回顾:

1.方法使用:先定义(在类中)，后调用(对象)

定义:

访问修饰符 返回值类型 方法名称（[数据类型 参数名称,数据类型 参数名称]）{

方法体…..

return 返回值;

}

1. 类类型作为参数传递和返回
2. 类类型作为参数

**public void** getAvg(Student stu){*//学生类作为参数* **int** avg=(stu.**java**+stu.**c**+stu.**sql**) /3;  
 …  
}

1. 类类型作为返回值

|  |
| --- |
| **Public Student** createStudent(){*//学生类作为返回值*  …  Return new Student(); } |

1. 使用方法架构程序.

面向对象编程[Object Oriented Programming, OOP]

是一种计算机编程架构,是20世纪90年代流行起来的一种软件编程思想。

OOP的一条基本原则就是计算机是由单个能够起到子程序[子程序就是方法]

作用的单元或对象组合而成的，采用这种架构编程主要起到重用性、灵活性和扩展性的强大作用。

项目需求:学生管理系统

管理学生的信息是:学号xh、姓名name、年龄age…、在读状态 0 在读 1休学

功能：

1. 添加功能 2.查询学生 3.退出
2. 查询学生分为:1.按姓名查询 2.显示所有 3.返回

效果:

1. 添加功能 2.查询学生 3.退出 一级菜单

请选择功能:2

1.按姓名查询 2.显示所有 3.返回 二级菜单

 实现学生管理系统类StudentManager:

1. 实现菜单的切换

一个菜单输出为一个方法

主菜单:mainMenu 二级菜单:twoMenu

1. 创建学生类Student

|  |
| --- |
| *//定义学生类型* **public class** Student {  *//学号xh、姓名name、年龄age…、在读状态 0 在读 1休学* **public int xh**; *//学号* **public** String **name**; *//姓名* **public int age**; *//年龄* **public short state**; *//在读状态 0 在读 1休学* } |

1. 在学生管理系统类中添加，学生数组用于存放学生信息

*//定义学生数组用于存储学生信息: 10*Student [] **stulist**=**new** Student[10];

1. 实现功能 ：一个功能一个方法

实现添加功能

|  |
| --- |
| *//控制学生人数* **int pos**=0;  *//一功能一方法: // 实现添加学生* **public void** addStudent(){  */\* if(pos>=stulist.length) {  System.out.println("学生人数已满");  //return; //退出方法  this.mainMenu();  }\*/* System.***out***.println(**"录入添加学生信息>>>"**);  *//创建学生对象* Student stu=**new** Student();  System.***out***.println(**"请输入学生学号:"**);  stu.**xh**=**input**.nextInt();  System.***out***.println(**"请输入学生姓名:"**);  stu.**name**=**input**.next();  System.***out***.println(**"请输入学生年龄:"**);  stu.**age**=**input**.nextInt();  *//System.out.println("在读状态采用默认值:");* stu.**state**=0; *//在读  //将学生存入数组* **stulist**[**pos**]=stu; *//????????* **pos**++; *//记录* System.***out***.println(**"添加完毕"**);   *//询问* System.***out***.println(**"是否想继续添加y|n"**);  String flag=**input**.next();  **if**(flag.equals(**"y"**)){  addStudent();  }**else** {  **this**.mainMenu();  } } |

1. 实现显示所有学生

|  |
| --- |
| *//实现显示所有学生信息* **public void** show(){  System.***out***.println(**">>>显示所有学生信息"**);  System.***out***.println(**"学号\t姓名\t年龄\t在读状态"**);  **for** (**int** i=0;i<**stulist**.**length**;i++){  Student s=**stulist**[i];  **if**(s!=**null**){ *//null表示空 解决空指针问题* System.***out***.println(s.**xh** +**"\t"**+s.**name**+**"\t"**+s.**age**+**"\t"**+(s.**state**==0?**"在读"**:**"休学"**));  }  }    System.***out***.println(**"是否想返回上一级y|n"**);  String flag=**input**.next();  **if**(flag.equals(**"y"**)){  **this**.mainMenu();  }**else** {  **this**.twoMenu();  } } |

1. 按姓名查找学生

|  |
| --- |
| *//实现显示所有学生信息* **public void** showStudentByName(){  System.***out***.println(**">>>请输入学生姓名:"**);  String name=**input**.next();   System.***out***.println(**"学号\t姓名\t年龄\t在读状态"**);  **for** (**int** i=0;i<**stulist**.**length**;i++){  Student s=**stulist**[i];  **if**(s!=**null** && s.**name**.equals(name)){ *//null表示空 解决空指针问题* System.***out***.println(s.**xh** +**"\t"**+s.**name**+**"\t"**+s.**age**+**"\t"**+(s.**state**==0?**"在读"**:**"休学"**));  }  }    System.***out***.println(**"是否想返回上一级y|n"**);  String flag=**input**.next();  **if**(flag.equals(**"y"**)){  **this**.mainMenu();  }**else** {  **this**.twoMenu();  } } |

作业:

1. 完成学生显示所有和通过姓名显示学生
2. 在主菜单中添加一个菜单

常规业务

>>>1.办理休学(通过学号) 2.办理复学 3.返回

选做:

显示所有学生时，要求按学生的年龄升序排列.

简单的:Collection.sort 查 或者 冒泡排序