘录入选择**修改**的学生信息；
2. 定义一个方法，用于修改学生信息；
  - 显示提示信息
  - 键盘录入要修改的学生学号
  - 遍历集合修改对应的学生信息
  - 给出修改成功提示
3. 调用方法

### 4.2 修改学生信息的方法说明图

-----欢迎来到学籍管理系统-----

- 1.添加学生
- 2.删除学生
- 3.修改学生
- 4.查看学生
- 5.退出

请输入你的选择：

2

请输入你要修改的学生的学号：

001

请输入新的学生姓名：

李四

请输入新的学生年龄：

40

请输入新的学生居住地：

开冲

修改学生信息成功！

CSDN @不爱生姜不吃醋

## 4.3 代码示例

修改学生方法

```
1 // 定义一个方法，用于修改学生信息；
2 public static void Updatestudent(ArrayList<Student> array){
3 //不爱生姜不吃醋
4 // 键盘录入选择修改的学生信息，显示提示信息
5 Scanner sc=new Scanner(System.in);
6 System.out.print("请输入要修改的学生学号：");
7 String num=sc.nextLine();
8
9 // 键盘录入要修改的学生信息
10
```

```
11      System.out.print("请输入修改后的学生姓名");
12      String name=sc.nextLine();
13      System.out.print("请输入修改后的学生年龄");
14      String age=sc.nextLine();
15      System.out.print("请输入修改后的学生地址");
16      String address=sc.nextLine();
17
18      //创建学生对象
19      Student s=new Student();
20      s.setNum(num);
21      s.setName(name);
22      s.setAge(age);
23      s.setAddress(address);
24      //遍历集合修改对应的学生信息
25      for(int i=0;i<array.size();i++){
26          Student student=array.get(i);
27          if(student.getNum().equals(num)){
28              array.set(i,s);
29              break;
30          }
31      }
32      //修改成功提示
33      System.out.println("修改成功!");
    }
```

修改学生信息方法代码写完后，注意要在 `main` 方法中调用：`Updatestudent (array);`

## 5.学号重复问题

### 5.1 思路

1. 定义一个方法，对学号是否被使用进行判断；
  - 如果与集合中的某一个学生学号相同，返回true
  - 如果都不相同，返回false;
2. 在添加学生录入学号后调用该方法
  - 如果返回true，提示重新输入学号
  - 如果返回false，正常添加学生对象

### 5.2 学号被占用的方法说明图

-----欢迎来到学籍管理系统-----

1.添加学生

2.删除学生

3.修改学生

4.查看学生

5.退出

请输入你的选择：

2

请输入学生的学号：

001

你输入的学生学号已经被占用，请重新输入！

请输入学生学号：

CSDN @不爱生姜不吃醋

### 5.3 代码示例

判断学号是否被使用方法

```
1 // 定义一个方法，对学号是否被使用进行判断；
2 public static boolean Isused(ArrayList<Student> array,String num){
3 //不爱生姜不吃醋
4 // 如果与集合中的某一个学生学号相同，返回true
5 // 如果都不相同，返回false;
6 boolean flag =false;
7
8 for(int i=0;i<array.size();i++){
9     Student s=array.get(i);
10    if(s.getNum().equals(num)){
11        flag = true;
12        break;
13    }
14 }
15 return flag;
16 // 在添加学生录入学号后调用该方法
17
```

```
17 // 如果返回true, 提示重新输入学号
18 // 如果返回false, 正常添加学生对象
19 }
```

## 5.4 代码示例

### 改进后的 添加学生信息方法

```
1 public static void Addstudent(ArrayList<Student> array){
2     // 定义一个方法用于添加学生;
3     Scanner sc=new Scanner(System.in);
4
5     String num;
6
7     //为了让程序回到这里, 我们使用循环实现
8     while(true){
9         System.out.print("请输入学生学号: ");
10        num = sc.nextLine();
11
12        boolean flag=Isused(array,num);
13        if(flag){
14            System.out.println("你输入的学号已经被使用, 请重新输入: ");
15        }else{
16            break;
17        }
18    }
19
20    System.out.print("请输入学生姓名: ");
21    String name=sc.nextLine();
22    System.out.print("请输入学生年龄: ");
23    String age=sc.nextLine();
24    System.out.print("请输入学生地址: ");
25    String address=sc.nextLine();
26
27    // 创建学生对象, 将键盘录入的数据赋值给学生对象的成员变量
28    Student s=new Student();
29    s.setNum(num);
30    s.setName(name);
31    s.setAge(age);
32    s.setAddress(address);
33
34    // 将学生对象添加到集合中
35    array.add(s);
36
37    // 给出添加成功提示
38 }
```

```
38         System.out.println("添加学生成功!");  
39     }  
40 }
```

两个方法代码写完后，注意要在 `main` 方法中调用：

首先调用判断学号被使用方法： `Isused (array);`

调用改进后的添加学生信息方法： `Addstudent (array);`

（注意：如果之前有调用了标题4中的方法，应先删去后在调用改进后的添加学生信息方法！）