Exercise 13 *ArrayList*

Exercise 1.13 Notes

13.1.1 Student notes

You want to save an object in the class `ArrayList` the grades obtained by a student in a given subject.
The program shall include the following operations, each carried out by a method using as parameter an object of the `ArrayList` containing the notes: **add a new note**, **view all notes**, **calculate the average**.
The `main`() program will perform a test with notes 12, 14, 9, starting by calculation of the average, then adding note 13 and recalculation of the average and display of the notes.

Question 13.1.2 - Error Correction
A new method will change a note of the table. This method can be useful in case of error in entering or correcting the copy.

Question 13.1.3 - Menu
Modify the `main`() program so that different operations are available through a menu.

Question 13.1.4 - Coefficients

Provide the notes with a certain integer coefficient to be taken into account when calculating the average. To do this, we will save the data in two coordinated arrays: an arrays of notes and an array of coefficients. The link between the two will be the index: the index of a note (i) placed in box of array notes is the as the index of coefficient box (i) stored in the array of coefficients.

Exercise 13.2 - Sets

Question 13.2.1 - Creating sets

We will use objects from the `ArrayList` class to represent sets of integers. The program will offer the following:

- **create** a singleton (set of single element, given as a parameter). This creates a new set, so a new object of the `ArrayList` class.
- **add** an element to an existing array. It will be changed.
- **display** a set.
- **calculate** the **union** of two sets. The result is a new set, so a new object of the `ArrayList` class.
- **calculate** the **intersection** of two sets. The result is a new set, so a new object of the `ArrayList` class.

The `main`() program will make some calls to different methods.

Question 13.2.2 - Properties of sets

Add the following two operations:

- **test** the membership of an element to a set.
- **give** the number of elements in a set.

For the second operation, be careful not to double count of the same item. The best way to do this is to ensure that an element appears only once in an object representing a whole set.