WRAV102/MSEV102: Homework exercise

Please attempt before your second session of the week

Objectives

- Classes and Objects, Text files
- List class, Arraylist
- Sorting and Searching

This task does not have to be submitted, but you are advised to attempt it before your second WRAV101/MSEV102 session of the week of 2 October.

Task

Create a console application which can be used to read in and do some calculations on marks.
Open the file `Marks.txt`, review this file and take note of the following:

- Each line of the file is representing the details for a specific student, namely: student number, surname, quiz average, prac average and test mark
- Your application should read in the details from the file, create a student object for each student and add the objects to a list of students

To store the list of students you should make use of a list class (e.g. StudentList) and store the Student objects in an arraylist.

Student class:

- Create a Student object class, with private properties for the following fields: student number, surname, quiz average, prac average, test mark and class mark. Note that the class mark should be calculated by using a method (more detail below)
- Add a constructor method.
- Add any accessor and mutator methods as required by the required functionality.
- Add a method to calculate the class mark for a specific student. The formula to calculate the class marks is as follows:
  `Class mark = Quiz average *0.2 + Prac average * 0.2 + Test mark * 0.6`
- Add a method (`displayStudent()`) to display the student number, surname and class mark for a specific student,. Class mark should be displayed with 2 decimal places

StudentList class:

- Make use of an arraylist to store the student objects.
- Keep track of the sorted state of the list, `sorted state = 0` if the list is unsorted, and `sorted state = 1` if the list is sorted in ascending order of `student number`.
- Implement the Insertion Sort algorithm to sort the list of student in ascending order of student number
- Implement a add, find, get, binary search and linear search method that will be needed for the required functionality of the application
- Add a display method which will use the display method from the object class to display all the student details from the list
- Add a method which will determine the highest class mark (you may not implement another sorting method to do this), and then uses the display method from the object class to display the details of the student who have obtained the highest mark

Application class:
Your application should provide the user with a menu of options, from which he can continuously select a task to complete.
Implement a Main method and any required methods that would be required to do the following”

- Allows the user to add a student.
- Allow the user to input a student number to search for a student in the list, and when found the details for that student should be displayed *(use the `displayStudent()` method to display the student details)*, otherwise an error message should be displayed. Your application must provide for both the linear and binary searching methods, and use the relevant one, based on the sorted state of the list.
- Display the list of students - display the student number, name and class mark (make use of the display method in the list class)
- Sort and display the list of students in ascending order of student number (use the method from the list class)
- Display the details of the student with the highest class mark (make use of the method created in the list class)

It is not required that you write the data from the student list back to a text file.