

Project::StudentList

Due::TBA

Overview:

In StudentList, you will create an **ArrayList** of Student objects. You will create methods that will allow the user to:

- add (add a new student record)
- delete (delete a student record)
- print (print a student record to the console)
- search (search for a student by name or student number)
- clear (delete all students from a list)

Library classes and methods:

ArrayList: `add(spot,item), add(item), set(spot, item), get(spot), size(), remove(), clear()`

Scanner: `nextLine(), nextInt(), nextDouble(), next()`

String: `equals(), length(), substring(a), substring(a,b), indexOf()`

StudListRunner class: Add a while loop that will run a main menu and allow the user to choose an activity

- When the user runs the program, a menu will allow the user to choose between the following program activities. These methods will be **called from** the runner class and implemented in the StudentList class. Assume you have a StudentList Object declared as: `StudentList studList = new StudentList()`. Then, in **StudentListRunner**, you can call StudentList methods with your StudentList object:

- `studList.addStudentToList()`
- `studList.deleteStudentFromList()`
- `studList.editStudentList()`
- `studList.clearList()`
- `studList.printAll()`
- `studList.printStudent()`

StudList class: Instance variables and methods

Field:

- Create an ArrayList of Student objects named: `studList`

Methods:

- `addStudent()` :: Allows the user to add a student to the end of the List. For each student, the user can input a student number, a GPA, and a student name. Names may be entered in one of three formats::
1. Last, First 2. Last, First Middle 3. First Middle Last
- `deleteStudent()` :: Allows the user to specify a Student last name deleted from the list.
- `editStudentList()` :: Allows the user to specify a Student number or student name and then update or change the Student name and GPA
- `clearList()` :: Deletes the entire list (null in each element without a student record).
- `printList()` :: Clears the screen and then prints the list to the screen. Does not print null records.
- `printStudent()` :: Asks the User to input a student name or number. Then clear the screen and then prints that student record to the screen. If the Student does not exist, print message to the console: "Student does not exist."

Student class: Instance variables and methods

→ Instance variables:

- ◆ `String firstName`
- ◆ `String middleName`
- ◆ `String lastName`
- ◆ `int stuNumber`
- ◆ `double gpa`

→ Methods:

- ❑ Create **getters** for
 - `fullName::format = "Last, First Middle".`
 - `gpa(Two decimal places)`
 - `stuNumber (six digits)`

- ❑ Create **setters** only for:
- ★ `firstName, middleName, lastName, stuNumber, gpa`

BenchMark One: Monday before class

StudListRuner completed:

- Instantiate a StudentList object
- Create and call menuNumber method that displays menu, gets user input, and returns the appropriate integer
- Select the appropriate StudList method according to user input OR end program

StudentList Class completed:

- Declare and initialize an ArrayList of student objects
- Create methods:
 - `addStudentToList()` :: The name String does **not need to be altered**.
 - `printStudentList()` :: Prints student info in formatted rows

Student Class completed:

- All instance variable created
- Getters and setters created according to instructions above

BenchMark Two: Monday During Class

Complete the methods:

- `addStudentToList()` in the Student List class
- User may input name in one of three formats:
 - " First Middle Last", "First Last", "Last, First Middle"
- Helper method `parseUserInput()` in the StudentList class, that will take a String and Student Object and set the instance variables in the student object according to the user input string. (e.g. `firstName`, `middleName`, and `lastName`)

BenchMark Three:

`StudList::deleteStudentFromList()`

- Input: User choice of student to be deleted-(Name or Stud ID)
- Check student exists: Throw a message
- Delete the student at that index from the list
- Throw exception for illegal input //!!!!!!!!!!!!!! Optional

BenchMark Four:

StudList::filterSearchStudentList()

- Gets filter: student number or gpa
- If user selects gpa input, prompt asks for minimum gpa
- If user selects last stuNum input, prompt asks for minimum stuNum
- Find and print all students whose gpa is less than or equal to the given GPA **OR** find and print all students whose stuNum is less than or equal to the given stuNum **OR** print "No such student exists"

Project Complete: