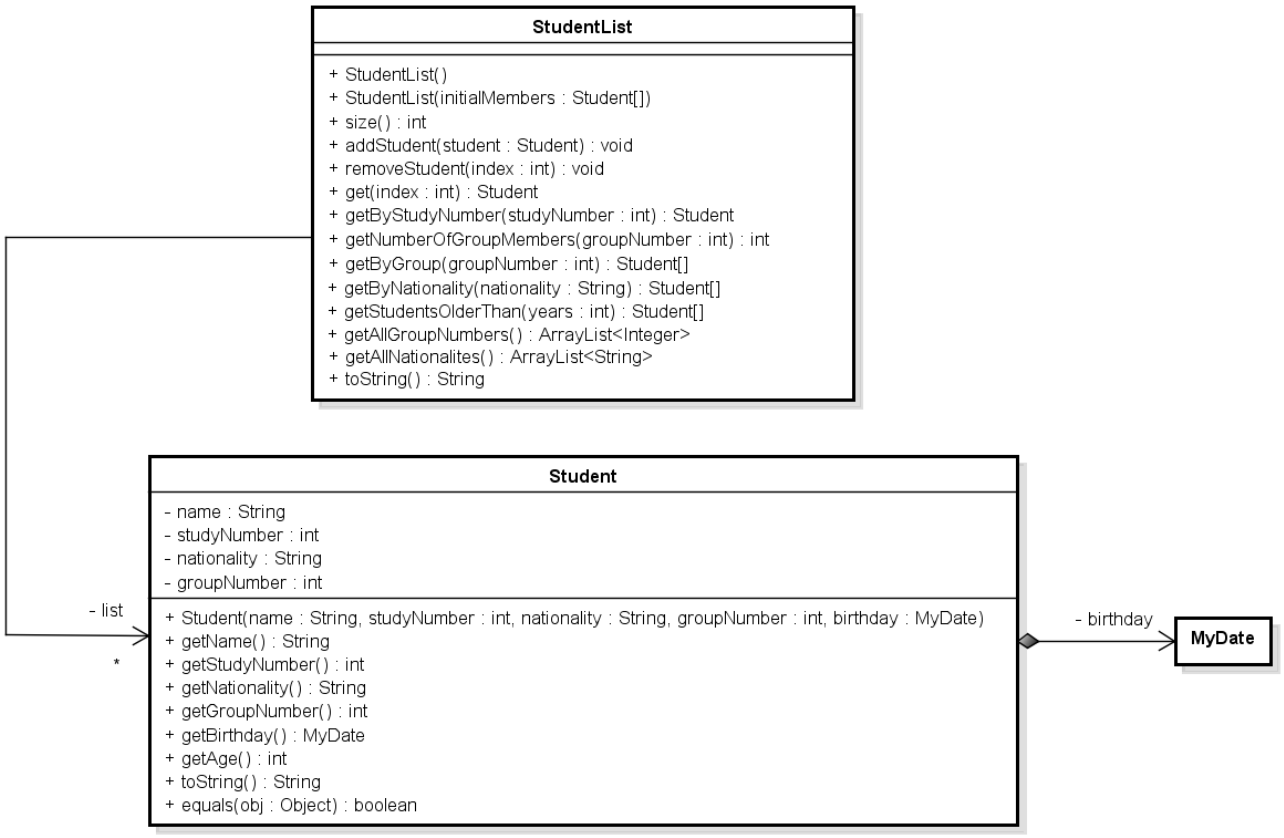
Exercise: StudentList

Implement the following system (reuse your class `MyDate`):



Notes to class `MyDate`: Reuse a version you already implemented, with at least:

- A zero-argument constructor creating todays date
- A method `yearsBetween`.
- An `equals` method
- A `copy` method

Notes to class `Student`:

- 5 instance variables
- Constructor (note the composition relation)
- Getters for all instance variables
- A method `getAge` returning the age of the student
- A `toString` and an `equals` method

Notes to class `StudentList`:

- An `ArrayList` of `Student`-objects as the only instance variable
- Two constructors. The one-argument constructor is adding a list (an array) of students to the ArrayList instance variable.
- The next 4 methods just delegates to the `ArrayList`
- Method `getByStudyNumber` uses a loop to find the student with the study number specified.

- Method `getNumberOfGroupMembers` uses a loop to count the number of students with the group number as specified.
- Method `getByGroup` creates an array and uses a loop to find and store the students with the group number as specified.
- Methods `getByNationality` return an array containing the students with the nationality as specified. Note that you need to return an array and you do not have a method to find its size.
- Method `getStudentsOlderThan` uses a loop to find and store the students with an age larger than the age specified.
- Method `getAllGroupNumbers` uses a loop to find each of the distinct group numbers to be returned as an `ArrayList` of `Integer` objects. Note that you inside the loop have to make sure that you do not add the same group number multiple times.
- Method `getAllNationalities` is similar to `getAllGroupNumbers` this time returning an `ArrayList` of `Strings` with all distinct nationalities.
- Method `toString` uses a loop to append each `Student`-object to a string to be returned. *Advanced version (if you like a challenge):* Let the string contain students orders by nationality (i.e. all students of one nationality, e.g. `DK` first, then students of another nationality e.g. `RO`, and so forth). You may also challenge yourselves to order students by group numbers instead.