实验内容

根据“实验2任务书”文件夹中给出的源程序Main.java，StuGroup.java（其中的类实现了接口MyGroup），Student.java（其中的Student是抽象类，print( )方法是抽象方法）：

1. 创建源程序MyGroup.java，其中的内容是接口MyGroup的声明。（注意，需要将源程序Main.java，StuGroup.java，Student.java补充完整。）
2. 创建源程序Undergraduate.java，其中的类Undergraduate继承Student.java中的抽象类Student，实现其中的抽象方法print( )。
3. 创建源程序Postgraduate.java，其中的类Postgraduate继承源程序Student.java中的抽象类Student，实现其中的抽象方法print( )，并且，在父类的基础上增加两个新的属性：导师和研究方向。
4. StuGroup.java，MyGroup.java，Undergraduate.java，Postgraduate.java和Student.java在同一个包中。Main.java在无名包中。
5. 在程序中做出注释，哪些地方是上转型？哪些地方体现了运行时多态？

上述源程序可以完成如下任务：

使用者根据提示信息，选择从键盘上输入本科生或者研究生的学生信息，可以添加学生信息、删除学生信息、显示学生信息、按照学生成绩排序等。

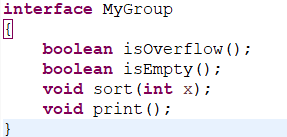
实现方法

通过抽象类 Student 建立了学生信息的通用结构，不同类型的学生可以继承并实现自己的输出方式。具体的学生类型（本科生和研究生）通过继承抽象类 Student 并实现 print() 方法来定义各自的信息输出方式。StuGroup 类实现了接口 MyGroup 中定义的方法，用于管理学生信息，通过数组来存储学生信息，进行溢出、空状态、添加、删除、排序和打印操作。

具体如下：

1. **接口 MyGroup：**

用于定义不同类型的表通用的方法，包括判断表是否溢出isOverflow()、判断表是否为空isEmpty()、对表中元素进行排序、打印等。



StuGroup 类是针对学生表的一种表，表中存的对象为Student类对象，该类实现了这个接口，提供了接口中定义的方法的具体实现。

1. **抽象类 Student：**

用于表示学生信息，包括学号（int）,姓名（String）,数学成绩（double），计算机成绩（double）等属性。

提供了具体的构造方法、Getter 和 Setter 方法，以及抽象的 print() 方法要求子类实现。

1. **具体类 Undergraduate：**

继承了 Student 抽象类，表示本科生。实现了 print() 方法，以定义本科生信息的输出格式。

1. **具体类 Postgraduate：**

继承了 Student 抽象类，表示研究生。实现了 print() 方法，以定义研究生信息的输出格式。包含额外的属性和方法，用于表示导师和研究方向。

1. **类 StuGroup：**

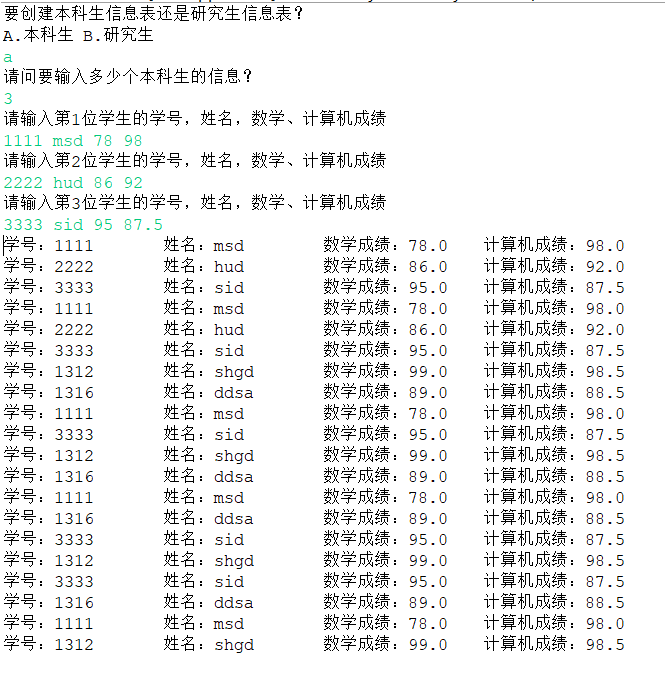
对接口 MyGroup的实现，用于实现管理学生信息，构造学生表。可以包含本科生和研究生的信息。Student 类型的数组stu用于存储学生信息，length 变量表示实际容量。

提供了判断是否溢出、判断是否为空、添加学生、删除学生、排序和打印等方法。其中判断是否溢出、判断是否为空、排序和打印是对接口中方法声明的实现。

实验结果

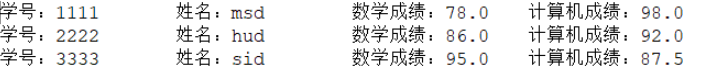
该项目中用指定输出测试程序运行结果，以下为程序截图。

**1.本科生信息表的写入和显示：**

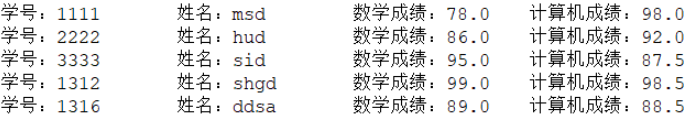


**显示分别为：**

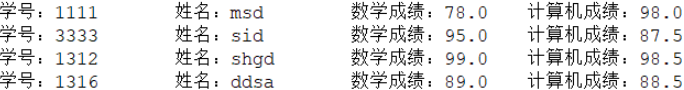
1. 首次输入完毕后的学生表



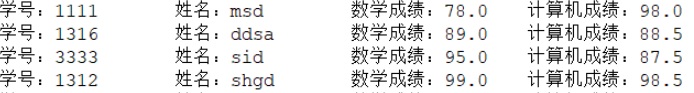
1. 新加入两个学生后的学生表



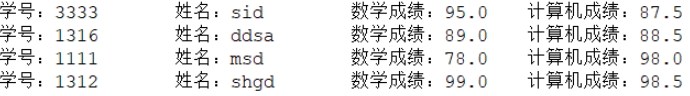
1. 删除第二个学生后的学生表



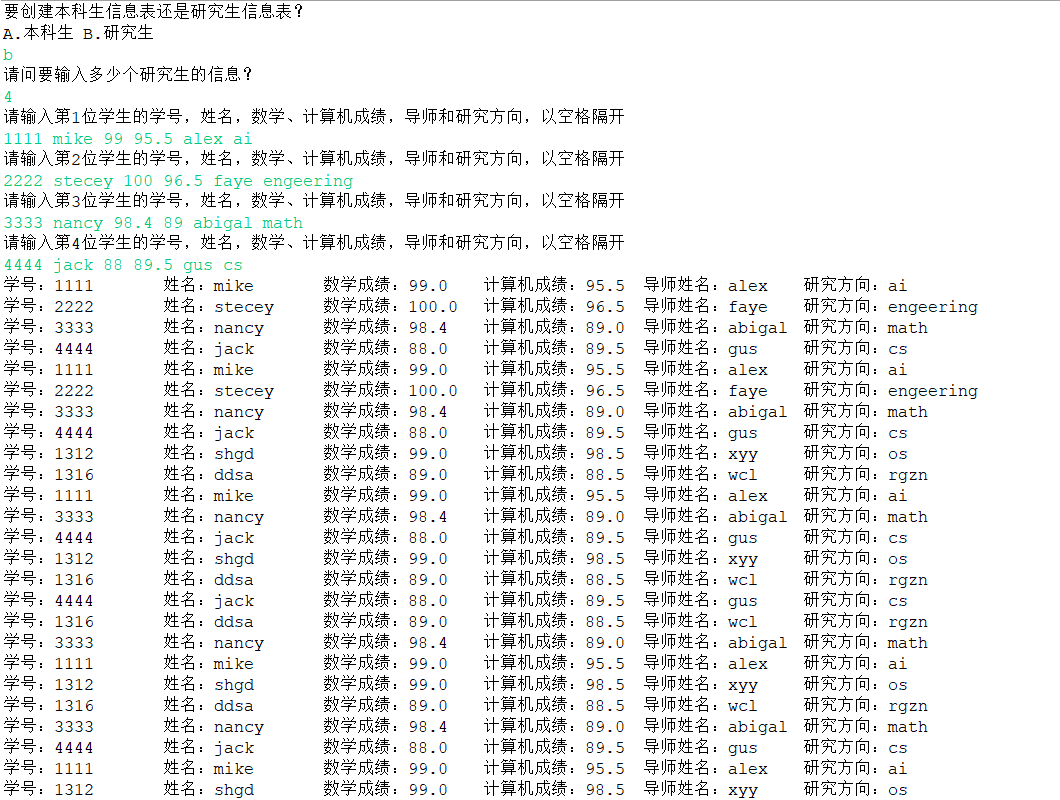
1. 按数学成绩排序后的学生表



1. 按计算机成绩排序后的学生表



**2.研究生信息表的写入和显示：**

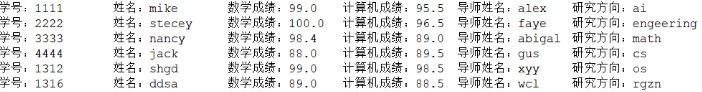


**显示分别为：**

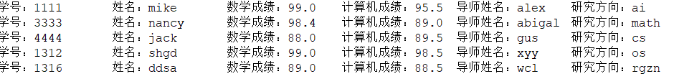
1. 首次输入完毕后的学生表



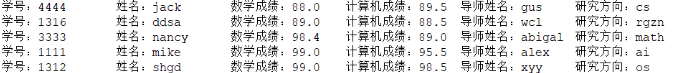
1. 新加入两个学生后的学生表



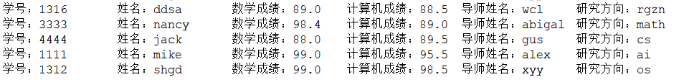
1. 删除第二个学生后的学生表



1. 按数学成绩排序后的学生表



1. 按计算机成绩排序后的学生表



结论分析

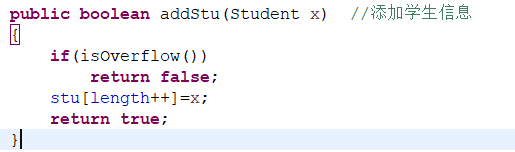
1. 对上转型和运行时多态的分析：

在类的高级特性中，上转型和运行时多态是重要概念，它们通常涉及父类和子类之间的关系。

1. **上转型**是将子类对象赋值给父类引用的操作，这使得可以通过父类引用访问子类对象的方法和属性。

例如：在往学生表中添加学生信息时，**addStu方法中传入的参数———Postgraduate或者Undergraduate对象转换为父类Student类对象。**实现了在学生表中添加不同类型的学生的操作。

addStu方法实现：

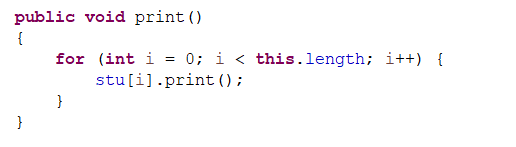


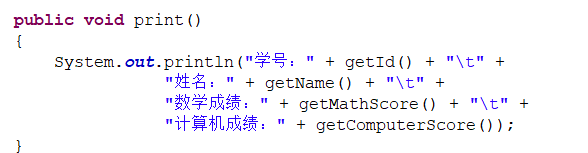
两种类对象作为参数：

group.addStu(**new** Undergraduate(id, name, mathScore, computerScore));//上转型

group.addStu(**new** Postgraduate(id, name, mathScore, computerScore, tutor, direction));//上转型

1. **运行时多态**是指在运行时根据对象的实际类型来调用适当的方法，而不是根据引用变量的类型。

例如：在**StuGroup类中的print方法中**，学生表中存储的是不同类型的学生信息，由于Student是不同学生类的父类，执行stu[i].print语句时，**在需要打印本科生的信息时调用Undergraduate的print方法，在需要打印研究生的信息时调用Postgraduate的print方法。**

undergraduate.print(); 

postgraduate.print();

